

Cisco Catalyst IR1800 Rugged Series Router Software Configuration Guide

First Published: 2020-08-13

Last Modified: 2026-09-28

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

© 2021–2026 Cisco Systems, Inc. All rights reserved.



CONTENTS

Full Cisco Trademarks with Software License ?

CHAPTER 1

Prerequisites 1

Cisco Catalyst IR1800 Rugged Series routers 1

Initial bootup security and configuration guidelines 2

Configure initial bootup security on a Cisco router 3

Accessing the CLI on Cisco IR1800 routers 5

Access the CLI via console port 6

Accessing the CLI from a remote console 7

Prepare to connect to the router console 7

Set up the router to run SSH 8

Use Telnet to access a console interface 9

CLI session management 10

Change the CLI session timeout 10

Lock a CLI session 11

CHAPTER 2

New Features 13

New Features for Cisco IOS XE 26.2.1 13

New Features for Cisco IOS XE 26.1.1 13

New Features for Cisco IOS XE 17.18.2 14

New Features for Cisco IOS XE 17.18.1a 14

New Features for Cisco IOS XE 17.17.1 14

New Features for Cisco IOS XE 17.16.1a 14

New Features for Cisco IOS XE 17.15.1a 14

New Features for Cisco IOS XE 17.14.1a 15

New Features for Cisco IOS XE 17.13.1 15

New Features for Cisco IOS XE 17.12.1a 15
New Features for Cisco IOS XE 17-11-1a 15
New Features for Cisco IOS XE 17.10.1a 16
New Features for Cisco IOS XE 17.9.1 16
New Features for Cisco IOS XE 17.8.1 16
Cellular serviceability enhancements 16
New Features for Cisco IOS XE 17.7.1 17
Support 1G SFPs 17

CHAPTER 3 **Basic Router CLI Configuration** 19
IR1800 Interface Naming 19
Basic Configuration 20
Configuring Global Parameters 24
Configuring the Gigabit Ethernet Interface 25
Support for sub-interface on GigabitEthernet0/0/0 26
Configuring a Loopback Interface 26
Enabling Cisco Discovery Protocol 27
Configuring Command-Line Access 28
Configuring Static Routes 29
Configuring Dynamic Routes 31
Configuring Routing Information Protocol 31
Configuring Enhanced Interior Gateway Routing Protocol 32
Modular QoS (MQC) 33
Configuring the Serial Interface 33
Specifying an Asynchronous Serial Interface 34
Specifying Asynchronous Serial Encapsulation 34
Configuring the Serial Port 34

CHAPTER 4 **Web User Interface** 37
Introduction to the Web User Interface 37
Day 0 Cellular Mode 38
Day 0 Web User Interface 38
Additional Modem Support for CAT 6 and CAT 7 Cellular Pluggable Modules 38
Additional Modem Support for Cellular Pluggable Modules 39

Galileo support on the LTE pluggable modules 40
GPS mode default enablement 41
Guidelines and Limitations 42
Configuring Your Computer to Connect to the Router 42
Connecting to the Router Using DHCP 43
Configuring Basic Mode WebUI through the Browser 45
Configuring Advanced Mode WebUI through the Browser 50
WebUI Dashboard 56
Configure access point name using Cisco WebUI 56

CHAPTER 5 **Secure Shell** 61
Secure shell 61
Prerequisites for configure secure shell 61
Restrictions for configuring secure shell 61
SSH and router access 62
SSH servers, integrated clients, and supported versions 62
Recommendation: SSH configuration guidelines 63
How to configure secure shell 63
Set up the router to run SSH 64
Configure the SSH server 65
Monitor the SSH configuration and status 66
Configure the router for local authentication and authorization 67
Secure copy protocol 68
Requirements for secure copy configuration 68
Restrictions for configuring secure copy 69
Configure secure copy 69
SSH rekeying 70
SSH rekeying types 71
Prerequisites for configuring SSH rekeying 71
Restrictions for SSH rekeying on Cisco routers 72
Configure SSH rekeying 73
Verify SSH rekey configuration 74
Additional references 75

CHAPTER 6 **Inline Tagging** 77

SGT Inline Tagging 77

Guidelines for Inline Tagging 78

Configuring SGT Static Inline Tagging 79

Example: Configuring SGT Static Inline Tagging 80

Troubleshoot the Security Group Tagging Configuration 80

CHAPTER 7 **NTP Timing Based on GPS Clock** 81

Configure NTP using GPS Time 81

CHAPTER 8 **Managing Configuration Files** 85

Understanding Configuration Files 85

Finding the Software Version 86

Managing and Configuring a Consolidated Package Using copy and boot Commands 86

Upgrading the Router Image through the WebUI 88

CHAPTER 9 **Using Cisco IOS XE Software** 91

Command modes 91

Use keyboard shortcuts 93

No and default command forms 93

Use the history buffer to recall commands 94

Configuration files management 94

Save configuration changes 95

Filtering output from show and more commands 95

Cisco feature navigator 96

Find support information for platforms and Cisco software images 96

Get help 97

Find command options 97

Software advisor 101

Use software release notes 101

CHAPTER 10 **Cisco IOS XE installation methods** 103

Bundle mode versus install mode 103
Installing software using install commands 104
Restrictions for installing the software using install commands 104
Install mode support 104
Information about installing the software using install commands 105
Installing software using install mode commands 106
Boot the platform in install mode 107
Convert from bundle mode to install mode or upgrade the platform using one-step installation 108
Three-step installation 109
Upgrading in install mode 111
Downgrading in install mode 111
Software installation termination 112
Configuration examples 112
One step installation 112
Complete three-step software installation 114
Show the installed packages 117
Show committed and uncommitted packages 118
Remove inactive packages 118
Troubleshoot software installation using install commands 119

CHAPTER 11 **Installing the Software** 121
Installing the Software 121
Licensing 121
Cisco Software Licensing 121
Consolidated Packages 121
Network-Essentials 122
Network-Advantage 122
How to Install the Software for Cisco IOS XE 122
Installing the Cisco IOS XE Release 122
ROMMON Images 124
File Systems 124
Option to Enable or Disable USB Access 125
Autogenerated File Directories and Files 126
Flash Storage 127

LED Indicators 127
Related Documentation 128
IOS XE downgrade warning 128
Enable the secure data wipe capability 128

CHAPTER 12 **Software Maintenance Upgrade (SMU) 131**
Software Maintenance Upgrade (SMU) 131
SMU Work-flow and Basic Requirements 132
SMU Example 132
Installing a Patch Image 132
Uninstalling the Patch Image 135
Uninstalling the Patch Image Using Rollback 135
Uninstalling the Patch Image Using Deactivate, Commit, and Remove 137

CHAPTER 13 **Smart Licensing Using Policy 141**
Smart licensing using policy on Cisco routers 141

CHAPTER 14 **Configuring Ethernet Switch Ports 143**
Configuring VLANs 143
VLAN Trunking Protocol (VTP) 144
Configuring 802.1x Authentication 144
Configuring Spanning Tree Protocol 145
Configuring MAC Address Table Manipulation 147
Configuring Switch Port Analyzer 148
Configuring IGMP Snooping 149
Configuring EVPN VXLAN VLAN-Aware Service 149
Profile based Configuration of EVPN VXLAN VLAN-Aware Service 150
Configure EVPN VXLAN VLAN-Aware instance 150
Apply the configuration on a Bridge-Domain 151
Manual configuration of EVPN VXLAN VLAN-Aware Service 152
Static Configuration of VXLAN VLAN-Aware EVPN Instance 152
Applying Configuration on the Bridge Domain 153
Configure Ingress Replication for EVPN VXLAN VLAN-Aware 153
Configure Static Replication for VLAN-Aware 155

Configuration Example 156
Verify Configuration of EVPN VXLAN VLAN-Aware Service 162
Multi-Homing Configuration in BGP EVPN VXLAN Fabric 169
Configuring Multi-Homing in a BGP EVPN VXLAN Fabric 170
Configure Ethernet Segment and Redundancy in the Ethernet Segment 171
Configure Multi-Homing VLAN and Associate EVPN Ethernet-Segment 171
Configure SVI Service Instance 173
Configure EVPN-Instance 173
Apply the Configuration on the Bridge-Domain 174
Configuration Example 175
Verify the Multi-Homing Configuration 182
Misconfiguration of Multihoming VLANs 190

CHAPTER 15 **IP Device Tracking** 195
IP Device Tracking on routers 195
Guidelines and limitations 196
Options to enable SISF-based device tracking 196
Create and attach a custom device tracking policy 196
Apply the default device tracking policy to an interface target 198
Configuration and verification examples 199

CHAPTER 16 **Power Over Ethernet (PoE)** 201
Power over Ethernet Overview 201
Device Detection and Power Allocation 201
Command Line Interface 201

CHAPTER 17 **vCPU and RAM Distribution** 205
Introduction 205
Distribution of vCPU and RAM Resources for Cisco IOx Applications 205
Higher CPU and RAM Allocation for IOx Applications 206
Configure Data Plane Heavy Template 206
Verify the Active vCPU and RAM Distribution 207
Configure Service Plane Heavy Template 207
Verify the Active vCPU and RAM Distribution 208

CHAPTER 18 Cellular pluggable interface modules 209

CHAPTER 19 Cisco Wi-Fi Interface Module (WIM) 217

Cisco Wi-Fi Interface Module (WIM) Overview 217

Cisco IoT Operations Dashboard (OD) Support to Configure and Manage the WP-WIFI6-x Module 217

WIM firmware upgrades 217

List of WIM firmware upgrade commands 218

CHAPTER 20 Digital I/O, Ignition, and CAN Bus Connectivity 219

Overview 219

Digital IO 220

Controller Area Network (CAN) Bus 220

IOx CAN Bus Support 221

Important Notes 221

Packet Capture Support for CANBUS 221

Configuring Digital IO 222

Ignition Power Management Overview 223

Features of Ignition Power Management 223

Ignition Sense Overview 224

IR1835 Ignition Switch 225

IR1800 Ignition and Battery Voltage 227

Command Line Interface (CLI) 227

Default Values 229

Ignition Power Management Yang Model 229

Support SNMP MIB for Ignition Power Management 230

CHAPTER 21 Configuring GPS 233

GPS Overview 233

Cellular Modem-Based GPS 235

GPS/Dead Reckoning module (IRM-GNSS-ADR) 235

GPS Dead Reckoning 235

Dead Reckoning Overview 236

Command Line Interface 236

Feature Limitations 240
IR1800 GPS DR Module Calibration 241
Dead Reckoning for GPS NMEA data streaming 244
GPS and Dead Reckoning Support for the J1939 Connector 244
National Marine Electronics Association (NMEA) IOx Support 246
NMEA UDP Socket Support 246
NMEA UDP Configuration with Yang 250
Yang Data Model Support 252
Example: Connecting to a Server Hosting a GPS Application 253
GNSS Support on the GPS/Dead Reckoning Module (IRM-GNSS-ADR) 254
Galileo support on the LTE pluggable modules 255
Access Accelerometer and Gyro Sensor Data from IRM-GNSS 256
Change in Vendor for GNSS Module 256
IOX Access to IR1800 On-board Accelerometer and Gyroscope 257
Configure DR Module using Cisco Catalyst SD-WAN Manager 258
GPS NMEA unique identifier 258
Configure dead reckoning GPS NMEA unique identifier 260

CHAPTER 22 **Information About SCADA** 263
Supervisory Control And Data Acquisition (SCADA) Overview 263
Role of the IR1800 263
Key Terms 264
Protocol Translation Application 264
Prerequisites 265
Guidelines and Limitations 266
Default Settings 266
Configuring Protocol Translation 266
Enabling the IR1800 Serial Port and SCADA Encapsulation 266
Enable Serial Port Example 267
Configuring T101 and T104 Protocol Stacks 267
Protocol Stack Prerequisites 267
Configuring the T101 Protocol Stack 267
T101 Protocol Stack Example 269
T101 Configuration Example 270

Configuring the T104 Protocol Stack 271
Configure T104 Protocol Stack Example 274
Configuring the DNP3 Protocol Stacks 275
Configuring DNP3 Serial 275
DPN3-Serial Protocol Stack Example 276
Configuring DNP3 IP 276
DNP3 IP Parameters Example 278
Starting and Stopping the Protocol Translation Engine 279
Start Protocol Translation Engine Example 279
SCADA enhancement for TNB 279
Verifying Configuration 280
SCADA Debug Commands 280

CHAPTER 23 **Raw Socket Transport** 283
Raw Socket Transport Overview 283
Information About Raw Socket Transport 283
TCP Transport 284
UDP Transport 285
Serial Data Processing 285
VRF-Aware Raw Socket 285
Prerequisites 286
Guidelines and Limitations 286
Default Settings 286
Configuring Raw Socket Transport 286
Enabling Raw Socket Transport on the Serial Interface 286
Enable Serial Port Example 287
Configuring Common Raw Socket Line Options 287
Configuring Common Raw Socket Line Options Example 288
Configuring Raw Socket TCP 288
Configuring the Raw Socket TCP Server 288
Configuring the Raw Socket TCP Client 289
Raw socket feature enhancement 291
Configuring a Raw Socket UDP Peer-to-Peer Connection 291
Raw Socket UDP Connection Example 292

Rawsocket Keepalive Configuration CLI 292
Verifying Configuration 293
Raw Socket Transport Configuration Examples 293
Raw Socket TCP 293
Raw Socket UDP Example 294
Raw Socket VRF 295

CHAPTER 24 IOx Application Hosting 297
Application Hosting 297
Information About Application Hosting 297
Need for Application Hosting 297
IOx Overview 298
Cisco Application Hosting Overview 298
IOXMAN 298
Application Hosting on the IR1800 Industrial Integrated Services Router 299
VirtualPortGroup 299
vNIC 300
How to Configure Application Hosting 301
Enabling IOx 301
Configuring a VirtualPortGroup to a Layer 3 Data Port 302
Installing and Uninstalling Apps 304
Overriding the App Resource Configuration 304
Verifying the Application Hosting Configuration 306
Configuration Examples for Application Hosting 307
Example: Enabling IOx 307
Example: Configuring a VirtualPortGroup to a Layer 3 Data Port 307
Example: Installing and Uninstalling Apps 307
Example: Overriding the App Resource Configuration 308
Native docker support 308
Digital IO for IOx container applications 309
Signed Application Support 310
Cisco ThousandEyes Enterprise Agent 310

CHAPTER 25 Serial Relay Service 313

IOx Serial Relay Service 313
Data Paths 313
Configuration Commands 315

CHAPTER 26 Support for MACsec 317
Software Supported MACsec 317

CHAPTER 27 ROM Monitor Overview 319
ROM Monitor Overview 319
Access ROM Monitor Mode 320
Checking the Current ROMMON Version 320
Commonly Used ROM Monitor Commands 321
Examples 322
Changing the ROM Monitor Prompt 322
Displaying the Configuration Register Setting 322
Environment Variable Settings 322
Frequently Used Environmental Variables 323
Displaying Environment Variable Settings 323
Entering Environment Variable Settings 323
Saving Environment Variable Settings 323
Exiting ROM Monitor Mode 324
ROMMON Configuration Example 324
Upgrading the ROMmon for a Router 325

CHAPTER 28 Connected Grid NMS Agent (CGNA) Support 327
Connected Grid NMS Agent (CGNA) Support 327
CGNA Overview 327
WebSocket Support 328

CHAPTER 29 CLI Output for the FN980 5G Modem 329
Change in CLI output for the FN980 5G modem 329

CHAPTER 30 Unified Threat Defence 331

Unified Threat Defense (UTD) 331

CHAPTER 31 **Support for CAPWAP and WGB Modes on the Cisco Wi-Fi Interface Module** 333

Support for CAPWAP and WGB Modes on the Cisco Wi-Fi Interface Module 333

Configuring IR1800 for deploying WGB 335

Configuring a QoS Profile 335

Configuring an SSID Profile With Open Authentication Without a QoS Profile Mapped 335

Configuring an SSID Profile With Open Authentication With a QoS Profile Mapped 336

Configuring an SSID Profile with WPA2 Personal Authentication Without a QoS Profile Mapped 337

Configuring an SSID Profile with WPA2 Personal Authentication With a QoS Profile Mapped 338

Configuring a Dot11radio in WGB Mode and Configuring Various Parameters 339

Configuring a Dot11Radio in uWGB Mode and Configuring Various Parameters 340

Configuring a Dot11radio in Root AP Mode and Configuring Various Parameters 341

Configure IPv4 address 342

Configure IPv6 address 343

Verify the WGB Mode Configuration, Monitoring Operational Status 343

Additional Commands 345

Firmware Upgrade 345

CHAPTER 32 **Additional Modem Support for Cellular Pluggable Modules** 347

Additional Modem Support for CAT 6 and CAT 7 Cellular Pluggable Modules 347

Additional Modem Support for Cellular Pluggable Modules 348

5G Standalone Mode (SA) Support 348

CHAPTER 33 **Support for P-LTE-450 Pluggable Interface Module** 351

Support for P-LTE-450 Pluggable Interface Module 351

Enhancing P-LTE-450 Network Management Through SNMP MIB Objects 352

VRF for the P-LTE-450 Module 353

Configure VRF for the P-LTE-450 module 354

Upgrade the P-LTE-450 PIM 355

CHAPTER 34 **Support for P-5GS6-RC-GL PIM** 357

Support for P-5GS6-RC-GL PIM 357

CHAPTER 35 **Cellular Boot Time Improvements** 359

Cellular boot time improvements 359

CHAPTER 36 **Digital Subscriber Line (DSL) SFP Support on the IR1800** 361

Digital Subscriber Line (DSL) SFP Support on the IR1800 361

CHAPTER 37 **LoRaWAN Pluggable Interface Module Support** 363

LoRaWAN Pluggable Interface Module Support 363

CHAPTER 38 **Cisco SD-WAN Support** 365

Cisco SD-WAN Overview 365

vManage Support for the WP-WIFI6-x Module 366

SD-WAN 367

Related Documentation 367

vManage Support for EWC Mode on the Cisco Wi-Fi Interface Module 368

CHAPTER 39 **Troubleshooting** 369

Troubleshooting 369

Understanding Diagnostic Mode 369

Before Contacting Cisco or Your Reseller 370

show interfaces Troubleshooting Command 370

Software Upgrade Methods 371

Change the Configuration Register 371

Configuring the Configuration Register for Autoboot 373

Reset the Router 373

Recovering a Lost Password 374

Reset the Password and Save Your Changes 375

Reset the Configuration Register Value 376

Configuring a Console Port Transport Map 376

Viewing Console Port, SSH, and Telnet Handling Configurations 378

Using the factory reset Commands 380

CHAPTER 40 **System Messages** 383

Information About Process Management 383

How to Find Error Message Details 383

CHAPTER 41 **Environmental Monitoring** 389

Environmental Monitoring and Reporting Functions 389

Environmental Monitoring Functions 389

Environmental Reporting Functions 391

SNMP Polling of Temperature OID 398

Additional References 398

Technical Assistance 399

CHAPTER 42 **Process Health Monitoring** 401

Monitoring Control Plane Resources 401

Avoiding Problems Through Regular Monitoring 401

Cisco IOS Process Resources 401

Overall Control Plane Resources 408

Monitoring Hardware Using Alarms 410

Router Design and Monitoring Hardware 410

BootFlash Disk Monitoring 411

Approaches for Monitoring Hardware Alarms 411

Viewing the Console or Syslog for Alarm Messages 411

Enabling the logging alarm Command 411

Network Management System Alerts a Network Administrator when an Alarm is Reported Through SNMP 411

CHAPTER 43 **WAN Monitoring** 413

Information About WANMon 413

Built-in Recovery Actions 413

Prerequisites 414

Guidelines and Limitations 414

Configuring WANMon 415

Verifying WANMon Configuration 416

Configuration Examples 417
WANMon Cellular Interface Configuration Example 417
Multiple WAN Link Monitoring Example 417

CHAPTER 44 Yang Data Models 419
Support for YANG Data Models (Call-home) 419
Yang Data Model Support for Raw Socket Transport 419
Yang Data Model Support for Scada 420

CHAPTER 45 gRPC Network Operations 421
gRPC Network Operations Interface updates 421
GNMI Broker update 421

CHAPTER 46 Cellular Network Slicing 423
Cellular network slicing 423
Restrictions for cellular network slicing 424
Configure cellular network slicing 424
Example for configuring cellular network slicing 425
Verify cellular network slicing configuration 425

CHAPTER 47 Change of Authorization 427
Change of authorization 427
Information about change of authorization 428
How change of authorization reauthentication work 428
Change of authorization requests 429
Limitations for change of authorization 430
How to configure change of authorization 430
Create RADIUS server group 430
Configure 802.1X authentication using a RADIUS server 431
Configure 802.1X authentication on an access port 432
Configure RADIUS change of authorization 434
Configuration examples for change of authorization 435
RADIUS server status example 435
Verification of change of authorization 435

Successful 802.1X authentication and CoA reauthentication 435
Troubleshoot RADIUS change of authorization configuration 435

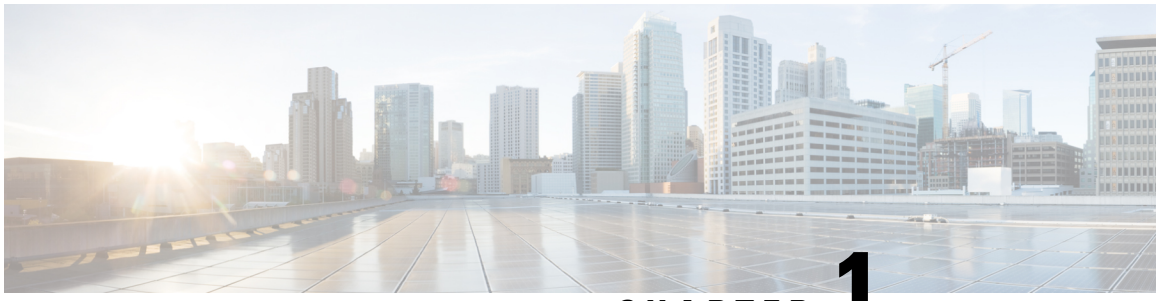
CHAPTER 48 **Password encryption types for wireless profile configuration 437**
Password encryption types for wireless profile configuration 437
Prerequisites 438
Limitations 438
Configure wireless module in EWC mode 439
Configure wireless module in WGB mode 439
Verify EWC autonomous and controller mode configurations 440
Verify WGB autonomous and controller mode configurations 440

CHAPTER 49 **IKEv2 and IPsec PFS Post Quantum Cryptography 443**
IKEv2 and IPsec PFS Post Quantum Cryptography 443
Benefits of PQC 444
Prerequisites 444
Limitations 444
Phased migration in Hub-and-Spoke deployments 445
ML-KEM algorithm and parameter sets 445
PFS negotiation behavior 446
Configure an IKEv2 proposal for PQC 446
Configure Perfect Forward Secrecy with PQC 448
Verify the PQC hybrid key exchange 449
Troubleshooting PQC hybrid key exchange 450

CHAPTER 50 **Post Quantum Cryptography for SSHv2 sessions 451**
Implementation of Post Quantum Cryptography 451
Overview 452
Factors to consider for deploying PQC 452
Protocol requirements 452
Restrictions for PQC 452
PQC negotiation scenarios 453
Configure PQC for SSHv2 sessions 453
Verify PQC for SSHv2 sessions 454

Key elements of PQC 454
Supported platforms 456

CHAPTER 51 DHCP Option 82 457
DHCP Option 82 457
Prerequisites 458
Key considerations 459
Limitations 459
DHCP Option 82 packet flow 459
Hexadecimal pattern and suboption decoding 460
Configure DHCP Option 82 on server 461
Configure DHCP address pool 463
Configure DHCP relay agent 464
Configure relay chaining on relay agents 465
Verify DHCP Option 82 466
Debug DHCP Option 82 logs 467



CHAPTER 1

Prerequisites

This section contains the following:

- Cisco Catalyst IR1800 Rugged Series routers, on page 1
- Initial bootup security and configuration guidelines, on page 2
- Accessing the CLI on Cisco IR1800 routers, on page 5
- Access the CLI via console port, on page 6
- Accessing the CLI from a remote console, on page 7
- CLI session management, on page 10

Cisco Catalyst IR1800 Rugged Series routers

Cisco Catalyst IR1800 Rugged Series Routers are modular industrial routers that:

- Features four base platforms with modularity for additional pluggable modules.
- Provides flexibility to add different interfaces to the base platform.
- Supports industrial-grade performance for harsh environments.

The series includes four base platforms:

- IR1821: Lite
- IR1831: Base B
- IR1833: Base M
- IR1835: Pro

Each platform supports pluggable modules, enabling customization for specific use cases.

It provides modular design with pluggable modules such as PIM, mSATA, GNSS, and Wi-Fi Interface Modules.

The table displays the features of the base platforms:

Features	IR1821	IR1831	IR1833	IR1835
Processor	600MHz	600MHz	600MHz	1200MHz
Memory	4GB	4GB	4GB	8GB

Features	IR1821	IR1831	IR1833	IR1835
PIM Slot(s)	1	2	2	2
WiFi Pluggable Module Slot	Yes	Yes	Yes	Yes
PoE	No	No	Yes	Yes
mSATA Pluggable Module	No	No	Yes	Yes
GNSS Pluggable Module	No	No	Yes	Yes
GPIO	No	No	No	Yes
Ignition Management	Yes	Yes	Yes	Yes
CAN Bus	Yes	Yes	Yes	Yes
Serial Interface	RS232 (1)	RS232 (2)	RS232 (2)	RS232 (1) RS232/RS485 (1)
Advanced Security	No	No	No	Yes, Cisco Umbrella Integration

Initial bootup security and configuration guidelines

This document outlines the initial security and configuration measures for a Cisco device during the first boot or after a factory reset.

When a Cisco device is first booted after a factory reset or new from the factory, these security measures and configurations are enforced:

- Enforce changing default password:

  In the initial configuration dialog, set a new enable password and enter the **enable secret** command for stronger encryption. Weak passwords are rejected until a strong password is entered, adhering to the standard mix of upper/lower case characters, special characters, and numbers.

- Password management:

  The **enable secret** command is recommended over the **enable secret** command due to its stronger encryption algorithm. The **service password-encryption** command can be used to encrypt passwords in the configuration file, but it is not designed to protect against sophisticated attacks.

- Telnet and HTTP configuration changes:

  Release 17.3.1 implements the following changes:

  - Telnet is disabled.
  - The HTTP server is disabled, but the HTTP client remains functional.

- SSH is enabled for secure access.
- The HTTPS server is enabled.

Configure initial bootup security on a Cisco router

To setup management information, security passwords, and network management settings, while also recognizing conditions that cause the dialog to be skipped, such as Cisco PnP centralized provisioning.

Follow the prompts to set up basic parameters such as hostname, passwords, and IP addresses.

Procedure

Step 1 Start the initial configuration dialog.

Example:

```
 --- System Configuration Dialog ---

Would you like to enter the initial configuration dialog? [yes/no]: yes

At any point you may enter a question mark '?' for help.
Use ctrl-c to abort configuration dialog at any prompt.
Default settings are in square brackets '[]'.
```

Step 2 Set global parameters.

Enter the hostname and configure the enable secret password.

```
Basic management setup configures only enough connectivity
for management of the system, extended setup will ask you
to configure each interface on the system

Would you like to enter basic management setup? [yes/no]: yes
Configuring global parameters:

  Enter host name [Router]: <your-host-name>

  The enable secret is a password used to protect access to
  privileged EXEC and configuration modes. This password, after
  entered, becomes encrypted in the configuration.
  Enter enable secret: <your-password>

  The enable password is used when you do not specify an
  enable secret password, with some older software versions, and
  some boot images.
  Enter enable password: <your-password>
```

Step 3 Configure virtual terminal password.

Set a password to secure network access.

Example:

```
The virtual terminal password is used to protect
  access to the router over a network interface.
  Enter virtual terminal password: <your-password>
Setup account for accessing HTTP server? [yes]: <return>
    Username  [admin]: <your-username>
```

```
    Password  [cisco]: <your-password>
    Password is UNENCRYPTED.
```

Step 4 Configure hostname and IP addresses.

Example:

Displays interface summary:

```
Current interface summary

Any interface listed with OK? value "NO" does not have a valid configuration

Interface              IP-Address      OK? Method Status                Protocol
GigabitEthernet0/0/0   unassigned      NO  unset  up                    up
GigabitEthernet0/1/0   unassigned      YES unset  down                  down
GigabitEthernet0/1/1   unassigned      YES unset  down                  down
GigabitEthernet0/1/2   unassigned      YES unset  down                  down
GigabitEthernet0/1/3   unassigned      YES unset  up                    up
Async0/2/0             unassigned      YES unset  up                    down
Vlan1                  unassigned      YES unset  up                    up
```

Displays example names and IP addresses:

```
Enter interface name used to connect to the
management network from the above interface summary: vlan1

Configuring interface Vlan1:
  Configure IP on this interface? [no]: yes
    IP address for this interface: 192.168.1.1
    Subnet mask for this interface [255.255.255.0] : <return>
    Class C network is 192.168.1.0, 24 subnet bits; mask is /24

Would you like to configure DHCP? [yes/no]: yes
 Enter DHCP pool name: wDHCPool
 Enter DHCP network: 192.168.1.0
 Enter DHCP netmask: 255.255.255.0
 Enter Default router: 192.168.1.1

The following configuration command script was created:

hostname <your-hostname>
enable secret 9 $9$Z6fl74fvoEdMgU$XZYs8l4phbqpXsb48l9bzCng3u4Bc2kh1STsoLoHNes
enable password <your-enable-password>
line vty 0 4
password <your-password>
username <your-username> privilege 15 password <your-password>
no snmp-server
!
!
interface GigabitEthernet0/0/0
shutdown
no ip address
!
interface GigabitEthernet0/1/0
!
interface GigabitEthernet0/1/1
!
interface GigabitEthernet0/1/2
!
interface GigabitEthernet0/1/3
!
interface Vlan1
no shutdown
ip address 192.168.1.1 255.255.255.0
```

```
no mop enabled
ip dhcp pool wDHCPool
network 192.168.1.0 255.255.255.0
default-router 192.168.1.1
!
end
```

Step 5 Save the configuration.

Example:

```
[0] Go to the IOS command prompt without saving this config.
[1] Return back to the setup without saving this config.
[2] Save this configuration to nvram and exit.

Enter your selection [2]: 2
Building configuration...

[OK]
Use the enabled mode 'configure' command to modify this configuration.
```

Step 6 Verify the configuration.

Example:

```
router-1>en
Password:
router-1#sh run | sec enable
enable secret 9 $9$emUzIshVXwlUaE$nTzhgi9STdZKzQc4VJ0kEaCqafjUNdCD7ZUf37SY9qg
```

Step 7 Review the initial configuration dialog.

If you answer **no** to the initial configuration dialog:

Example:

```
Would you like to enter the initial configuration dialog? [yes/no]: no
The enable secret is a password used to protect access to
  privileged EXEC and configuration modes. This password, after
  entered, becomes encrypted in the configuration.
  Enter enable secret: ********
  Confirm enable secret: ********
Would you like to terminate autoinstall? [yes]: yes

.
```

Accessing the CLI on Cisco IR1800 routers

There are two methods to access the router's CLI:

- Accessing the CLI using a router console
- Accessing the CLI from a remote console

Accessing the CLI using a router console

The Cisco IR1800 router has a micro-B USB console port located on the front panel of the chassis. This port only supports USB connectivity. The default baud rate for the console connection is 9600 bps.

For detailed steps, see *Access the CLI via console port*.

Accessing the CLI from a remote console

The CLI of the Cisco IR1800 routers can also be accessed remotely using Telnet or SSH. By default, Telnet is disabled, and the more secure SSH protocol is recommended for use.

For detailed steps, see *Access the CLI via remote console*.

Access the CLI via console port

The purpose of this task is to enable access to the CLI in privileged EXEC mode, allowing the you to execute necessary commands for device configuration and management.

Before you begin

- Install the required drivers from the manufacturer or download them from Silicon Labs or FTDI.
- The latest VCP drivers may not support macOS 10.14.x and beyond. For older macOS versions, use driver version 3.1.
- Follow the prompts to set up basic parameters such as hostname, passwords, and IP addresses. See *Configure initial bootup security on a Cisco router*. If the router is pre-configured for Cisco PnP services, the initial dialog may be skipped.

Procedure

Step 1 Enter the enable command.

Example:

```
Router > enable
```

Step 2 Enter the system password (if configured).

At the password prompt, type your system password.

Example:

```
Password: enablepass
```

Step 3 Once accepted, the privileged EXEC mode prompt is displayed.

Example:

```
Router#
```

You now have access to the CLI in privileged EXEC mode and can execute necessary commands.

Step 4 Exit the console session.

Example:

```
Router# quit
```

Accessing the CLI from a remote console

This topic provides an overview of accessing the IR1800 router's CLI remotely via Telnet or SSH. By following the outlined procedures, you can securely and effectively manage the router remotely.

There are two methods to access the router's CLI remotely:

- Telnet:

  An older protocol that transmits data in plaintext, making it less secure. Telnet is disabled by default on the Cisco IR1800 router.

- SSH (Secure Shell):

  A more secure protocol that encrypts data during transmission. SSH should be used for remote access whenever possible. For detailed SSH configuration instructions, refer to the SSH chapter.

Here are the necessary steps to establish a remote CLI session with the IR1800 router:

1. Prepare to connect to the router console.
2. Set up the router to run SSH.
3. Use Telnet to access a console interface.

Prepare to connect to the router console

To prepare the router console remotely using Telnet from a TCP/IP network:.

Before you begin

- Ensure the router has a valid hostname and an IP address configured.
- For SSH access, configure user authentication for local or remote access.
- Refer to the Cisco IOS-XE Device Hardening Guide for additional security configurations.

Procedure

Step 1 Configure the router to support virtual terminal lines.

Use the **line vty** global configuration command.

Step 2 Set up login requirements and specify a password.

Use the **password** and **login** commands.

Step 3 If using AAA, configure the `login authentication` command and ensure the AAA authentication list is properly set up.

For more information about AAA services, see the Cisco IOS XE Security Configuration Guide: Secure Connectivity and the Cisco IOS Security Command Reference documents. For more information about the **login line-configuration** command, see the Cisco IOS Terminal Services Command Reference document.

Step 4 Optionally, configure diagnostic and wait banners to provide status indicators for Telnet or SSH attempts.

Set up the router to run SSH

SSH provides secure remote access to network devices. This configuration enables encrypted communication and helps protect against unauthorized access attempts.

Note Ensure that you configure user authentication for local or remote access.

Follow these steps to set up the router to run SSH.

Procedure

Step 1 Use the **enable** command to enter privileged EXEC mode.

Example:

```
Router> enable
```

Step 2 Use the **configure terminal** command to enter global configuration mode.

Example:

```
Router# configure terminal
```

Step 3 Use the **hostname** *your_hostname* and **ip domain-name** *your_domain_name* commands to configure a hostname and domain name for the device.

Example:

```
Router(config)# hostname <your_hostname>
Router(config)# ip domain-name <your_domain_name>
```

Step 4 Use the **crypto key generate rsa** command to generate an RSA key pair to enable SSH.

Example:

```
Router(config)# crypto key generate rsa
```

Note
Use a modulus size of at least 1024 bits for security.

Step 5 Use the **end** command to exit global configuration mode.

Example:

```
Router(config)# end
```

Step 6 Use the **show running-config** command to verify the configuration.

Example:

```
Router# show running-config
```

Step 7 (Optional) Use the **copy running-config startup-config** command to save the configuration.

Example:

```
Router# copy running-config startup-config
```

The router is now configured to accept SSH connections. You can connect remotely using an SSH client with the configured hostname and the appropriate user credentials.

Use Telnet to access a console interface

To access the console interface using Telnet.

Before you begin

Since Telnet is disabled by default, enable it only if necessary, and understand its security limitations. Use Telnet for initial testing or in trusted network environments. For detailed steps, see Configuring Telnet

Procedure

Step 1 From your terminal or PC, enter one of the following commands:

Example:

```
connect host [port] [keyword]
```

```
telnet host [port] [keyword]
```

- **<host>**: Router's hostname or IP address.
- **[port]**: Port number (default is 23).

Example:

Example shows how to use the **telnet** command to connect to a router named **router**:

```
unix_host% telnet router
Trying 172.20.52.40...
Connected to 172.20.52.40.
Escape character is '^]'.
unix_host% connect
```

Step 2 Enter your login password when prompted.

Example:

```
User Access Verification
Password: mypassword
```

If no password has been configured, press **Return**.

Step 3 Access the privileged EXEC mode.

Use the **enable** command and provide the system password.

Example:

```
Router> enable
```

```
Password: enablepass
```

Privileged EXEC mode prompt is displayed. You now have access to the CLI in privileged EXEC mode and can execute necessary commands.

```
Router#
```

Step 4 Exit the Telnet session.

Use the **exit** or **logout** command.

Example:

```
Router# logout
```

CLI session management

A CLI session is a management interface that allows configuration of inactivity timeouts. It provides session locking to prevent overwriting of changes by multiple users and reserves spare capacity to ensure CLI access even under high system load.

Two methods of CLI session management

- Change the CLI session timeout
- Lock a CLI session

Change the CLI session timeout

To manage a CLI session.

Procedure

Step 1 Enter global configuration mode

Example:

```
Router# configure terminal
```

Step 2 Specify the console line.

Example:

```
Router(config)# line console 0
```

Step 3 Set the session timeout value in minutes.

Example:

```
Router(config)# session-timeout <minutes>
```

Use **0** to disable the timeout.

The **minutes** value determines the CLI duration. Increasing the CLI session timeout enhances session security.

Step 4 Verify the session timeout configuration.

Example:

```
Router(config)# show line console 0
```

Lock a CLI session

To lock a CLI session.

Procedure

Step 1 Enter the global configuration mode.

Example:

```
Router# configure terminal
```

Step 2 Specify the console line to be locked.

Example:

```
Router(config)# line console 0
```

Step 3 Enable the line to be lockable.

Example:

```
Router(config)# lockable
```

Step 4 Exit the configuration mode:

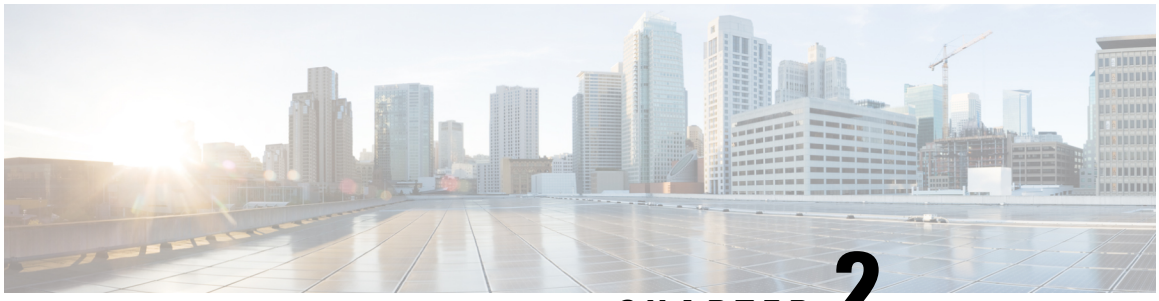
Example:

```
Router(config)# exit
```

Step 5 Lock the session and set a temporary password.

Example:

```
Router# lock
Password: <password>
Again: <password>
Locked
```



CHAPTER 2

New Features

- New Features for Cisco IOS XE 26.2.1, on page 13
- New Features for Cisco IOS XE 26.1.1, on page 13
- New Features for Cisco IOS XE 17.18.2, on page 14
- New Features for Cisco IOS XE 17.18.1a, on page 14
- New Features for Cisco IOS XE 17.17.1, on page 14
- New Features for Cisco IOS XE 17.16.1a, on page 14
- New Features for Cisco IOS XE 17.15.1a, on page 14
- New Features for Cisco IOS XE 17.14.1a, on page 15
- New Features for Cisco IOS XE 17.13.1, on page 15
- New Features for Cisco IOS XE 17.12.1a, on page 15
- New Features for Cisco IOS XE 17-11-1a, on page 15
- New Features for Cisco IOS XE 17.10.1a, on page 16
- New Features for Cisco IOS XE 17.9.1, on page 16
- New Features for Cisco IOS XE 17.8.1, on page 16
- New Features for Cisco IOS XE 17.7.1, on page 17

New Features for Cisco IOS XE 26.2.1

New features in this release are:

- Password encryption types for wireless profile configuration
- DHCP Option 82
- IKEv2 and IPsec PFS Post Quantum Cryptography
- Post Quantum Cryptography for SSHv2 sessions
- Support for P-5GS6-RC-GL PIM

New Features for Cisco IOS XE 26.1.1

New features in this release are:

- Change of Authorization, on page 427

- GPS NMEA unique identifier , on page 258

New Features for Cisco IOS XE 17.18.2

New features in this release are:

- Cellular Network Slicing
- Cisco ThousandEyes Enterprise Agent, on page 310
- Common HSEC License

New Features for Cisco IOS XE 17.18.1a

New feature in this release is listed below:

- IP Device Tracking
- VRF for the P-LTE-450 module
- Inline Tagging on switch interfaces

New Features for Cisco IOS XE 17.17.1

New features in this release is listed below:

- Multi-Homing Configuration in BGP EVPN VXLAN Fabric

New Features for Cisco IOS XE 17.16.1a

New features in this release are listed below:

- Access Accelerometer and Gyro Sensor Data from IRM-GNSS, on page 256
- Configure DR Module using Cisco Catalyst SD-WAN Manager
- Configuring EVPN VXLAN VLAN-Aware Service, on page 149
- Monitoring Radio Signal Parameters and GPS Coordinates of Cellular Telemetry using Syslog Messages
- Enhancing P-LTE-450 Network Management Through SNMP MIB Objects

New Features for Cisco IOS XE 17.15.1a

New features in this release are listed below:

- Software Supported MACsec, on page 317

- Higher CPU and RAM Allocation for IOx Applications

New Features for Cisco IOS XE 17.14.1a

New features in this release are listed below:

- Support for CAPWAP and WGB Modes on the Cisco Wi-Fi Interface Module
- Additional Modem Support for Cellular Pluggable Modules
- Support for P-LTE-450 Pluggable Interface Module

New Features for Cisco IOS XE 17.13.1

This chapter contains the following sections:

- Change in Vendor for GNSS Module
- IOX Access to IR1800 On-board Accelerometer and Gyroscope
- Unified Threat Defense (UTD)
- vManage Support for EWC Mode on the Cisco Wi-Fi Interface Module
- Additional Modem Support for Cellular Pluggable Modules
- SD-WAN Remote Access
- Change in CLI Output for the FN980 5G Modem

New Features for Cisco IOS XE 17.12.1a

New features in this release are listed below:

- Access Accelerometer and Gyro Sensor Data from IRM-GNSS
- 5G Standalone Mode (SA) Support
- Uncapped License Implementation

New Features for Cisco IOS XE 17-11-1a

New features in this release are listed below:

- LoRaWAN Pluggable Interface Module Support
- GNSS Support on the GPS/Dead Reckoning Module (IRM-GNSS-ADR)
- Galileo Support on the LTE Pluggable Modules

- Change to Smart Licensing Packaging
- Cisco IoT Operations Dashboard (OD) Support to Configure and Manage the WP-WIFI6-x Module

New Features for Cisco IOS XE 17.10.1a

New features in this release are listed below:

- Digital Subscriber Line (DSL) SFP Support on the IR1800
- vManage Support for the WP-WIFI6-x Module
- Enable Secure Data Wipe Capabilities
- Rawsocket Keepalive Configuration CLI

New Features for Cisco IOS XE 17.9.1

New features in this release are listed below:

- Install Mode Support
- Cellular Boot Time Improvements
- IOS XE Downgrade Warning
- SNMP Polling of Temperature OID
- GPS Mode Enabled By Default
- Packet Capture Support for CANBUS
- GPS and Dead Reckoning Support for the J1939 Connector

New Features for Cisco IOS XE 17.8.1

- Raw socket feature enhancement, on page 291
- SCADA enhancement for TNB, on page 279
- gRPC Network Operations Interface updates, on page 421
- GNMI Broker update, on page 421

Cellular serviceability enhancements

Cellular serviceability enhancements are debugging features that:

- enable trigger points and debug code via controller cellular CLIs for automatic generation and trapping of debug data without manual intervention,

- provide context-based debug logs including tracebacks and GPS locations, and
- create well-formatted debug messages with vendor-specific debug data at a broader range.

Available CLI options

The following CLI options are available for cellular serviceability:

```
(config-controller)#lte modem serviceability ?
  gps               GPS debugging
  interface-resets  Interface resets/Bearer deletion
  modem-crash       Modem-crash debugging
  modem-resets      IOS initiated unknown modem-resets
```

The debug data includes:

- Context Based debug logs (tracebacks, and GPS locations).
- Well formatted debug messages.
- Vendor specific debug data at a broader range.

The debug logs are located in the following location of flash:

```
router#dir flash:servelogs
Directory of bootflash:/servelogs/

259340  -rw-              122   Sep 7 2021 17:40:44 +00:00  gpslog-slot5-20210907-174044
259339  -rw-             1734   Sep 7 2021 12:14:07 +00:00  celllog-slot5-20210905-164628
```

GPS and cellular log files are created separately with file names using the timestamp at the time of the creation. These files are created as follows:

- If the existing file has reached 10Mb, a new file will be created.
- A new file will be created if the feature (GPS, or cellular) is completely disabled, and then re-enabled.

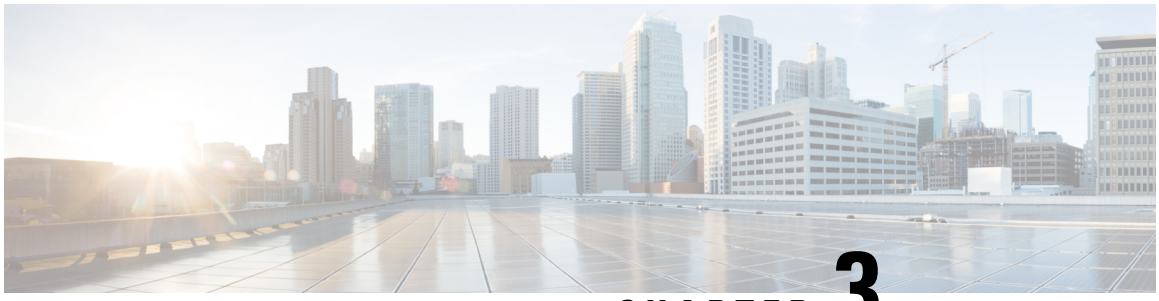
New Features for Cisco IOS XE 17.7.1

Support 1G SFPs

Release 17.7.1 will add support for these SFPs:

- GLC-T-RGD
- CWDM-SFP-1470=
- CWDM-SFP-1610=
- CWDM-SFP-1530=
- DWDM-SFP-3033=
- DWDM-SFP-3112=
- GLC-BX-D-I=
- GLC-BX-U-I=

- GLC-TE



CHAPTER 3

Basic Router CLI Configuration

This chapter contains the following sections:

- IR1800 Interface Naming, on page 19
- Basic Configuration, on page 20
- Configuring Global Parameters, on page 24
- Configuring the Gigabit Ethernet Interface, on page 25
- Support for sub-interface on GigabitEthernet0/0/0, on page 26
- Configuring a Loopback Interface, on page 26
- Enabling Cisco Discovery Protocol, on page 27
- Configuring Command-Line Access, on page 28
- Configuring Static Routes, on page 29
- Configuring Dynamic Routes, on page 31
- Modular QoS (MQC), on page 33
- Configuring the Serial Interface, on page 33

IR1800 Interface Naming

Descriptions and graphics of the router interfaces are found in the Hardware Installation Guide.

The supported hardware interfaces and their naming conventions are in the following table:

Hardware Interface	Naming Convention
Gigabit Ethernet combo port	GigabitEthernet0/0/0
Gigabit Ethernet ports	GigabitEthernet0/1/0 GigabitEthernet0/1/1 GigabitEthernet0/1/2 GigabitEthernet0/1/3
Cellular Interface	cellular 0/4/0 cellular 0/4/1 cellular 0/5/0 cellular 0/5/1

Hardware Interface	Naming Convention
Asynchronous Serial Interface	async 0/2/0 async 0/2/1 (When the base platform supports two async serial interfaces)
USB	usbflash0:
mSATA	msata
GPIO	alarm contact 1-4

Basic Configuration

The basic configuration is a result of the entries you made during the initial configuration dialog. This means the router has at least one interface set with an IP address to be reachable, either through WebUI or to allow the PnP process to work. Use the **show running-config** command to view the initial configuration, as shown in the following example:

```
Router# show running-config
Building configuration...

Current configuration : 7008 bytes
!
! Last configuration change at 00:01:55 GMT Sun Sep 20 2020
!
version 17.6
service timestamps debug datetime msec
service timestamps log datetime msec
service call-home
platform qfp utilization monitor load 80
platform punt-keepalive disable-kernel-core
!
hostname IR1800
!
boot-start-marker
boot system bootflash:/ir1800-universalk9.17.06.01prd18.SPA.bin
boot-end-marker
!
!
!
no aaa new-model
clock timezone GMT -7 0
!
ignition off-timer 120
!
ignition undervoltage threshold 9 600
!
no ignition sense
!
no ignition enable
!
!
!
!
!
!
!
```

```
ip domain name cisco.com
ip dhcp excluded-address 10.0.0.1
!
ip dhcp pool webui_int
import all
network 10.0.0.0 255.255.255.0
dns-server 10.0.0.1
default-router 10.0.0.1
lease 0 2
!
!
!
login block-for 60 attempts 3 within 30
login delay 3
login on-success log
!
!
!
!
!
!
!
subscriber templating
!
!
!
!
!
!
multilink bundle-name authenticated
!
!
!
!
!
!
!
!
!
crypto pki trustpoint SLA-TrustPoint
enrollment pkcs12
revocation-check crl
!
crypto pki trustpoint TP-self-signed-2276770909
enrollment selfsigned
subject-name cn=IOS-Self-Signed-Certificate-2276770909
revocation-check none
rsakeypair TP-self-signed-2276770909
!
!
crypto pki certificate chain SLA-TrustPoint
certificate ca 01
30820321 30820209 A0030201 02020101 300D0609 2A864886 F70D0101 0B050030
32310E30 0C060355 040A1305 43697363 6F312030 1E060355 04031317 43697363
6F204C69 63656E73 696E6720 526F6F74 20434130 1E170D31 33303533 30313934
3834375A 170D3338 30353330 31393438 34375A30 32310E30 0C060355 040A1305
43697363 6F312030 1E060355 04031317 43697363 6F204C69 63656E73 696E6720
526F6F74 20434130 82012230 0D06092A 864886F7 0D010101 05000382 010F0030
82010A02 82010100 A6BCBD96 131E05F7 145EA72C 2CD686E6 17222EA1 F1EFF64D
CBB4C798 212AA147 C655D8D7 9471380D 8711441E 1AAF071A 9CAE6388 8A38E520
1C394D78 462EF239 C659F715 B98C0A59 5BBB5CBD 0CFEBEA3 700A8BF7 D8F256EE
4AA4E80D DB6FD1C9 60B1FD18 FFC69C96 6FA68957 A2617DE7 104FDC5F EA2956AC
7390A3EB 2B5436AD C847A2C5 DAB553EB 69A9A535 58E9F3E3 C0BD23CF 58BD7188
68E69491 20F320E7 948E71D7 AE3BCC84 F10684C7 4BC8E00F 539BA42B 42C68BB7
```

```
C7479096 B4CB2D62 EA2F505D C7B062A4 6811D95B E8250FC4 5D5D5FB8 8F27D191
C55F0D76 61F9A4CD 3D992327 A8BB03BD 4E6D7069 7CBADF8B DF5F4368 95135E44
DFC7C6CF 04DD7FD1 02030100 01A34230 40300E06 03551D0F 0101FF04 04030201
06300F06 03551D13 0101FF04 05300301 01FF301D 0603551D 0E041604 1449DC85
4B3D31E5 1B3E6A17 606AF333 3D3B4C73 E8300D06 092A8648 86F70D01 010B0500
03820101 00507F24 D3932A66 86025D9F E838AE5C 6D4DF6B0 49631C78 240DA905
604EDCDE FF4FED2B 77FC460E CD636FDB DD44681E 3A5673AB 9093D3B1 6C9E3D8B
D98987BF E40CBD9E 1AECA0C2 2189BB5C 8FA85686 CD98B646 5575B146 8DFC66A8
467A3DF4 4D565700 6ADF0F0D CF835015 3C04FF7C 21E878AC 11BA9CD2 55A9232C
7CA7B7E6 C1AF74F6 152E99B7 B1FCF9BB E973DE7F 5BDDEB86 C71E3B49 1765308B
5FB0DA06 B92AFE7F 494E8A9E 07B85737 F3A58BE1 1A48A229 C37C1E69 39F08678
80DDCD16 D6BACECA EEBC7CF9 8428787B 35202CDC 60E4616A B623CDBD 230E3AFB
418616A9 4093E049 4D10AB75 27E86F73 932E35B5 8862FDAE 0275156F 719BB2F0
D697DF7F 28
quit
crypto pki certificate chain TP-self-signed-2276770909
certificate self-signed 01
30820330 30820218 A0030201 02020101 300D0609 2A864886 F70D0101 05050030
31312F30 2D060355 04031326 494F532D 53656C66 2D536967 6E65642D 43657274
69666963 6174652D 32323736 37373039 3039301E 170D3230 30393230 31343036
30365A17 0D333030 39323031 34303630 365A3031 312F302D 06035504 03132649
4F532D53 656C662D 5369676E 65642D43 65727469 66696361 74652D32 32373637
37303930 39308201 22300D06 092A8648 86F70D01 01010500 0382010F 00308201
0A028201 0100BAC3 88D3A9B7 259E58A4 0FCF6DB2 2794CC97 CF8DC253 D1CFB83B
ACFA305A 28BA6174 2452EE0B C45E92EA BBA30235 C142D2D3 DA04C6FD A916507C
BAFE6806 BBAB6B02 86B5AC61 05FB5A67 C5449A92 EFAA9519 9A2A084E 94A29BF5
B78604F2 76927505 371AD917 67D8EACF CEBBA6A1 278F5647 DDDBE8AF 8E451772
4709D928 04039C51 C2FA72E2 0C03C426 BB844F76 0BE65C37 60DFDA8E 38EBAFD8
9B3908BF 9B5A50B2 37539BF4 9D3256D9 B118DDF4 BC912AA1 B1E9DFF0 34729AE9
4B594142 B46D7C93 13FF997B 2FECC956 2362A8CC A0CD51EF 5691A2C3 9EB200FE
F4D341AE F35D3C06 8BCC1ACF 42E983FF F8C0B5A5 70906FCD 07F854D3 41CE9402
0572AE66 EF050203 010001A3 53305130 0F060355 1D130101 FF040530 030101FF
301F0603 551D2304 18301680 145C7DAD E37AB191 53C24775 8FC918B5 8059336C
12301D06 03551D0E 04160414 5C7DADE3 7AB19153 C247758F C918B580 59336C12
300D0609 2A864886 F70D0101 05050003 82010100 673243D9 3BBC0321 1FAC5459
926E99BF 60E55344 123B8A22 359B5DA8 E98E0A4F 5FDD49FC 5AF99F8B 87F30704
E74BEC68 DF4D2116 9DBD58D0 F4ABEE17 D9155CAE DBB7E94E 7A058507 CFA8DFB2
90E44C50 F95AD87F 934F904D 8C07CE47 5AEBBB7A EBA3E0C9 6CBA7B34 CC4642B6
DE641222 E045CEF4 27625FD2 FE51853C 574CCCA8 F036874B 93C97278 3D3776F1
E6419A07 46065203 FB81BFFD 1B2D5270 84FA9BAE CC06EE2A DF667257 DA97D96D
3E226378 28CE8460 2570D7D3 4D78C9E2 66FBA5B1 9A6E46AD E466D67F 425BFC40
FA717361 CBAA9AA0 7DB343F9 563B675B F1B6D193 12162EAA 6389A57C CF65AA08
53B07581 87A0C15A D5B6900B E3F98713 F3918F89
quit
!
!
no license feature hseck9
license udi pid IR1835-K9 sn FHH2416P00V
memory free low-watermark processor 47775
!
diagnostic bootup level minimal
!
spanning-tree extend system-id
!
!
redundancy
mode none
!
!
controller Gps-Dr
!
!
vlan internal allocation policy ascending
!
```

```
!
interface GigabitEthernet0/0/0
no ip address
shutdown
negotiation auto
!
interface GigabitEthernet0/1/0
shutdown
!
interface GigabitEthernet0/1/1
switchport mode access
!
interface GigabitEthernet0/1/2
shutdown
!
interface GigabitEthernet0/1/3
!
interface Wlan-GigabitEthernet0/1/4
!
interface Vlan1
no ip address
!
interface Async0/2/0
no ip address
encapsulation scada
!
interface Async0/2/1
no ip address
encapsulation scada
!
ip http server
ip http auth-retry 3 time-window 1
ip http authentication local
ip http secure-server
ip forward-protocol nd
ip dns server
ip nat inside source list 197 interface GigabitEthernet0/0/0 overload
!
!
!
ip access-list extended 197
10 permit ip any any
!
!
!
!
control-plane
!
!
!
mgcp behavior rsip-range tgcp-only
mgcp behavior comedia-role none
mgcp behavior comedia-check-media-src disable
mgcp behavior comedia-sdp-force disable
!
mgcp profile default
!
!
!
!
!
!
line con 0
stopbits 1
```

```
line 0/0/0 0/0/1
line 0/2/0 0/2/1
line vty 0 4
login
transport input all
transport output all
line vty 5 15
login
transport input all
transport output all
!
call-home
! If contact email address in call-home is configured as sch-smart-licensing@cisco.com
! the email address configured in Cisco Smart License Portal will be used as contact email
 address to send SCH notifications.
contact-email-addr sch-smart-licensing@cisco.com
profile "CiscoTAC-1"
active
destination transport-method http
!

end
```

Configuring Global Parameters

To configure global parameters for your router, follow these steps.

Procedure

	Command or Action	Purpose

Step 1	**configure terminal**
	Example:
	`Router> enable`
	`Router# configure terminal`
	`Router(config)#`
		Enters global configuration mode when using the console port.
		Use the following to connect to the router with a remote terminal:
		`telnet router-name or address`
		`Login: login-id`
		`Password: *********`
		`Router> enable`

Step 2	**hostname** *name* **Example:** `Router(config)# hostname Router`	Specifies the name for the router.
Step 3	**enable password** *password* or **enable secret password** *password* **Example:** `Router(config)# enable password cr1ny5ho`	Specifies a password to prevent unauthorized access to the router. **Note** In this form of the command, password is not encrypted. To encrypt the password use enable secret password as noted in the previously mentioned Device Hardening Guide.

Configuring the Gigabit Ethernet Interface

The default configuration for the Gigabit Ethernet Interface (GI0/0/0) on the IR1800 is Layer 3 (L3). The Gigabit Ethernet Interface on the IR1800 is a combo port, which means it is a RJ45+SFP connector. If you use an SFP as your interface, you need to set the media type for SFP.

```
Router(config-if)# media-type sfp
```

The correct connector must be selected, refer to the Hardware Installation Guide.

To manually define the Gigabit Ethernet interface, follow these steps, beginning from global configuration mode.

Procedure

	Command or Action	Purpose
Step 1	**configure terminal**	
Step 2	**ipv6 unicast-routing** **Example:** `Router# ipv6 unicast-routing`	Enables forwarding of IPv6 unicast data packets.
Step 3	**interface GigabitEthernet** *slot/bay/port* **Example:** `Router(config)# interface GigabitEthernet 0/0/0`	Enters the configuration mode for an interface on the router.
Step 4	**ip address** *ip-address mask* **Example:** `Router(config-if)# ip address 192.168.12.2 255.255.255.0`	Sets the IP address and subnet mask for the specified interface. Use this Step if you are configuring an IPv4 address.
Step 5	**ipv6 address** *ipv6-address/prefix* **Example:** `Router(config-if)# ipv6 address 2001.db8::ffff:1/128`	Sets the IPv6 address and prefix for the specified interface. Use this step instead of Step 2, if you are configuring an IPv6 address. IPv6 unicast-routing needs to be set-up as well, see further information in the **IPv6 Addressing and Basic Connectivity Configuration Guide** located here: https://www.cisco.com/c/en/us/td/docs/ios-xml/ios/ipv6_basic/configuration/xe-16-10/ip6b-xe-16-10-book/read-me-first.html
Step 6	**no shutdown** **Example:** `Router(config-if)# no shutdown`	Enables the interface and changes its state from administratively down to administratively up.

	Command or Action	Purpose
Step 7	**exit** **Example:** `Router(config-if)# exit`	Exits the configuration mode of interface and returns to the global configuration mode.

Support for sub-interface on GigabitEthernet0/0/0

Cisco IOS-XE supports sub-interfaces and dot1q configuration on the g0/0/0 interface. For example:

```
Router(config)#interface g0/0/0 ?
  <1-4294967295>  GigabitEthernet interface number
Router(config-subif)#encapsulation ?
  dot1Q           IEEE 802.1Q Virtual LAN
```

Configuring a Loopback Interface

Before you begin

The loopback interface acts as a placeholder for the static IP address and provides default routing information.

To configure a loopback interface, follow these steps.

Procedure

	Command or Action	Purpose
Step 1	**configure terminal**	
Step 2	**interface** *type number* **Example:** `Router(config)# interface Loopback 0`	Enters configuration mode on the loopback interface.
Step 3	(Option 1) **ip address** *ip-address mask* **Example:** `Router(config-if)# ip address 10.108.1.1 255.255.255.0`	Sets the IP address and subnet mask on the loopback interface. (If you are configuring an IPv6 address, use the **ipv6 address** *ipv6-address/prefix* command described below.
Step 4	(Option 2) **ipv6 address** *ipv6-address/prefix* **Example:** `Router(config-if)# ipv6 address 2001:db8::ffff:1/128`	Sets the IPv6 address and prefix on the loopback interface.

	Command or Action	Purpose
Step 5	**exit** **Example:** `Router(config-if)# exit`	Exits configuration mode for the loopback interface and returns to global configuration mode.

Example

Verifying Loopback Interface Configuration

Enter the **show interface loopback** command. You should see an output similar to the following example:

```
Router# show interface loopback 0
Loopback0 is up, line protocol is up
  Hardware is Loopback
  Internet address is 192.0.2.0/16
  MTU 1514 bytes, BW 8000000 Kbit, DLY 5000 usec,
     reliability 255/255, txload 1/255, rxload 1/255
  Encapsulation LOOPBACK, loopback not set
  Last input never, output never, output hang never
  Last clearing of "show interface" counters never
  Queueing strategy: fifo
  Output queue 0/0, 0 drops; input queue 0/75, 0 drops
  5 minute input rate 0 bits/sec, 0 packets/sec
  5 minute output rate 0 bits/sec, 0 packets/sec
     0 packets input, 0 bytes, 0 no buffer
     Received 0 broadcasts, 0 runts, 0 giants, 0 throttles
     0 input errors, 0 CRC, 0 frame, 0 overrun, 0 ignored, 0 abort
     0 packets output, 0 bytes, 0 underruns
     0 output errors, 0 collisions, 0 interface resets
     0 output buffer failures, 0 output buffers swapped out
```

Alternatively, use the **ping** command to verify the loopback interface, as shown in the following example:

```
Router# ping 192.0.2.0
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 192.0.2.0, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 1/2/4 ms
```

Enabling Cisco Discovery Protocol

Cisco Discovery Protocol (CDP) is enabled by default on the router. It may be disabled if needed for security purposes.

For more information on using CDP, see Cisco Discovery Protocol Configuration Guide, Cisco IOS XE Release 3S.

Configuring Command-Line Access

To configure parameters to control access to the router, follow these steps.



Note Transport input must be set as explained in the previous Telnet and SSH sections of the guide.

Procedure

	Command or Action	Purpose
Step 1	**line [aux | console | tty | vty]** *line-number* **Example:** `Router(config)# line console 0`	Enters line configuration mode, and specifies the type of line. The example provided here specifies a console terminal for access.
Step 2	**password** *password* **Example:** `Router(config-line)# password 5dr4Hepw3`	Specifies a unique password for the console terminal line.
Step 3	**login** **Example:** `Router(config-line)# login`	Enables password checking at terminal session login.

Step 4	**exec-timeout** *minutes* [*seconds*]
	Example:
	`Router(config-line)# exec-timeout 5 30`
	`Router(config-line)#`
		Sets the interval during which the EXEC command interpreter waits until user input is detected. The default is 10 minutes. Optionally, adds seconds to the interval value.
		The example provided here shows a timeout of 5 minutes and 30 seconds. Entering a timeout of **0 0** specifies never to time out.

Step 5	**exit** **Example:** `Router(config-line)# exit`	Exits line configuration mode to re-enter global configuration mode.

Step 6	**line [aux | console | tty | vty]** *line-number*
	Example:
	`Router(config)# line vty 0 4`
	`Router(config-line)#`
		Specifies a virtual terminal for remote console access.

	Command or Action	Purpose
Step 7	**password** *password* **Example:** `Router(config-line)# password aldf2ad1`	Specifies a unique password for the virtual terminal line.
Step 8	**login** **Example:** `Router(config-line)# login`	Enables password checking at the virtual terminal session login.
Step 9	**end** **Example:** `Router(config-line)# end`	Exits line configuration mode, and returns to privileged EXEC mode.

Example

The following configuration shows the command-line access commands. Note that transport input none is the default, but if SSH is enabled this must be set to ssh.

You do not have to input the commands marked **default**. These commands appear automatically in the configuration file that is generated when you use the **show running-config** command.

```
!
line console 0
exec-timeout 10 0
password 4youreyesonly
login
transport input none (default)
stopbits 1 (default)
line vty 0 4
password secret
login
!
```

Configuring Static Routes

Static routes provide fixed routing paths through the network. They are manually configured on the router. If the network topology changes, the static route must be updated with a new route. Static routes are private routes unless they are redistributed by a routing protocol.

To configure static routes, follow these steps.

Procedure

	Command or Action	Purpose
Step 1	(Option 1) **ip route** *prefix mask* {*ip-address* | *interface-type interface-number* [*ip-address*]} **Example:** `Router(config)# ip route 192.10.2.3 255.255.0.0 10.10.10.2`	Specifies a static route for the IP packets. (If you are configuring an IPv6 address, use the **ipv6 route** command described below.)
Step 2	(Option 2) **ipv6 route** *prefix/mask* {*ipv6-address* | *interface-type interface-number* [*ipv6-address*]} **Example:** `Router(config)# ipv6 route 2001:db8:2::/64 2001:db8:3::0`	Specifies a static route for the IP packets. See additional information for IPv6 here: https://www.cisco.com/c/en/us/td/docs/ios-xml/ios/ipv6_basic/configuration/xe-16-10/ip6b-xe-16-10-book/read-me-first.html
Step 3	**end** **Example:** `Router(config)# end`	Exits global configuration mode and enters privileged EXEC mode.

In the following configuration example, the static route sends out all IP packets with a destination IP address of 192.168.1.0 and a subnet mask of 255.255.255.0 on the Gigabit Ethernet interface to another device with an IP address of 10.10.10.2. Specifically, the packets are sent to the configured PVC.

You do not have to enter the command marked **default**. This command appears automatically in the configuration file generated when you use the **show running-config** command.

```
!
ip classless (default)
ip route 2001:db8:2::/64 2001:db8:3::0
```

Verifying Configuration

To verify that you have configured static routing correctly, enter the **show ip route** command (or **show ipv6 route** command) and look for static routes marked with the letter S.

When you use an IPv4 address, you should see verification output similar to the following:

```
Router# show ip route
Codes: C - connected, S - static, R - RIP, M - mobile, B - BGP
       D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
       E1 - OSPF external type 1, E2 - OSPF external type 2
       i - IS-IS, su - IS-IS summary, L1 - IS-IS level-1, L2 - IS-IS level-2
       ia - IS-IS inter area, * - candidate default, U - per-user static route
       o - ODR, P - periodic downloaded static route

Gateway of last resort is not set
```

```
     10.0.0.0/24 is subnetted, 1 subnets
C       10.108.1.0 is directly connected, Loopback0
S*   0.0.0.0/0 is directly connected, GigabitEthernet0
```

When you use an IPv6 address, you should see verification output similar to the following:

```
Router# show ipv6 route
IPv6 Routing Table - default - 5 entries
Codes: C - Connected, L - Local, S - Static, U - Per-user Static route
       B - BGP, R - RIP, H - NHRP, I1 - ISIS L1
       I2 - ISIS L2, IA - ISIS interarea, IS - ISIS summary, D - EIGRP
       EX - EIGRP external, ND - ND Default, NDp - ND Prefix, DCE -
Destination
       NDr - Redirect, O - OSPF Intra, OI - OSPF Inter, OE1 - OSPF ext 1
       OE2 - OSPF ext 2, ON1 - OSPF NSSA ext 1, ON2 - OSPF NSSA ext 2
       ls - LISP site, ld - LISP dyn-EID, a - Application

C   2001:DB8:3::/64 [0/0]
       via GigabitEthernet0/0/2, directly connected
S   2001:DB8:2::/64 [1/0]
       via 2001:DB8:3::1
```

Configuring Dynamic Routes

In dynamic routing, the network protocol adjusts the path automatically, based on network traffic or topology. Changes in dynamic routes are shared with other routers in the network.

All of the Cisco IOS-XE configuration guides can be found here: https://www.cisco.com/c/en/us/support/ios-nx-os-software/ios-xe-amsterdam-17-3-1/model.html

Configuring Routing Information Protocol

To configure the RIP on a router, follow these steps.

Procedure

	Command or Action	Purpose
Step 1	**router rip** **Example:** `Router(config)# router rip`	Enters router configuration mode, and enables RIP on the router.
Step 2	**version** {**1** | **2**} **Example:** `Router(config-router)# version 2`	Specifies use of RIP version 1 or 2.

	Command or Action	Purpose

Step 3	**network** *ip-address*
	Example:
	`Router(config-router)# network 192.168.1.1`
	`Router(config-router)# network 10.10.7.1`
		Specifies a list of networks on which RIP is to be applied, using the address of the network of each directly connected network.

Step 4	**no auto-summary** **Example:** `Router(config-router)# no auto-summary`	Disables automatic summarization of subnet routes into network-level routes. This allows subprefix routing information to pass across classful network boundaries.
Step 5	**end** **Example:** `Router(config-router)# end`	Exits router configuration mode, and enters privileged EXEC mode.

Example

Verifying Configuration

To verify that you have configured RIP correctly, enter the **show ip route** command and look for RIP routes marked with the letter R. You should see an output similar to the one shown in the following example:

```
Router# show ip route
Codes: C - connected, S - static, R - RIP, M - mobile, B - BGP
       D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
       E1 - OSPF external type 1, E2 - OSPF external type 2
       i - IS-IS, su - IS-IS summary, L1 - IS-IS level-1, L2 - IS-IS level-2
       ia - IS-IS inter area, * - candidate default, U - per-user static route
       o - ODR, P - periodic downloaded static route

Gateway of last resort is not set

     10.0.0.0/24 is subnetted, 1 subnets
C       10.108.1.0 is directly connected, Loopback0
R    3.0.0.0/8 [120/1] via 2.2.2.1, 00:00:02, Ethernet0/0/0
```

Configuring Enhanced Interior Gateway Routing Protocol

The Enhanced Interior Gateway Routing Protocol (EIGRP) is an enhanced version of the Interior Gateway Routing Protocol (IGRP) developed by Cisco. The convergence properties and the operating efficiency of EIGRP have improved substantially over IGRP, and IGRP is now obsolete.

The convergence technology of EIGRP is based on an algorithm called the Diffusing Update Algorithm (DUAL). The algorithm guarantees loop-free operation at every instant throughout a route computation and allows all devices involved in a topology change to synchronize. Devices that are not affected by topology changes are not involved in recomputations

Details on configuring Enhanced Interior Gateway Routing Protocol (EIGRP), are found in the following guide: https://www.cisco.com/c/en/us/td/docs/ios-xml/ios/iproute_eigrp/configuration/xe-16-10/ire-xe-16-10-book/ire-enhanced-igrp.html

Modular QoS (MQC)

This section provides an overview of Modular QoS CLI (MQC), which is how all QoS features are configured on the IoT Integrated Services Router. MQC is a standardized approach to enabling QoS on Cisco routing and switching platforms.

Follow the procedures that are in the QoS Modular QoS Command-Line Interface Configuration Guide, Cisco IOS XE 17 guide.

Configuring the Serial Interface

This section describes configuring serial interface management.

The IR1800 supports asynchronous serial interface protocols used for SCADA, Raw Socket, or Encapsulation Relay. Depending on the product type, the router has one or two serial interfaces.

Table 1: Naming Conventions

Serial Interface	Line Number	Internal Mapping
Async 0/2/0 (DTE)	Line 0/2/0 (50)	ttyS1
Async 0/2/1 (DCE)	Line 0/2/1 (51)	ttyUSB0

IR1821

The IR1821 has only a single Async serial port.

IR1831

The IR1831 has two ports, a DTE and DCE port with RS232 only.

IR1833

The IR1833 has two ports, a DTE and DCE port with RS232 only.

IR1835

The IR1835 has two ports, a DCE with RS232/RS485 and DTE port with RS232. With media-type RS485, there is support for both half and duplex settings.

Note Async serial cabling is documented in the IR1800 Hardware Installation Guide.

Specifying an Asynchronous Serial Interface

To specify an asynchronous serial interface and enter interface configuration mode, use one of the following commands in global configuration mode.

Command or Action	Purpose
Router(config)# interface async 0/2/0	Enters interface configuration mode.

Specifying Asynchronous Serial Encapsulation

The two serial interfaces will be marked as async 0/2/0 and 0/2/1. The bay number for async is 2.

The asynchronous serial interfaces support the following serial encapsulation methods:

- Raw Socket
- Line Relay
- SCADA protocol translation

Command or Action	Purpose
Router(config-if)# **encapsulation** *{raw-tcp|raw-udp|scada|relay-line}*	Configures asynchronous serial encapsulation.

Encapsulation methods are set according to the type of protocol or application you configure in the Cisco IOS software.

The remaining encapsulation methods are defined in their respective books and chapters describing the protocols or applications.

Configuring the Serial Port

The IR1835 Pro Device has RS232/RS485 combo DCE port Async 0/2/1. The remaining devices in the IR1800 series only support RS232 media-type.

Table 2: Configuration Examples

`#sh run int Async 0/2/1`
`Building configuration...`
`Current configuration : 95 bytes`
`!`
`interface Async0/2/1`
`  no ip address`
`  encapsulation scada`
`  media-type rs485`
`  full-duplex`
`end`
	DCE Port with media-type RS485

#**sh run int Async 0/2/1** Building configuration... Current configuration : 64 bytes ! interface Async0/2/1 no ip address encapsulation raw-tcp end	DCE Port with media-type RS232 [Default Configuration]
sh run int Async 0/2/0 Building configuration... Current configuration : 64 bytes ! interface Async0/2/0 no ip address encapsulation relay-line end	DTE Port with default media-type RS232. (RS485 not supported).

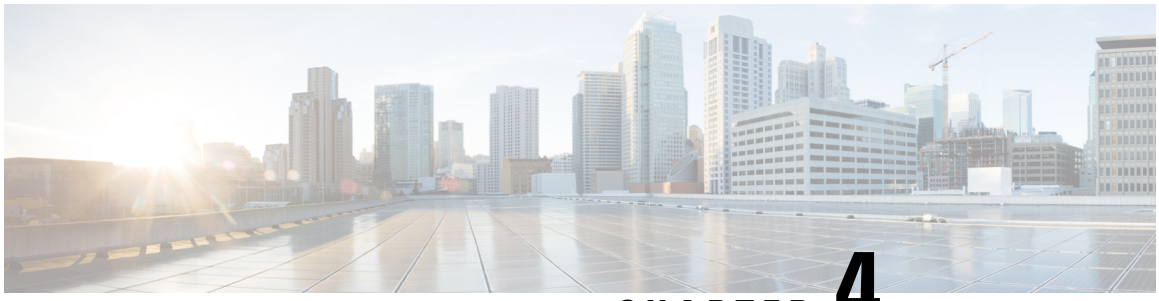
The following configuration example is for media-type RS485 and RS232.

```
IR1800#sh run int async 0/2/1
Building configuration...
Current configuration : 100 bytes
!
interface Async0/2/1
no ip address
encapsulation relay-line
media-type rs485
half-duplex
..................
IR1800#sh run int async 0/2/0
Building configuration...

Current configuration : 64 bytes
!
interface Async0/2/0
no ip address
encapsulation scada
end
```

Line(s) not in async mode -or- with no hardware support:

Tty Line Typ Tx/Rx A Modem Roty AccO AccI Uses Noise Overruns Int1, 4-49, 52-73, 89-735



CHAPTER 4

Web User Interface

This chapter contains the following sections:

- Introduction to the Web User Interface, on page 37
- Day 0 Cellular Mode, on page 38
- Day 0 Web User Interface, on page 38
- Additional Modem Support for CAT 6 and CAT 7 Cellular Pluggable Modules, on page 38
- Additional Modem Support for Cellular Pluggable Modules, on page 39
- Galileo support on the LTE pluggable modules, on page 40
- GPS mode default enablement, on page 41
- Guidelines and Limitations, on page 42
- Configuring Your Computer to Connect to the Router, on page 42
- Connecting to the Router Using DHCP, on page 43
- Configuring Basic Mode WebUI through the Browser, on page 45
- Configuring Advanced Mode WebUI through the Browser, on page 50
- WebUI Dashboard, on page 56
- Configure access point name using Cisco WebUI, on page 56

Introduction to the Web User Interface

The Web User Interface (WebUI) provides network administrators with a single solution for provisioning, monitoring, and optimizing devices. After you complete the hardware installation, you need to setup the device with a configuration required to enable traffic to pass through the network. On your first day with your new device, you can perform a number of tasks to ensure that your device is online, reachable and easily configured. This is referred to as the Day 0 interface.

Note A Day 0 configuration is defined as a device that is fresh out of the box with no startup-configuration.

After the initial Day 0 configuration, the WebUI can be used for day to day configuration.

Once the router boots up in Day 0, the PC can connect to the 192.168.1.x network and can access WebUI using the IP address of 192.168.1.1 with any browser. After the configuration is applied through the WebUI, the router will display the message "Day 0 config done. Stopping autoinstall".

Day 0 Cellular Mode

Cisco IOS XE release 17.9.1 provides new functionality allowing the router to be configured on Day 0 through the cellular pluggable module. This assumes a cellular pluggable module is already installed.

This mode helps configuring the Cellular APN, assuming the customer gets a private APN (or private LTE/5G) as WAN backhaul. By doing so, the APN value is stored in the modem. Once the router reboots, it is reset to factory-default, enabling the router to perform PnP over Cellular when private APN is used.

Note Advanced Mode is needed in order to set up Cellular WAN, including public or private APN. This should be provided by your SIM's service provider.

Note The pluggable interface is not hot swappable. If you wish to change a SIM, power off the router.

The steps to configure through the cellular pluggable module follow:

1. Select the Cellular interface in the **`WAN type`**.
2. Enter the APN name.
3. There is no need to select a backup WAN.
4. Reboot the router.

PnP will now be able to run with private APN to connect to IOS OD, vManage, or DNA-C.

Day 0 Web User Interface

Starting from IOS XE Release 17.1.1, the Day 0 Web User Interface (WebUI) will be supported on the IR1101. Day 0 WebUI is supported only on LAN ports. These are FastEthernet ports 0/0/1 – 0/0/4 on the IR1101. Connect either a Windows, Linux or Mac PC/Laptop to one of the LAN ports of the IR1101 and boot the router on Day 0. The PC/Laptop should be configured to obtain an IP address through DHCP.

Additional Modem Support for CAT 6 and CAT 7 Cellular Pluggable Modules

The LTE Cat6 Pluggable Interface Modules (PIMs) will be updated with Cat7 modems. The following table shows the product transition:

Table 3: Cat6 to Cat7 Transition

Cat6 (Current)	Cat7 (Refreshed)
Sierra Wireless EM7455/7430	Sierra Wireless EM7411/7421/7431
Cat6 LTE Advanced	Cat7 LTE Advanced

Support for additional modems on the IR1101

The following are the new PIDs that will be available:

- P-LTEA7-NA
- P-LTEA7-EAL
- P-LTEA7-JP
- P-5GS6-R16SA

Support for additional modems on the IR1800

The following are the new PIDs that will be available:

- P-LTEA7-NA
- P-LTEA7-EAL
- P-LTEA7-JP

Important For the new PIDs mentioned above, the following cellular functions have not been tested, and are not supported with IOS XE release 17.13.1 although the CLI commands may permit:

- GNSS/NMEA
- Cellular Dying-Gasp
- eSIM/eUICC support

Note There is no new or changed command line interface with these new modems.

Additional Modem Support for Cellular Pluggable Modules

Cisco IOS-XE Release 17.14.1 enhances connectivity options and throughput on the IR1101 and IR1800 platforms by supporting additional cellular modems:

- CAT 7 Modems:
  - P-LTEA7-NA

- P-LTEA7-EAL
- P-LTEA7-JP

- 5G Modem:
  - P-5GS6-R16SA-GL

Note CAT 7 modems support GNSS and NMEA streaming, while currently P-5GS6-R16SA-GL module does not support GPS and NMEA streaming.

Galileo support on the LTE pluggable modules

The Cisco IOS XE 17.11.1a release introduces support for the Galileo constellation on LTE pluggable modules.

GPS was the only GNSS constellation supported in earlier releases. The Cisco IOS XE 17.11.1a release introduces support for Galileo constellation.

Note You can enable only one constellation at a time.

Configuration command examples:

```
config# controller cellular<slot/port>
(config-controller)# <no> lte gps constellation <gps | galileo | gnss >
```

Example:

```
(config-controller)# lte gps constellation ?
galileo  select Galileo as active constellation
gps      select GPS as active constellation
gnss      select multiple GNSS as active constellation
```

Note The default setting is GPS mode.

The galileo and gnss options in this CLI configure Galileo and simultaneous GNSS modes, such as GPS and Galileo.

If you disable the GPS configuration, make sure that no constellation is configured. This maintains consistency with GPS mode configuration.

Example:

```
config#


controller Cellular 0/4/0


(config-controller)# no lte gps constellation gps
```

Show commands example:

Note The example shows the current GNSS constellation as Galileo.

Example:

```
config#

show cellular 0/4/0 gps detail

GPS Feature =  enabled
GPS Mode Configured =  standalone
Current Constellation Configured =  galileo
GPS Port Selected =  Dedicated GPS port
GPS Status =  GPS acquiring
```

Any changes made to the configuration will require the router to be rebooted.

More information is available in the Cellular Pluggable Interface Module Configuration Guide .

GPS mode default enablement

GPS mode default enablement is a configuration enhancement that:

- automatically enables GPS mode by default starting with IOS XE 17.9.1,
- sets GPS to standalone mode without requiring additional user configuration, and
- reduces cellular link up time by eliminating the need for manual modem power-cycling.

GPS mode configuration changes

The GPS mode configuration behavior varies between IOS XE versions:

- **Prior to IOS XE 17.9.1**: GPS was enabled by default, but GPS Mode was disabled by default, requiring users to perform an additional modem power-cycle after the router came up to use GPS.
- **Starting with IOS XE 17.9.1**: GPS Mode is enabled by default and set to standalone mode, helping reduce cellular link up time.

Note This only applies to the cellular based GPS. This does not apply to the GPS/GNSS module in IR1800 (DR module), IR8140 (native GPS) and IR8340 (Timing module).

Use the following command to check cellular GPS status:

```
Router# show cellular <slot> gps
auto-reset Enable reset modem automatically after configuring GPS enable or mode
```

Guidelines and Limitations

The following are Guidelines and Limitations for the IR1101 and the IR1800:

IR1101

Effective with IOS-XE Release 17.3.1, the Day 0 Web User Interface (WebUI) will be supported on the IR1101. Day 0 WebUI is supported only on LAN ports. These are FastEthernet ports 0/0/1 – 0/0/4 on the IR1101. Connect a PC to one of the LAN ports of the IR1101 and boot the router on Day 0. The PC can be configured to use DHCP or with a static IP address of 192.168.1.2/255.255.255.0.

The following are limitations to the Day 0 feature:

- The WebUI is not supported on the 1G port because this interface is dedicated to PnP. It is only supported on the 100M ports 1-4.
- Plug and Play (PNP) cannot be used if router is being used to configure using Day 0 WebUI as PNP will be aborted once the configuration is applied through Day 0 WebUI.
- Starting from release 17.1.2, an explicit **write memory** is not needed once the configuration is applied through the WebUI.

IR1800

The Day 0 Web User Interface (WebUI) is supported on the IR1800. Day 0 WebUI is supported only on LAN ports. These are GigabitEthernet ports 0/1/0 – 0/1/3 on the IR1800. Connect a PC to one of the LAN ports of the IR1800 and boot the router on Day 0. The PC can be configured to use DHCP or with a static IP address of 192.168.1.2/255.255.255.0.

The following are limitations to the Day 0 feature:

- The WebUI is not supported on the GigabitEthernet 0/0/0 port. It is only supported on the LAN ports GigabitEthernet0/1/0 through GigabitEthernet0/1/3.
- Plug and Play (PNP) cannot be used if router is being used to configure using Day 0 WebUI as PNP will be aborted once the configuration is applied through Day 0 WebUI.

Configuring Your Computer to Connect to the Router

The following section provides guidance for configuring your computer to properly interface with the IR1101.

You can access the application from a client web browser. Ensure that the following web client requirements are met:

- Hardware—A Mac (OS version 10.9.5) or Windows (OS version 10) laptop or desktop compatible with one of the following tested and supported browsers:
  - Google Chrome 59 or later
  - Mozilla Firefox 54 or later
  - Apple Safari 10 or later

- Microsoft Edge browser

- Display resolution—We recommend that you set the screen resolution to 1280 x 800 or higher.

Connecting to the Router Using DHCP

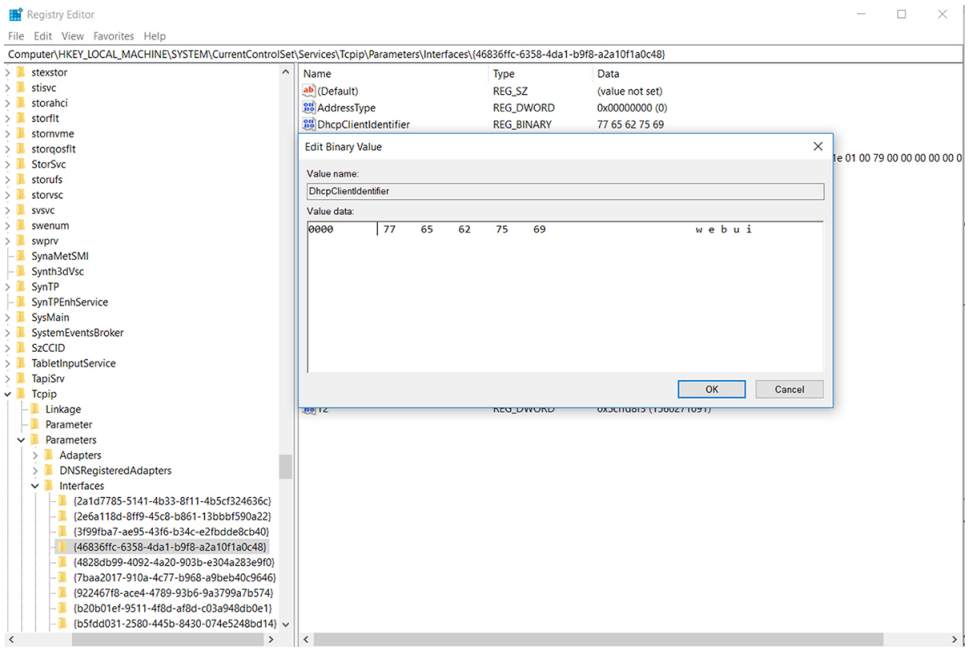
Set up the DHCP Client Identifier on the client to get the IP address from the router, and to be able to authenticate with Day 0 login credentials.

Setting up the DHCP Client Identifier on the client for Windows

1. Type **regedit** in the Windows search box on the taskbar and press **enter**.
2. If prompted by User Account Control, click **Yes** to open the Registry Editor.
3. Navigate to

   Computer\HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\Tcpip\Parameters\Interfaces and locate the **Ethernet Interface Global Unique Identifier (GUID)**.
4. Add a new REG_BINARY **DhcpClientIdentifier** with Data **77 65 62 75 69** for **`webui`**. You need to manually type in the value.

Figure 1: Setting up DHCP Client Identifier on Windows

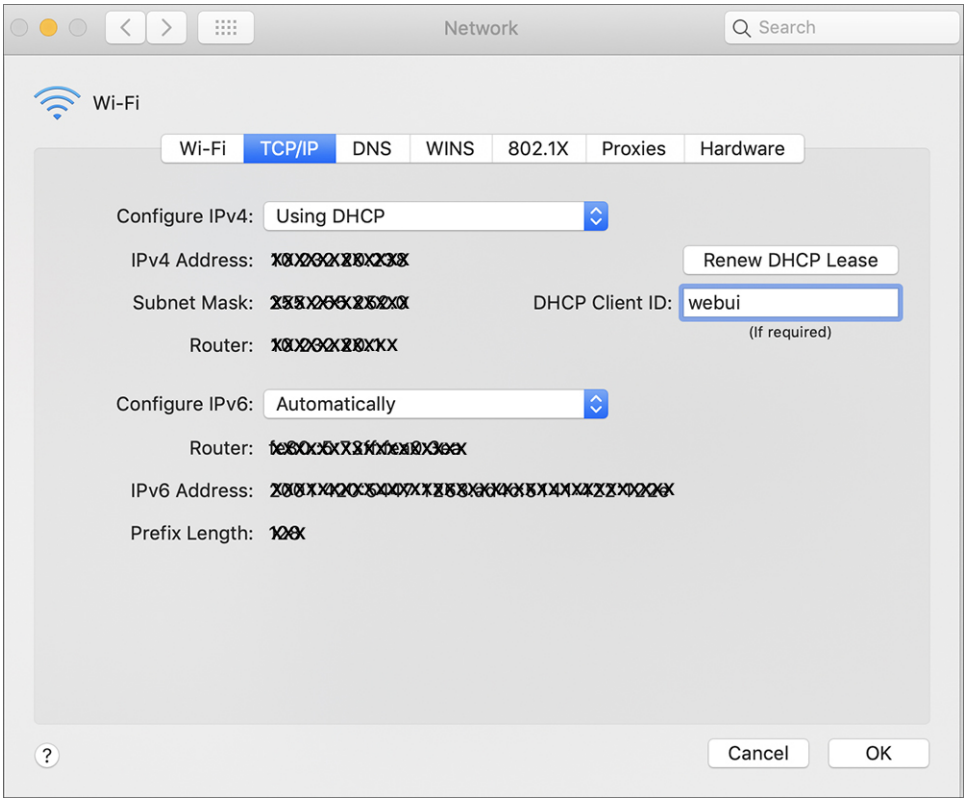


5. Restart the PC for the configuration to take effect.

Setting up the DHCP Client Identifier on the client for MAC

1. Go to **System Preferences >Network >Advanced >TCP >DHCP Client ID:** and enter **webui**.

Figure 2: Setting up DHCP Client Identifier on MAC



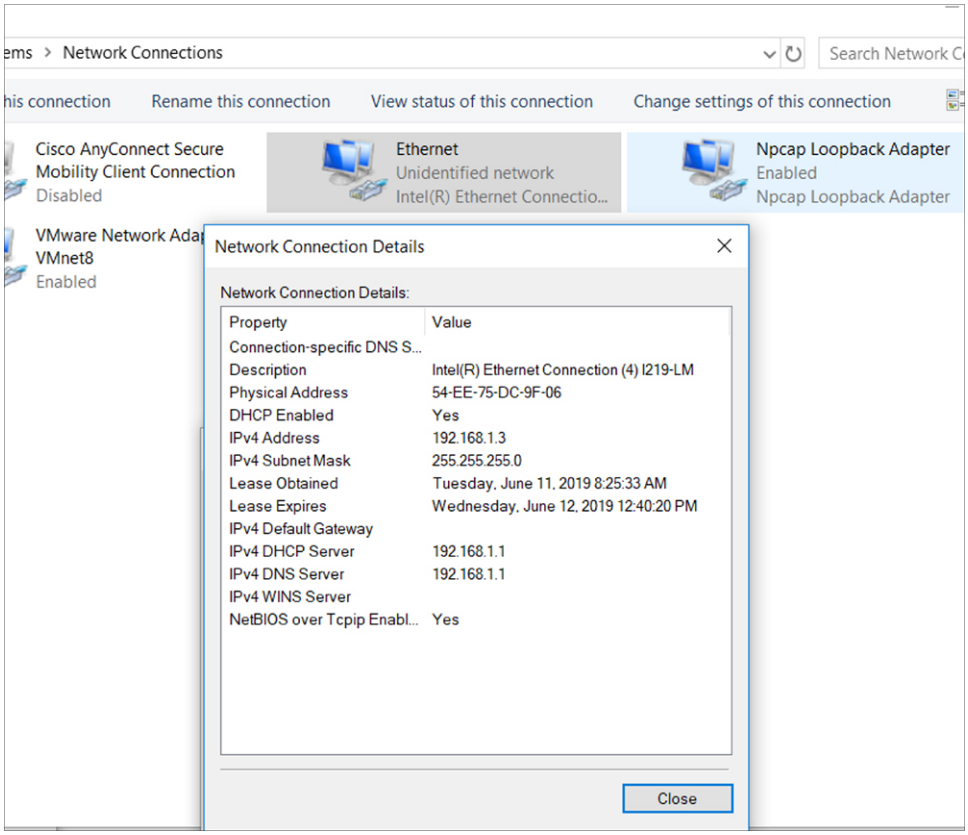
2. Click **OK** to save the changes.

Continuing with the Configuration Wizard

The bootup script runs the configuration wizard, which prompts you for basic configuration input: (**Would you like to enter the initial configuration dialog? [yes/no]:**). To configure Day 0 settings using the web UI, do not enter a response. Perform the following tasks instead:

1. Make sure that no devices are connected to the router.
2. Connect one end of an ethernet cable to one of the downlink (non-management) ports on the active supervisor and the other end of the ethernet cable to the host (PC/MAC).
3. Set up your PC/MAC as a DHCP client, to obtain the IP address of the router automatically. You should get an IP address within the 192.168.1.x/24 range.

Figure 3: Obtaining the IP Address



It may take up to three mins. You must complete the Day 0 setup through the web UI before using the router terminal.

4. Launch a web browser on the PC and enter the router IP address (`https://192.168.1.1`) in the address bar.
5. Enter the Day 0 username **webui** and password **cisco**.

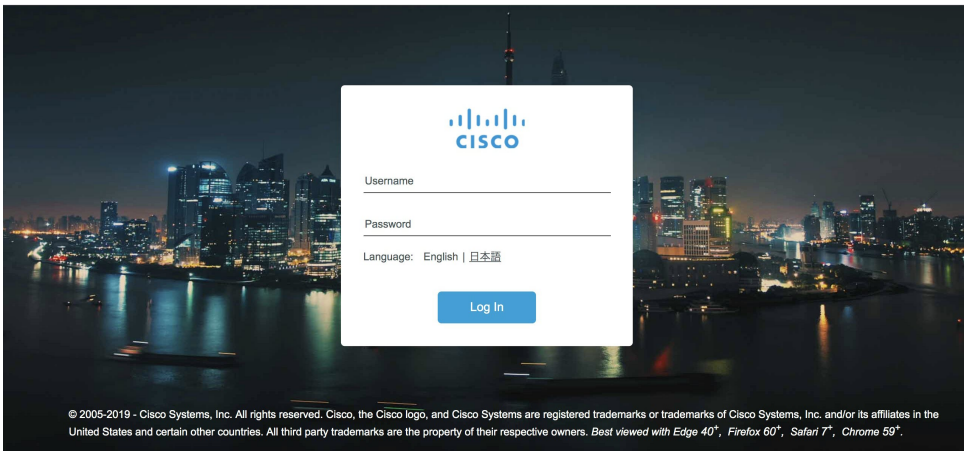
Configuring Basic Mode WebUI through the Browser

The following steps guide you through the process of using the browser on your PC/laptop to configure the WebUI.

Procedure

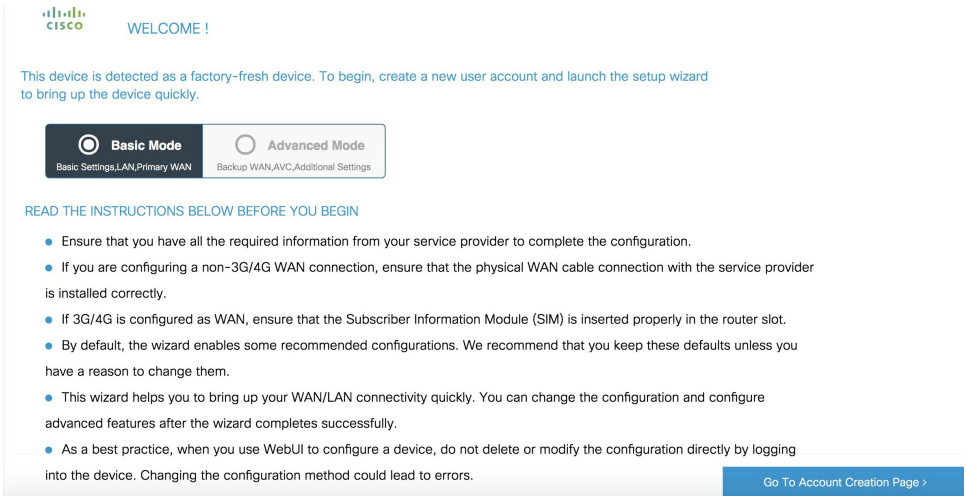
Step 1 Open your browser and enter 192.168.1.1in the address bar. The Login Screen appears. Enter the Username **webui** and the Password **cisco**. Then click **Log In**.

Figure 4: Login Screen



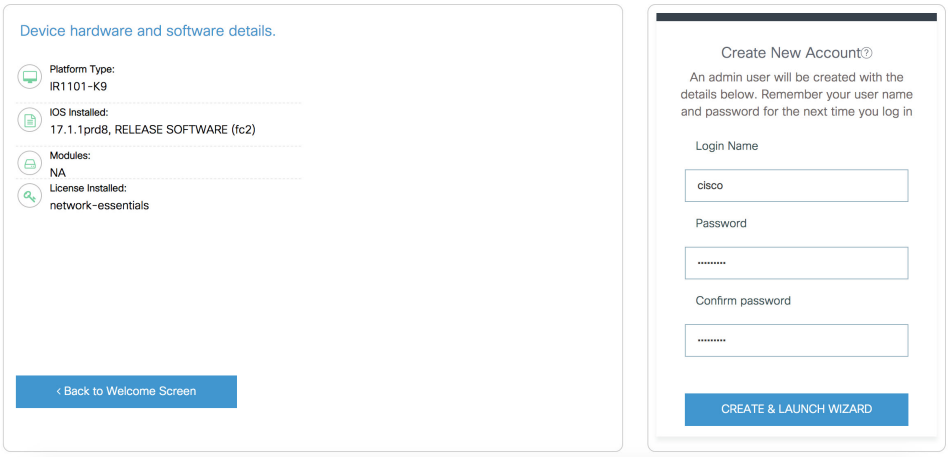
Step 2 The Welcome Screen appears. Select Advanced Mode or Basic Mode. Basic Mode allows for configuring Basic settings, LAN, and a Primary WAN. Advanced Mode allows you to configure an additional Backup WAN, AVC, as well as additional settings. For the purposes of this section, Basic Mode is used. Select **Basic Mode**.

Figure 5: Welcome Screen



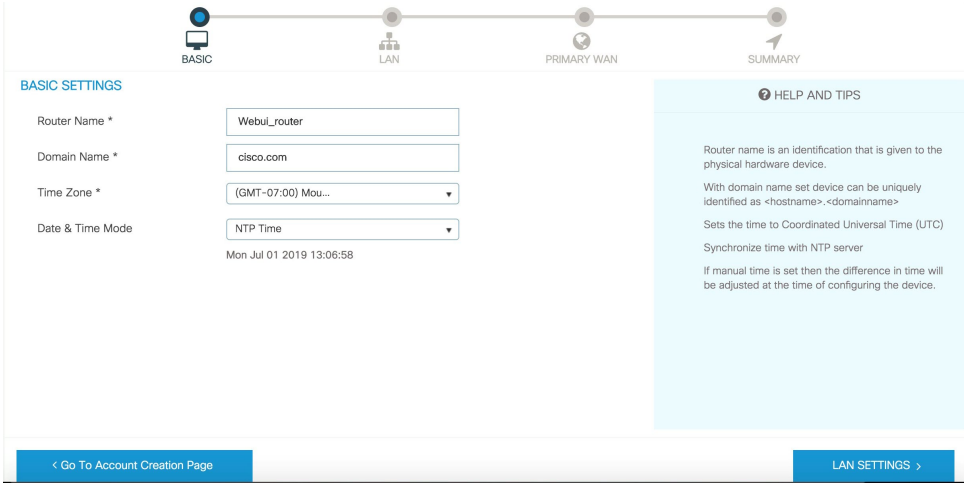
Step 3 Click **Go To Account Creation Page**. The Create New Account Screen appears. Create a new Login Name and Password to access the WebUI.

Figure 6: Create New Account Screen



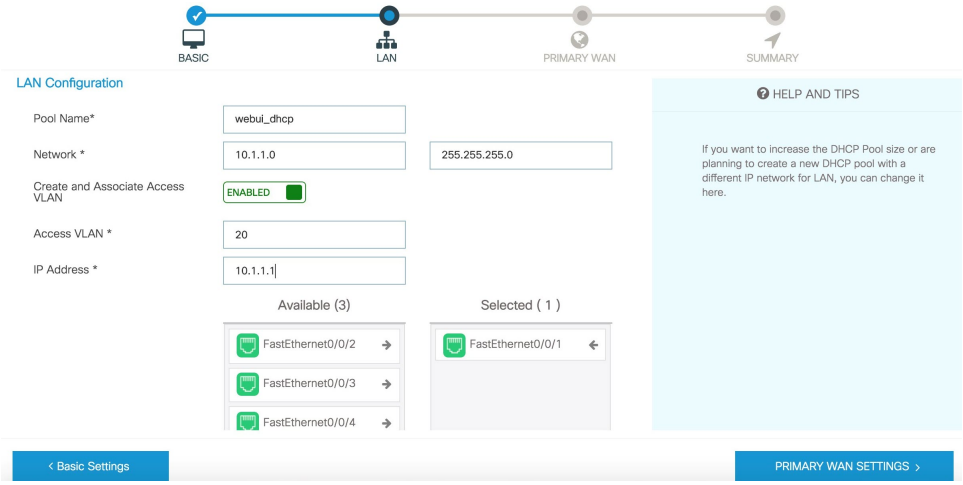
Step 4 Click **CREATE & LAUNCH WIZARD**. The Basic Settings Screen appears. Provide a Router Name (hostname), Domain Name, Time Zone and Date & Time Mode.

Figure 7: BASIC SETTINGS Screen



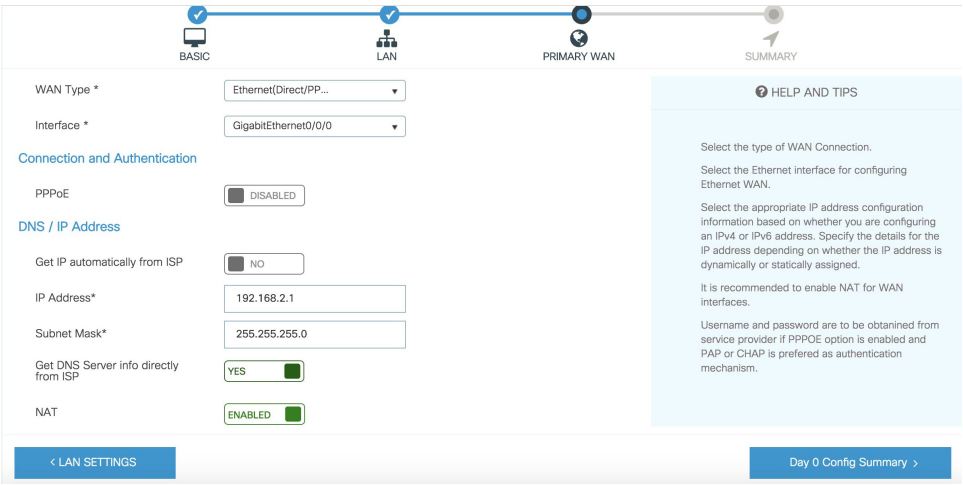
Step 5 Click **LAN SETTINGS**. The LAN Configuration Screen appears. Enter the webui_dhcp Pool Name, VLAN interface IP address, and select the interface that is connected to your laptop from the list of available interfaces.

Figure 8: LAN Configuration Screen



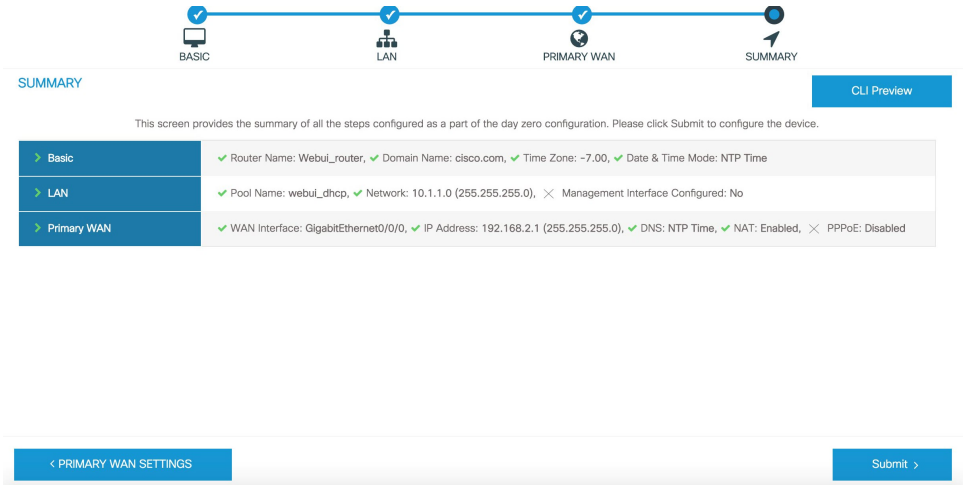
Step 6 Click **PRIMARY WAN SETTINGS**. The PRIMARY WAN SETTINGS Screen appears. Configure the WAN interface by selecting the WAN Type and Interface from the available options. Next enter your DNS IP address information and select Enable/Disable NAT.

Figure 9: Primary WAN Interface Screen



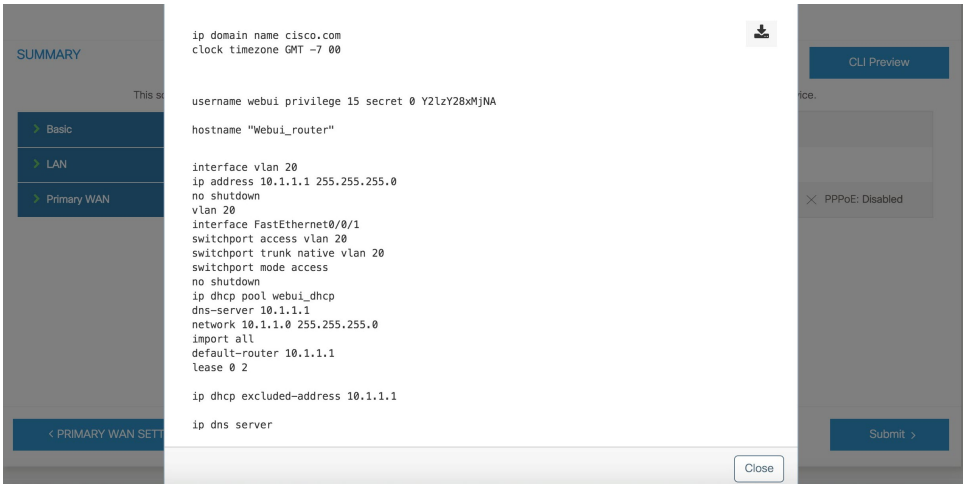
Step 7 Click **Day 0 Config Summary**. The Review Summary Screen appears. Verify your entries before applying the configuration.

Figure 10: Summary Screen



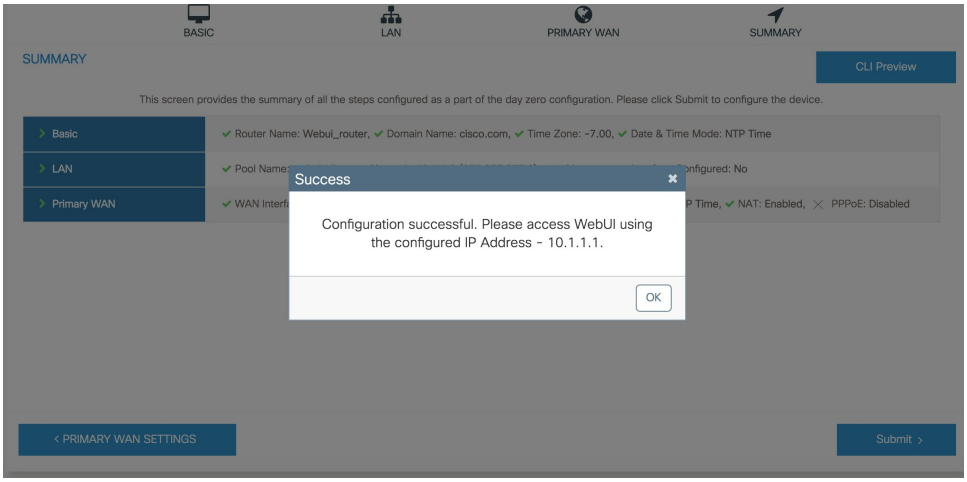
Step 8 (Optional) You can click on **CLI Preview** to see the Configuration that is being applied to the router. Close the CLI Preview and if you are ready, Click **Submit**.

Figure 11: CLI Preview Screen



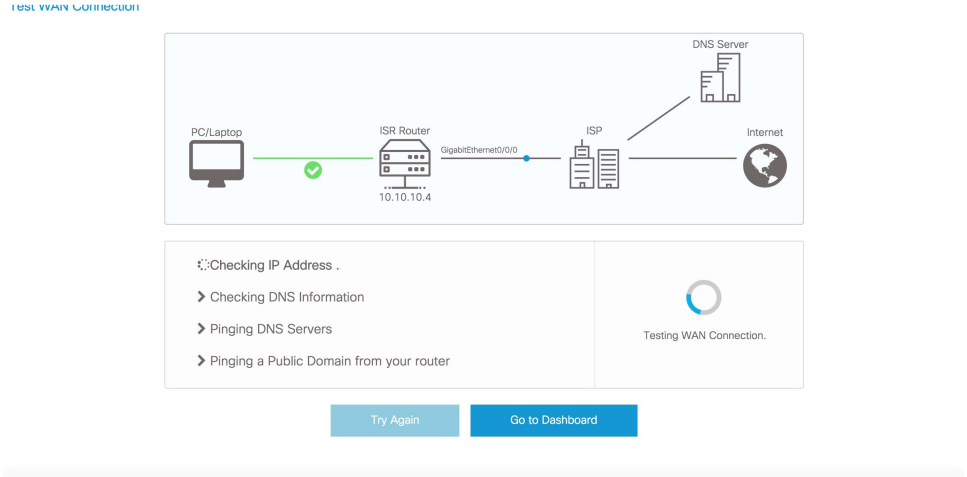
Step 9 After clicking on **Submit**, a dialog box will appear which informs you that the configuration has been applied successfully. The new WebUI ip address is also presented.

Figure 12: Submit Dialog Box



Step 10 If you have web connectivity, the device will try to connect. It is recommended that you close the browser session and move to the newly configured WebUI ip address.

Figure 13: Test VLAN Connection Screen



Configuring Advanced Mode WebUI through the Browser

The following steps guide you through the process of using the browser on your PC to configure the WebUI.

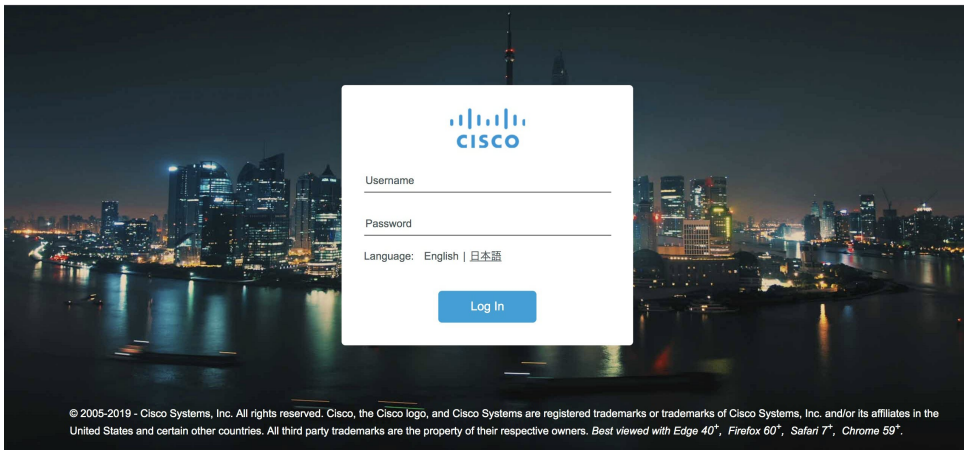
Make sure your laptop is configured to obtain an IP address through DHCP, or assign an IP address *n.n.n.n* matching the default subnet.

Note Advanced Mode is needed in order to set up Cellular WAN, including public or private APN.

Procedure

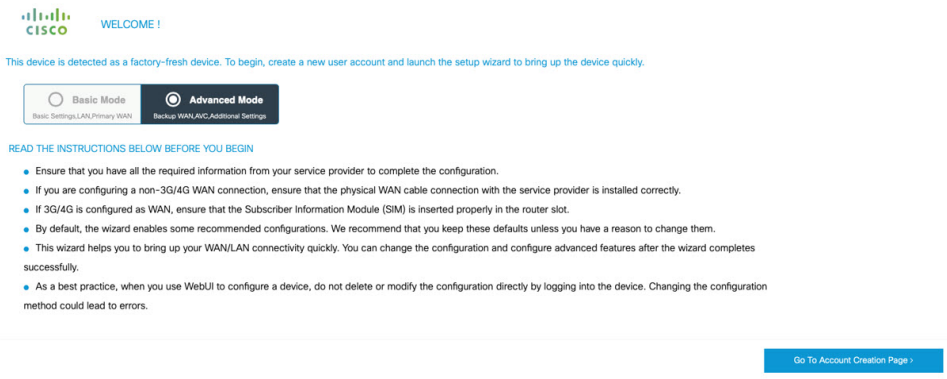
Step 1 Open your browser and enter 192.168.1.1 in the address bar. The Login Screen appears. Enter the Username **webui** and the Password **cisco**. Then click **Log In**.

Figure 14: Login Screen



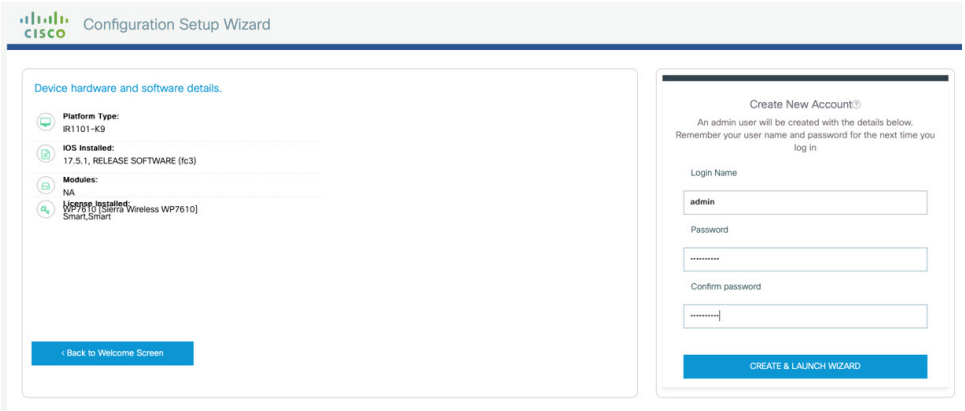
Step 2 The WELCOME screen appears. Select Advanced Mode or Basic Mode. Basic Mode allows for configuring Basic settings, LAN, and a Primary WAN. Advanced Mode allows you to configure an additional Backup WAN, AVC, as well as additional settings. For the purposes of this section, Advanced Mode is used.

Figure 15: WELCOME Screen



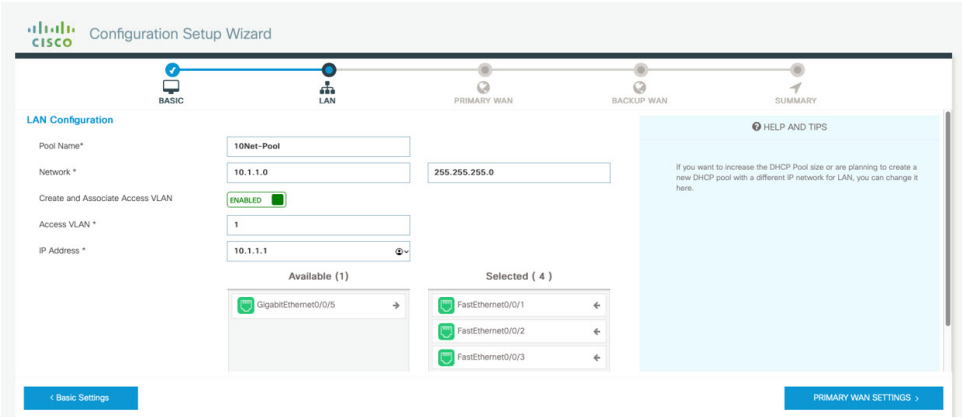
Step 3 Select **Advanced Mode**, then click **Go To Account Creation Page**. The Create New Account screen appears. Create a new Login Name and Password to access the WebUI.

Figure 16: Create New Account Screen



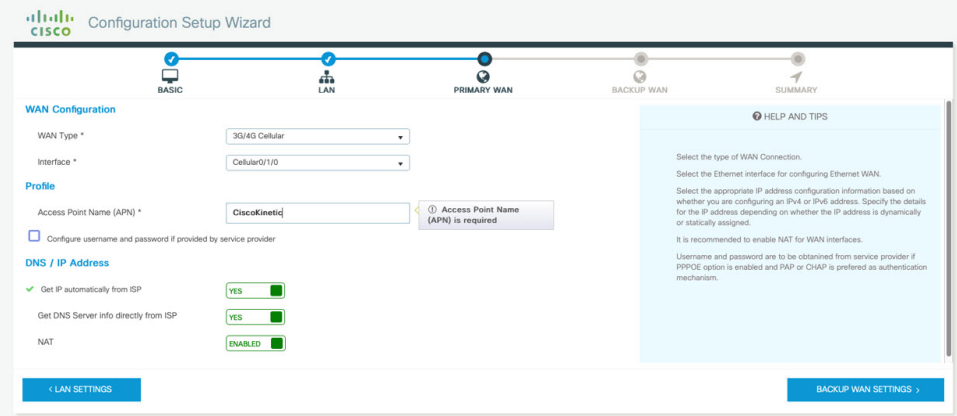
Step 4 Click **CREATE & LAUNCH WIZARD** The LAN Configuration screen appears. Provide a Pool Name, Network IP Address, Subnet, Access VLAN, and Device IP Address. A list of available interfaces is shown to select from.

Figure 17: LAN Configuration Screen



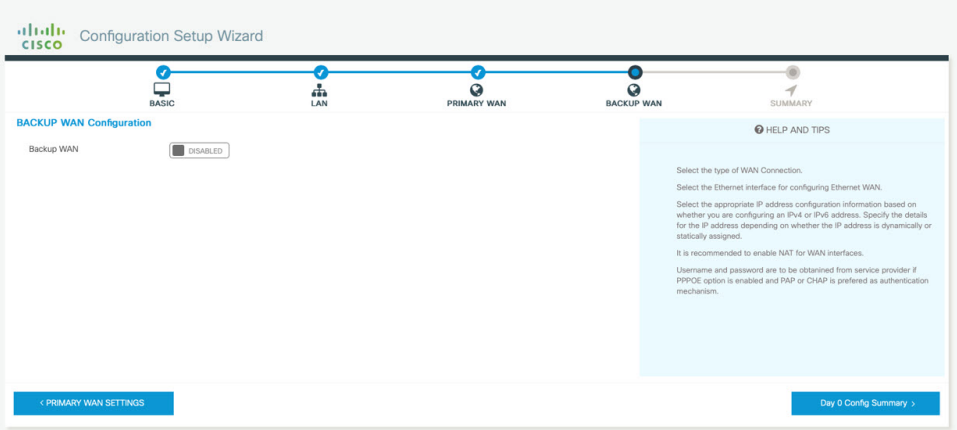
Step 5 Click **PRIMARY WAN SETTINGS**. The WAN Configuration screen appears. Select the WAN Type and Interface from the pull-downs. Provide an APN (Access Point Name) from your LTE Service Provider, and then select the DNS and IP Address settings for your network.

Figure 18: WAN Configuration Screen



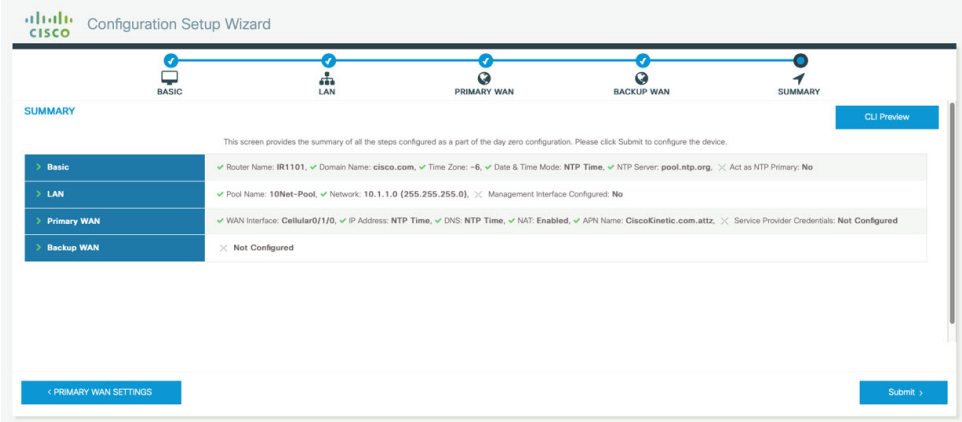
Step 6 Click **BACKUP WAN SETTINGS**. The BACKUP WAN Configuration screen appears. Select the button to Enable or Disable a backup WAN.

Figure 19: BACKUP WAN Configuration



Step 7 Click **Day 0 Config Summary**. The SUMMARY screen appears. Verify your entries before applying the configuration.

Figure 20: Summary Screen

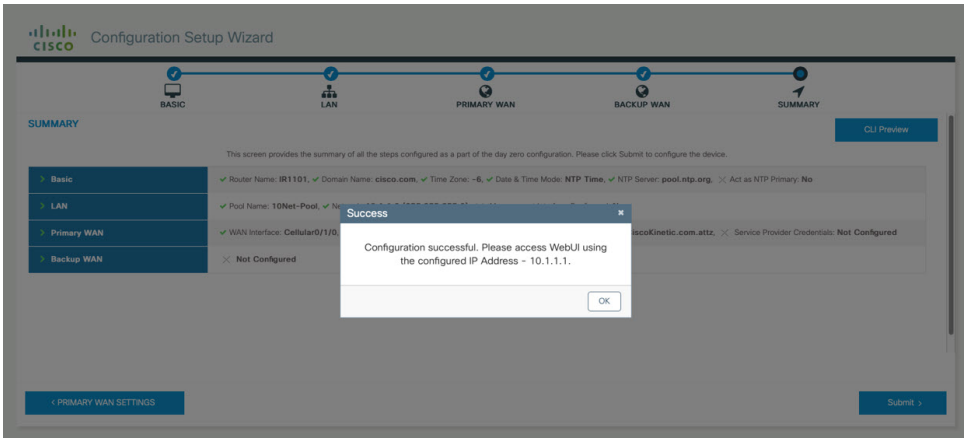


Step 8 (Optional) You can click on **CLI Preview** to see the Configuration that is being applied to the router. Close the CLI Preview, and if you are ready, click **Submit**.

Note
A CLI Preview example is found at the end of this section.

Step 9 After clicking on **Submit**, a dialog box will appear which informs you that the configuration has been applied successfully. The new WebUI ip address is also presented.

Figure 21: Submit Dialog Box



Example

The following is an example of a CLI Preview:

```
 ip domain name cisco.com
clock timezone GMT -6 00
ntp server pool.ntp.org

username admin privilege 15 secret 0 Mjc1N0dsb2NrIQ==

hostname "IR1101"
```

```
interface vlan 1
ip address 10.1.1.1 255.255.255.0
no shutdown
vlan 1
interface FastEthernet0/0/1
switchport access vlan 1
switchport trunk native vlan 1
switchport mode access
no shutdown
interface FastEthernet0/0/2
switchport access vlan 1
switchport trunk native vlan 1
switchport mode access
no shutdown
interface FastEthernet0/0/3
switchport access vlan 1
switchport trunk native vlan 1
switchport mode access
no shutdown
interface FastEthernet0/0/4
switchport access vlan 1
switchport trunk native vlan 1
switchport mode access
no shutdown
ip dhcp pool 10Net-Pool
dns-server 10.1.1.1
network 10.1.1.0 255.255.255.0
import all
default-router 10.1.1.1
lease 0 2

ip dhcp excluded-address 10.1.1.1

ip dns server
ip dns view default
default dns forwarder
default dns forwarding
default domain lookup
default domain name-server
interface Cellular0/1/0
description primary_wan
ip address negotiated
dialer in-band
dialer-group 1
pulse-time 1
shutdown
no shutdown
ip nat outside
exit
dialer-list 1 protocol ip permit

controller Cellular 0/1/0
lte sim data-profile 2 attach-profile 2 slot 0

ip route 0.0.0.0 0.0.0.0 Cellular0/1/0

ip nat inside source list 197 interface Cellular0/1/0 overload
access-list 197 permit ip any any
```

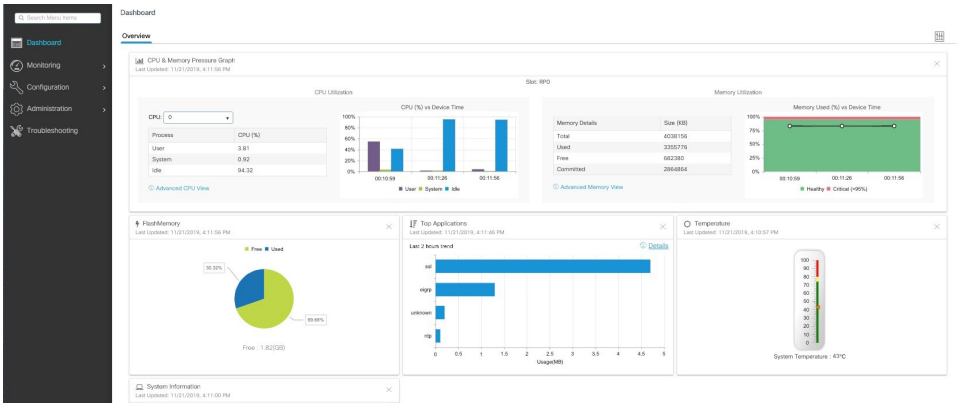
WebUI Dashboard

After completing the Day 0 setup, the WebUI can now be used for day to day administration. The WebUI opens up to an easy to use dashboard.

Note WebUI feature support may vary based on the license and platform type of your device.

The following figure shows the dashboard:

Figure 22: Dashboard



The following table provides an overview of the dashboard.

Dashboard	View dashlets that give you a snapshot of CPU and memory utilization and system information.
Monitoring	Monitor your network on a daily basis and perform other ad hoc operations related to network device inventory and configuration management.
Configuration	Configure your device.
Administration	Specify system configuration settings and user administration settings.
Troubleshooting	Troubleshoot connectivity problems and packet loss using Ping and Traceroute, and monitor device health and performance using web server logs and syslogs.

Configure access point name using Cisco WebUI

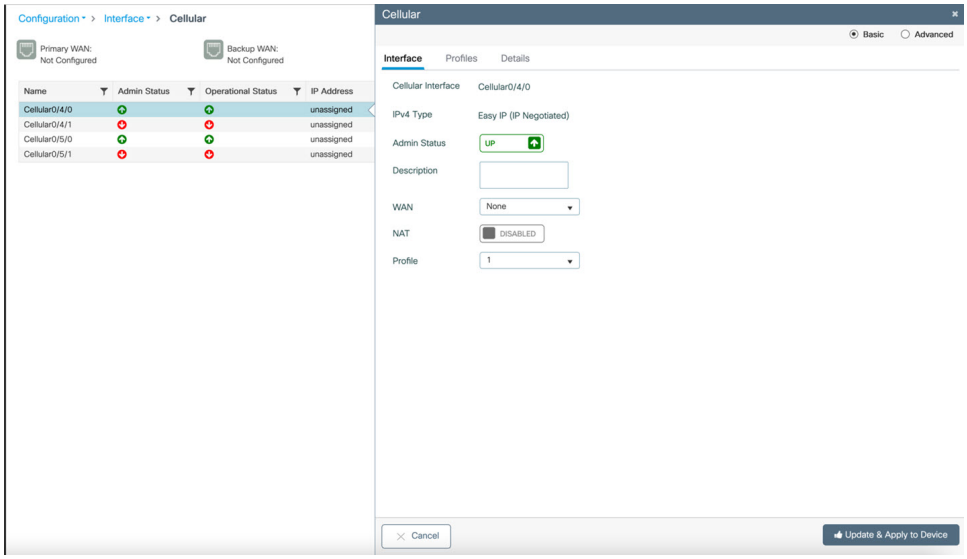
Starting with Cisco IOS XE Release 17.9.1, you can add, edit, or delete the APN from the Cisco WebUI interface. These steps provide an overview of how to perform this function.

Note This section describes only the new functionality and is not a complete overview of the WebUI.

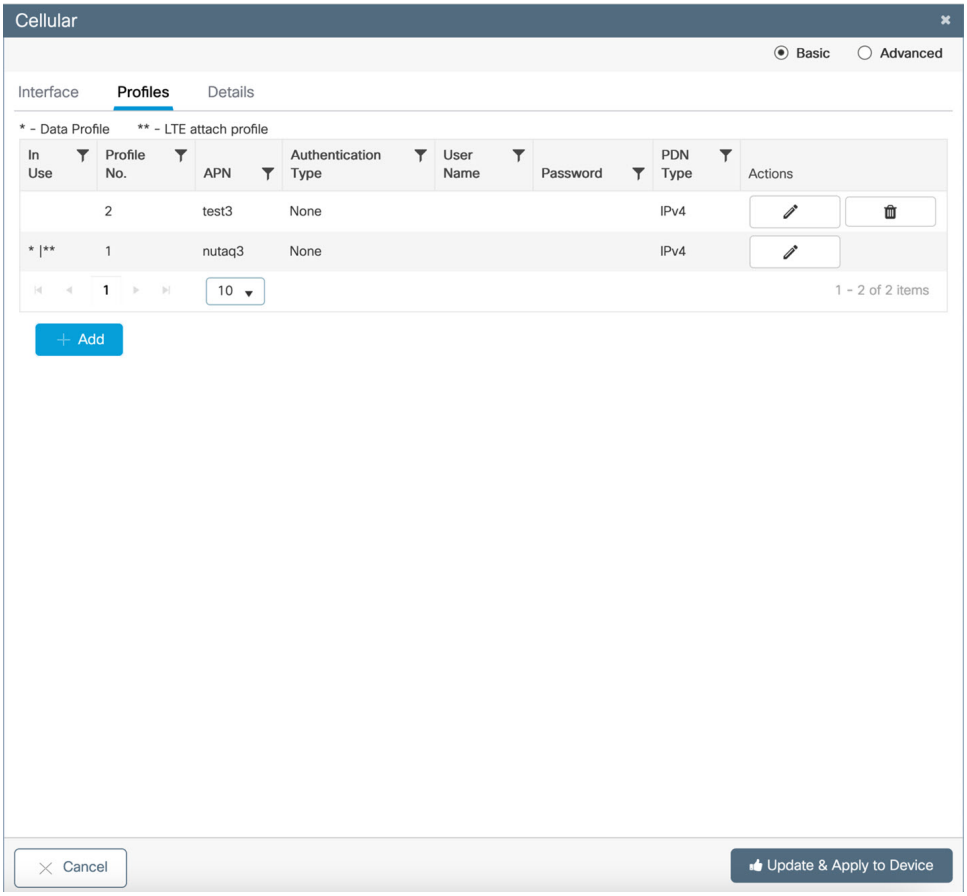
Procedure

Step 1 Navigate to **Configuration > Interface > Cellular**.

Step 2 Double-click the cellular interface based on your platform.



Step 3 On the Cellular window, click the **Profiles** tab.

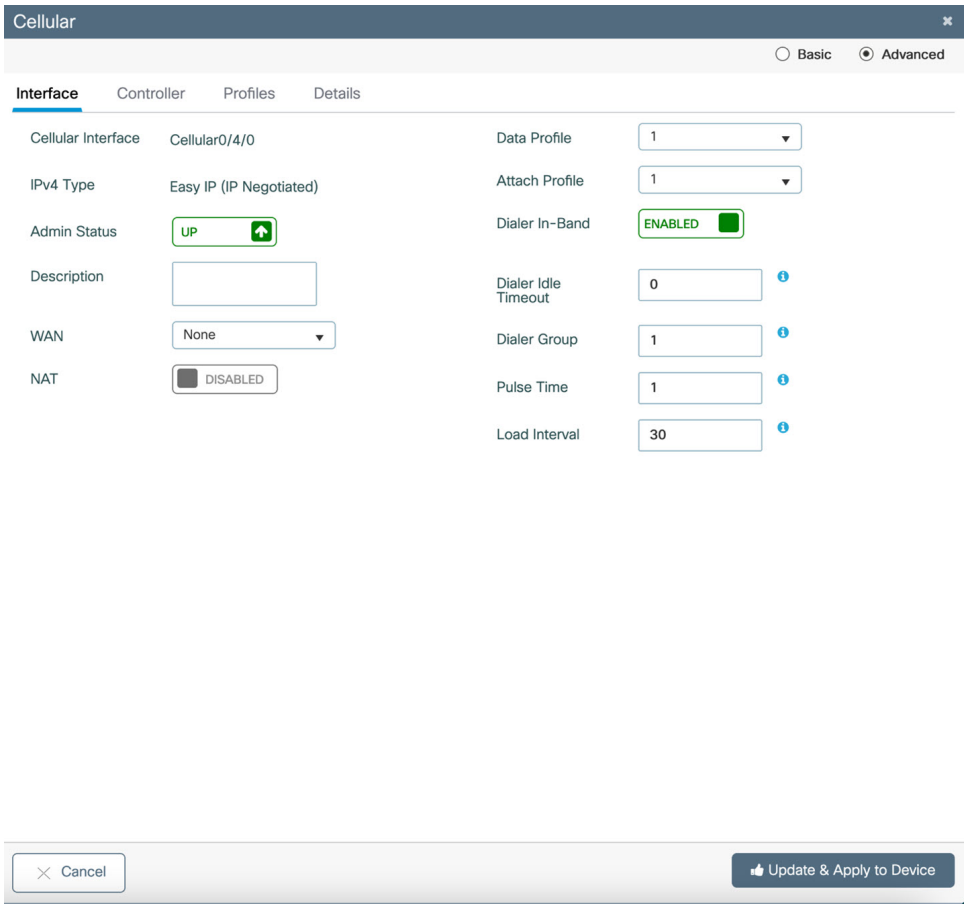


Step 4 From the **Profiles** tab, add, delete, or edit the APN as needed.

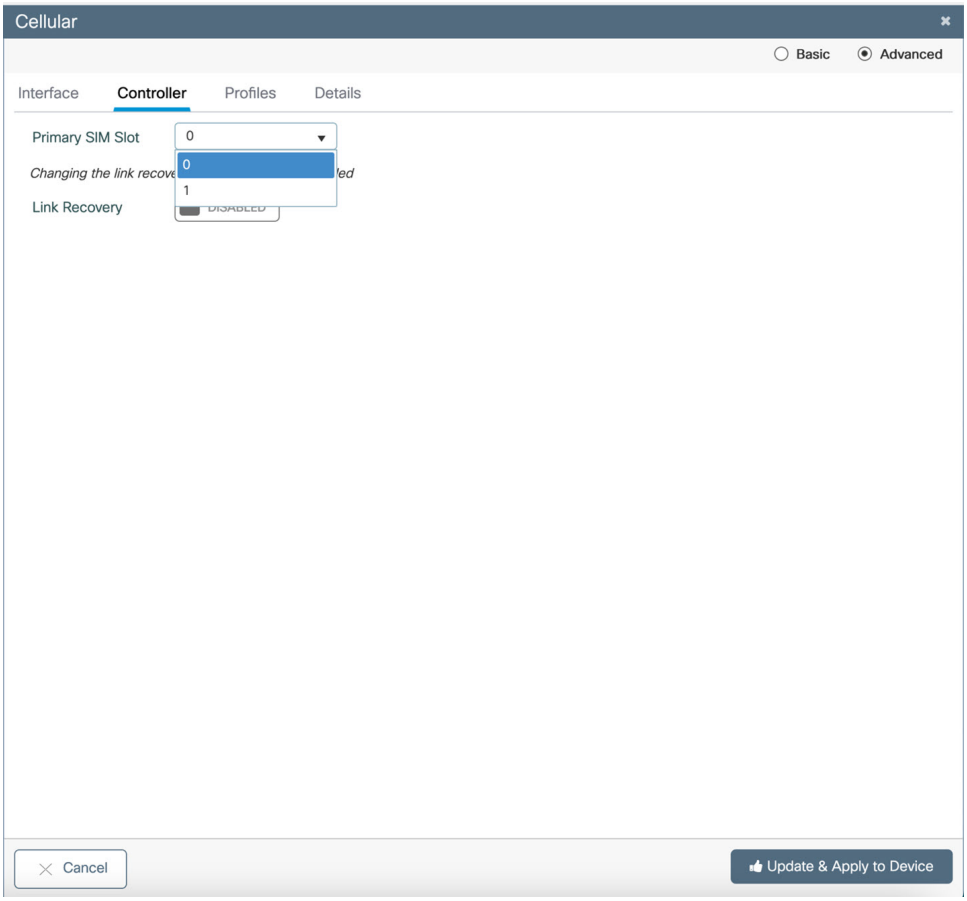
Step 5 Click **Update & Apply to Device** at the bottom of the window.

Step 6 To change the SIM slot, click the **Advanced** radio button at the top of the window.

By default, the APN is attached to SIM slot 0. You can change the APN to SIM slot 1 by using the WebUI.

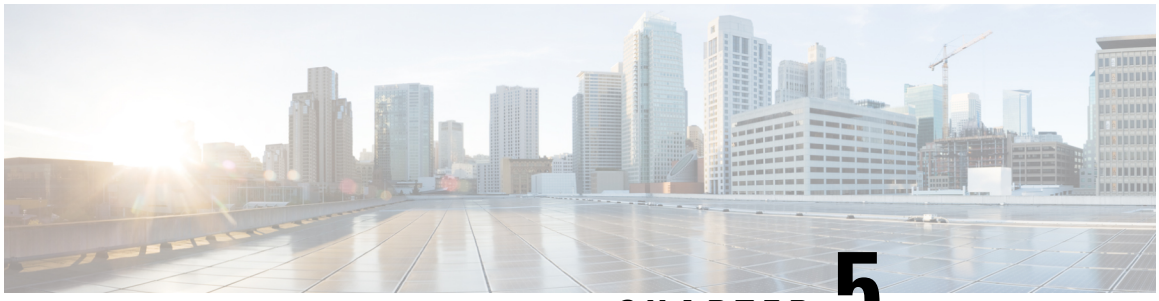


Step 7 Click the **Controller** tab at the top of the window.



Step 8 From the **Primary SIM Slot** drop-down list, select slot 1.

Step 9 Click **Update & Apply to Device** at the bottom of the window.



CHAPTER 5

Secure Shell

This section contains the following topics:

- Secure shell, on page 61
- How to configure secure shell, on page 63
- Secure copy protocol, on page 68
- SSH rekeying, on page 70
- Additional references, on page 75

Secure shell

Secure Shell (SSH) is a protocol that

- provides a secure, remote connection to a device
- provides more security for remote connections than Telnet by providing strong encryption when a device is authenticated, and
- supports SSH Version 1 (SSHv1) and SSH Version 2 (SSHv2) in this software release.

Prerequisites for configure secure shell

The following are the prerequisites for configuring the device for secure shell (SSH):

- For SSH to work, the switch needs an RSA public/private key pair.
- The Secure Shell (SSH) server requires an IPsec (Data Encryption Standard [DES] or 3DES) encryption software image; the SSH client requires an IPsec (DES or 3DES) encryption software image.)
- Configure a hostname and host domain for your device by using the hostname and ip domain-name commands in global configuration mode. Use the **hostname** and **ip domain-name** commands in global configuration mode.

Restrictions for configuring secure shell

Consider these restrictions when configuring the router for secure shell:

- The router supports RSA authentication.

- SSH supports only the execution-shell application.
- The SSH server and the SSH client are supported only on Data Encryption Standard (DES) (56-bit) and 3DES (168-bit) data encryption software. In DES software images, DES is the only encryption algorithm available. In 3DES software images, both DES and 3DES encryption algorithms are available.

Note Cisco highly recommends the 3DES encryption as it is stronger. See the Cisco IOS-XE Device hardening guide at https://www.cisco.com/c/en/us/support/docs/ip/access-lists/13608-21.html for details.

- This software release supports IP Security (IPSec).
- The router supports the Advanced Encryption Standard (AES) encryption algorithm with a 128-bit key, 192-bit key, or 256-bit key. However, symmetric cipher AES to encrypt the keys is not supported.
- The login banner is not supported in Secure Shell Version 1. It is supported in Secure Shell Version 2, which Cisco recommends due to its better security.
- The -l keyword and userid :{number} {IP-address} delimiter and arguments are mandatory when configuring the alternative method of Reverse SSH for console access.

SSH and router access

Secure Shell (SSH) is a protocol that

- provides a secure, remote connection to a device
- provides more security for remote connections than Telnet does by providing strong encryption when a device is authenticated, and
- supports SSH Version 1 (SSHv1) and SSH Version 2 (SSHv2).

SSH protocol characteristics

SSH functions the same in IPv6 as in IPv4. For IPv6, SSH supports IPv6 addresses and enables secure, encrypted connections with remote IPv6 nodes over an IPv6 transport.

SSH servers, integrated clients, and supported versions

The Secure Shell (SSH) Integrated Client feature is an application that

- runs over the SSH protocol to provide device authentication and encryption
- enables a Cisco device to make a secure, encrypted connection to another Cisco device or to any other device running the SSH server, and
- allows for secure communication over an unsecured network with authentication and encryption.

SSH server and client compatibility

The SSH server and SSH integrated client are applications that run on the switch. This connection provides functionality similar to that of an outbound Telnet connection except that the connection is encrypted.

SSH compatibility includes these components:

- SSH server works with the SSH client supported in this release and with non-Cisco SSH clients
- SSH client works with publicly and commercially available SSH servers
- SSH client supports the ciphers of Data Encryption Standard (DES), 3DES, and password authentication



Note The SSH client functionality is available only when the SSH server is enabled.

User authentication is performed like that in the Telnet session to the device. SSH also supports these user authentication methods:

- TACACS+
- RADIUS
- Local authentication and authorization

Recommendation: SSH configuration guidelines

Follow these guidelines when configuring the device as an SSH server or SSH client:

- An RSA key pair generated by a SSHv1 server can be used by an SSHv2 server, and the reverse.
- If you get CLI error messages after entering the **crypto key generate RSA global** configuration command, an RSA key pair has not been generated. Reconfigure the hostname and domain, and then enter the **crypto key generate RSA** command.
- When generating the RSA key pair, the message *No hostname specified* might appear. If it does, you must configure an IP hostname by using the **hostname** global configuration command.
- When generating the RSA key pair, the message No domain specified might appear. If it does, you must configure an IP domain name by using the **IP domain-name** global configuration command.
- When configuring the local authentication and authorization authentication method, make sure that AAA is disabled on the console.

How to configure secure shell

This reference provides the configuration parameters and settings required to establish secure remote connections to network devices using Secure Shell (SSH).

Set up the router to run SSH

SSH provides secure remote access to network devices. This configuration enables encrypted communication and helps protect against unauthorized access attempts.



Note Ensure that you configure user authentication for local or remote access.

Follow these steps to set up the router to run SSH.

Procedure

Step 1 Use the **enable** command to enter privileged EXEC mode.

Example:

```
Router> enable
```

Step 2 Use the **configure terminal** command to enter global configuration mode.

Example:

```
Router# configure terminal
```

Step 3 Use the **hostname** *your_hostname* and **ip domain-name** *your_domain_name* commands to configure a hostname and domain name for the device.

Example:

```
Router(config)# hostname <your_hostname>
Router(config)# ip domain-name <your_domain_name>
```

Step 4 Use the **crypto key generate rsa** command to generate an RSA key pair to enable SSH.

Example:

```
Router(config)# crypto key generate rsa
```

Note
Use a modulus size of at least 1024 bits for security.

Step 5 Use the **end** command to exit global configuration mode.

Example:

```
Router(config)# end
```

Step 6 Use the **show running-config** command to verify the configuration.

Example:

```
Router# show running-config
```

Step 7 (Optional) Use the **copy running-config startup-config** command to save the configuration.

Example:

```
Router# copy running-config startup-config
```

The router is now configured to accept SSH connections. You can connect remotely using an SSH client with the configured hostname and the appropriate user credentials.

Configure the SSH server

Configure the SSH server on a device to enable secure remote CLI access and to control SSH protocol parameters. Use this procedure when you need to set up the device as an SSH server to provide secure command-line interface access for remote management.

Note This procedure is required only if you are configuring the device as an SSH server.

Follow these steps to configure the SSH server.

Procedure

Step 1 Use the **enable** command to enable privileged EXEC mode. Enter your password if prompted.

Example:

```
Router> enable
```

Step 2 Use the **configure terminal** command to enter global configuration mode.

Example:

```
Router# configure terminal
```

Step 3 (Optional) Use the **ip ssh version** [**2**] command to configure the device to run SSH Version 2.

Example:

```
Router(config)# ip ssh version 2
```

Note
If you do not enter this command or do not specify a keyword, the SSH server selects the latest SSH version supported by the SSH client. For example, if the SSH client supports SSHv1 and SSHv2, the SSH server selects SSHv2.

Step 4 Use the **ip ssh** {**timeout** *seconds* | **authentication-retries** *number*} command to configure the SSH control parameters.

Example:

```
Router(config)# ip ssh timeout 90
Router(config)# ip ssh authentication-retries 2
```

Note

- Specify the timeout value in seconds; the default is 120 seconds, and the range is 0 to 120 seconds. This parameter applies to the SSH negotiation phase. After the connection is established, the device uses the default timeout values of the CLI-based sessions.
- By default, up to five simultaneous, encrypted SSH connections for multiple CLI-based sessions over the network are available (session 0 to session 4). After the execution shell starts, the CLI-based session timeout value returns to the default of 10 minutes.

- Specify the number of times that a client can re-authenticate to the server. The default is 3, and the range is 0 to 5.
- Repeat this step when configuring both parameters.

Step 5 (Optional) Use the **line vty** *line-number* [*ending-line-number*] command to enter line configuration mode and configure the virtual terminal line settings. Then, use the **transport input ssh** command to limit the device to SSH connections only.

Example:

```
Router(config)# line vty 1 10
Router(config-line)# transport input ssh
```

Note

- For the *line-number* and *ending-line-number* arguments, the range is from 0 to 15.
- The `transport input ssh` command specifies that the device prevents non-SSH Telnet connections, limiting the device to only SSH connections.

Step 6 Use the **end** command to exit line configuration mode and return to privileged EXEC mode.

Example:

```
Router(config-line)# end
```

Step 7 Use the **show running-config** command to verify your entries.

Example:

```
Router# show running-config
```

Step 8 (Optional) Use the **copy running-config startup-config** command to save your entries in the configuration file.

Example:

```
Router# copy running-config startup-config
```

The SSH server is configured and ready to accept secure connections with the specified parameters for version, timeout, and authentication settings.

Monitor the SSH configuration and status

This reference provides commands to monitor and display SSH server configuration and status information.

Table 4: Commands for displaying the SSH server configuration and status

Command	Purpose
show ip SSH	Shows the version and configuration information for the SSH server.
show SSH	Shows the status of the SSH server.

Configure the router for local authentication and authorization

You can configure AAA to operate without a server by setting the router to implement AAA in local mode. The router then handles authentication and authorization. No accounting is available in this configuration.

Note To secure the router for HTTP access by using AAA methods, you must configure the router with the `ip http authentication aaa` global configuration command. Configuring AAA authentication alone does not secure the router for HTTP access.

Follow these steps to configure AAA to operate without a server by setting the router to implement AAA in local mode.

Procedure

Step 1 Use the **configure terminal** command to enter global configuration mode.

Example:

```
Router# configure terminal
```

Step 2 Use the **aaa new-model** command to enable AAA.

Example:

```
Router(config)# aaa new-model
```

Step 3 Use the **aaa authentication login default local** command to set the login authentication to use the local username database.

Example:

```
Router(config)# aaa authentication login default local
```

Note
The `default` keyword applies the local user database authentication to all ports.

Step 4 Use the **aaa authorization exec local** command to configure user AAA authorization, check the local database, and allow the user to run an EXEC shell.

Example:

```
Router(config)# aaa authorization exec local
```

Step 5 Use the **aaa authorization network local** command to configure user AAA authorization for all network-related service requests.

Example:

```
Router(config)# aaa authorization network local
```

Step 6 Use the **username** *name* [**privilege** *level*] **password** *encryption-type password* command to enter the local database and establish a username-based authentication system.

Example:

```
Router(config)# username your_user_name privilege 1 password 7 secret567
```

Note

- Repeat this command for each user.
- For *name*, specify the user ID as one word. Spaces and quotation marks are not allowed.
- (Optional) For *level*, specify the privilege level the user has after gaining access. The range is 0 to 15. Level 15 gives privileged EXEC mode access; level 0 gives user EXEC mode access.
- For *encryption-type*, enter 0 to specify that an unencrypted password follows, or enter 7 to specify that a hidden password follows.
- For *password*, specify the password the user must enter to gain access to the router. The password must be from 1 to 25 characters, can contain embedded spaces, and must be the last option specified in the `username` command.

Step 7 Use the **end** command to exit configuration mode and return to privileged EXEC mode.

Example:

```
Router(config)# end
```

Step 8 Use the **show running-config** command to verify your entries.

Example:

```
Router# show running-config
```

Step 9 (Optional) Use the **copy running-config startup-config** command to save your entries in the configuration file.

Example:

```
Router# copy running-config startup-config
```

The router is configured for local authentication and authorization using AAA methods. Users can authenticate using the local username database, and authorization is handled locally by the router.

Secure copy protocol

The Secure Copy Protocol (SCP) is a network protocol that

- provides a secure and authenticated method for copying router configuration or router image files
- relies on Secure Shell (SSH), an application and a protocol that provide a secure replacement for the Berkeley r-tools, and
- enables secure file transfer operations between network devices.

Requirements for secure copy configuration

Before enabling SCP, you must correctly configure SSH, authentication, and authorization on the switch.

- Because SCP relies on SSH for its secure transport, the router must have an RSA key pair.
- SCP relies on SSH for security.

- SCP requires that authentication, authorization, and accounting (AAA) authorization be configured so the router can determine whether the user has the correct privilege level.
- A user must have appropriate authorization to use SCP.
- A user who has appropriate authorization can use SCP to copy any file in the Cisco IOS File System (IFS) to and from a switch by using the **copy** command. An authorized administrator can also do this from a workstation.

Restrictions for configuring secure copy

Before enabling SCP, you must correctly configure SSH, authentication, and authorization on the router.

When using SCP, you cannot enter the password into the **copy** command. You must enter the password when prompted.

Configure secure copy

This task enables SCP server functionality on a Cisco router to allow secure file transfers between the router and remote hosts. SCP provides a secure method for copying files to and from network devices.



Note Before you configure SCP, ensure that proper authentication mechanisms are in place.

Follow these steps to configure the Cisco router for Secure Copy (SCP) server-side functionality.

Procedure

Step 1 Use the **enable** command to enable privileged EXEC mode. Enter your password if prompted.

Example:

```
Router> enable
```

Step 2 Use the **configure terminal** command to enter global configuration mode.

Example:

```
Router# configure terminal
```

Step 3 Use the **aaa new-model** command to enable the AAA access control model.

Example:

```
Router(config)# aaa new-model
```

Step 4 Use the **aaa authentication login** {**default** | *list-name*} *method1* [*method2...*] command to set AAA authentication at login.

Example:

```
Router(config)# aaa authentication login default group tacacs+
```

Step 5 Use the **username** *name* [**privilege** *level*] **password** *encryption-type encrypted-password* command to establish a username-based authentication system.

Example:

```
Router(config)# username superuser privilege 2 password 0 superpassword
```

Note
You can omit this step if a network-based authentication mechanism, such as TACACS+ or RADIUS, is configured.

Step 6 Use the **ip scp server enable** command to enable SCP server-side functionality.

Example:

```
Router(config)# ip scp server enable
```

Step 7 Use the **exit** command to exit global configuration mode and return to privileged EXEC mode.

Example:

```
Router(config)# exit
```

Step 8 (Optional) Use the **show running-config** command to display the SCP server-side functionality.

Example:

```
Router# show running-config
```

Step 9 (Optional) Use the **debug ip scp** command to troubleshoot SCP authentication problems.

Example:

```
Router# debug ip scp
```

SCP server functionality is now enabled on the router, allowing secure file transfers. You can now use SCP commands to copy files to and from the router.

```
Router# copy scp <somefile> your_username@remotehost:/<some/remote/directory>
```

SSH rekeying

SSH rekeying is a security feature that:

- replaces the encryption and integrity keys during an active SSH session
- initiates a new key exchange based on a configured time or data threshold, and
- helps reduce the risk of long-term key exposure.

SSH rekeying characteristics

SSH rekeying can be initiated by either the SSH client or server. During rekeying, data transfer halts until the key exchange completes. The encryption algorithm negotiated at session startup remains in use during rekeying. OpenSSH-based clients version 7.5 or later support SSH rekey exchange.

SSH rekeying scenarios

- Rekeying every hour to ensure session security over long connections.
- Rekeying after 1 GB of data transfer to limit the data encrypted with a single key.

SSH rekeying types

SSH rekeying types are configuration methods that:

- enable periodic renewal of encryption keys during SSH sessions
- provide time-based and volume-based triggering options using the **ip SSH rekey** command, and
- enhance security by preventing prolonged use of the same encryption keys.

Configuration methods

You can configure SSH rekeying on Cisco routers in two ways by using the **ip SSH rekey** command:

- Time-based rekeying: Specify the interval (in minutes or seconds) after which rekeying occurs.
- Volume-based rekeying: Specify the data volume (in KB or MB) that triggers rekeying.

By default the **ip SSH rekey** command is disabled. If you configure these settings using the **ip SSH rekey** command without specifying time or volume, then it will apply both time and volume rekey with default values. The default value for time is 60 minutes and volume is 1 GB.

Rekeying by time initiates a key exchange after a specific time interval:

- In Common Criteria mode, the maximum allowed time is less than 60 minutes or 3600 seconds.
- The configurable range is 10 to 1440 minutes.
- For some devices, the default is 60 minutes, and the command accepts values in minutes.

Rekeying by volume enforces a new key exchange when the transmitted or received data reaches a configured threshold:

- In Common Criteria mode, the maximum rekey volume is 1024 MB (1 GB).
- The configurable range for the rekey volume is 100 KB to 4 GB.
- For some devices, the default is 1 GB, and the command accepts values in kilobytes.

Prerequisites for configuring SSH rekeying

To configure SSH rekeying on Cisco routers, ensure these prerequisites are met.

- The device runs Cisco IOS with SSH enabled.
- SSH version 2 is recommended.
- Administrative privilege access is required.

Best practices for SSH rekeying configuration

Best practices for SSH rekeying configuration are:

- Enable SSH rekeying to reduce the risk from potential key compromise, especially in high-security environments.
- Ensure rekey intervals comply with security policies, such as Common Criteria, by setting appropriate time or volume thresholds.

SSH session idle timeout settings

SSH sessions can be configured to disconnect after a period of inactivity.

- SSH idle timeout: **line vty <0-240>** (in minutes; 0 means never timeout)
- **exec-timeout <0-240>** (in minutes; 0 means never timeout)

Restrictions for SSH rekeying on Cisco routers

The following limitations apply to SSH rekeying on Cisco routers:

- During SCP downloads, the device acting as receiver cannot control the sender's data volume, so rekey is triggered only after the configured volume is received.
- Large SSH bulk mode settings (for example, 1 GB) can lead to fewer rekeys than expected due to larger data windows.
- SSH sessions established without rekey configuration do not support rekeying
- When rekey is enabled, existing sessions will only adopt rekeying after the next exchange initiated from the other side.
- Removing the rekey configuration disables SSH rekey for existing sessions.
- Changing the rekey configuration does not affect active sessions until the subsequent rekey exchange.
- When a rekey exchange occurs, the related data and time parameters for the session reset to the new configured value.
- Do not modify the encryption method during rekeying.

The actual number of rekeys during SCP downloads may not match calculated expectations.

This behavior is by design and aligns with current SSH implementation.

Example observation:

Transferring a 631.3 MB file with a 100 KB rekey threshold could yield an expected 6,463 rekeys, but only 689 may occur.

This discrepancy is documented and not considered a defect.

Configure SSH rekeying

SSH rekeying enhances session security by periodically renewing encryption keys. Rekeying can be based on time intervals or data volumes. The configuration involves enabling SSH, setting the SSH version, and defining the rekey parameters.

Follow these steps to configure SSH rekeying.

Procedure

Step 1 Use the **configure terminal** command to enter global configuration mode.

Example:

```
Router# configure terminal
```

Step 2 Use the **ip domain-name** *domain-name* command to set the domain name.

Example:

```
Router(config)# ip domain-name example.com
```

Step 3 Use the **crypto key generate rsa modulus** *modulus-size* command to generate the RSA keys.

Example:

```
Router(config)# crypto key generate rsa modulus 2048
```

Step 4 Use the **ip ssh version 2** command to set the SSH version to 2.

Example:

```
Router(config)# ip ssh version 2
```

Step 5 Use the **ip ssh rekey time** *minutes* command to configure time-based rekey.

Example:

```
Router(config)# ip ssh rekey time 120
```

Note
The valid range is 10 to 1440 minutes (the default is 60 minutes).

Step 6 Use the **ip ssh rekey volume** *size-in-kb* command to configure volume-based rekey.

Example:

```
Router(config)# ip ssh rekey volume 10240
```

Note

- This example triggers a rekey after 10,240 KB.
- The valid range is 100 to 4194303 KB (the default is 1 GB).

Step 7 (Optional) Use the **no ip ssh rekey time** or **no ip ssh rekey volume** command to disable time-based or volume-based rekeying.

Example:

```
Router(config)# no ip ssh rekey time
```

Example:

```
Router(config)# no ip ssh rekey volume
```

SSH rekeying is enabled on the device, improving session security through periodic key renewal.

Verify SSH rekey configuration

Verification ensures that SSH rekeying is correctly configured and operational on the device.

Follow these steps to verify the SSH rekey configuration.

Procedure

Step 1 Use the **show ip ssh** command to display the SSH configuration details.

Example:

```
Router# show ip ssh
            SSH Enabled - version 2.0
            Authentication methods:publickey,keyboard-interactive,password
            Authentication Publickey
Algorithms:ecdsa-sha2-nistp256,ecdsa-sha2-nistp384,ecdsa-sha2-nistp521,ssh-ed25519,x509v3-ecdsa-sha2-nistp256,x509v3-ecdsa-sha2-nistp384,x509v3-ecdsa-sha2-nistp521,rsa-sha2-256,rsa-sha2-512,x509v3-rsa2048-sha256

            Hostkey
Algorithms:ecdsa-sha2-nistp256,ecdsa-sha2-nistp384,ecdsa-sha2-nistp521,rsa-sha2-512,rsa-sha2-256

            Encryption
Algorithms:chacha20-poly1305@openssh.com,aes128-gcm@openssh.com,aes256-gcm@openssh.com,aes128-gcm,aes256-gcm,aes128-ctr,aes192-ctr,aes256-ctr

            MAC
Algorithms:hmac-sha2-256-etm@openssh.com,hmac-sha2-512-etm@openssh.com,hmac-sha2-256,hmac-sha2-512

            KEX
Algorithms:curve25519-sha256,curve25519-sha256@libssh.org,ecdh-sha2-nistp256,ecdh-sha2-nistp384,ecdh-sha2-nistp521,diffie-hellman-group14-sha256,diffie-hellman-group16-sha512

            Authentication timeout: 120 secs; Authentication retries: 3
            Minimum expected Diffie Hellman key size : 2048 bits
            IOS Keys in SECSH format(ssh-rsa, base64 encoded): TP-self-signed-1538213291
            Modulus Size : 2048 bits
           ssh-rsa AAAAB3NzaC1yc2EAAAADAQABAAABAQCw/l05OHVk/3MAC30fIpLf6rphTETPZtnIdlYbZPCc

           L+gLy+dxsxq8kYXZ5lTQq+lstB+PB868xDz/mkk9OfNFcEwa2SavoyHuDeqN58FMFmU4z9guwfe66dIk

           sdt1qA086Wc45BUBYrE3R4uVC1gQdkuHyIZmN54c3lvSRuToaGNo3M10rRjIELOWOV1Qbcr8e0nrAAmf

           Xznx2lDqO2QPdynLEpLWinLpnwEFEJtQ2DXRAZO+a1SqGvfTbLZ+EQ7SCI39X0xC5YVCFRjD3NF+6NpO

            Q6wt0my/5xNXr1m5uabfOco96lAXbCD4w55JDuv0Eq3siefVAuXXtgGpDCVB
            IOS Keys in SECSH format(ssh-ec, base64 encoded): NONE
```

Note

The SSH session attributes and algorithms include:

- SSH version and enablement status
- Supported authentication methods and algorithms (public key, keyboard-interactive, password)
- Hostkey and encryption algorithms

- MAC and key exchange algorithms
- Authentication timeout and retry settings
- Minimum expected Diffie-Hellman key size
- Current session keys (in SECSH format)

Step 2 Use the **show running-config | include ssh** command to display the SSH-related running configuration details.

Example:

```
Router# show running-config | include ssh
            ip ssh bulk-mode 13107245
            ip ssh rekey volume 100
            transport input ssh
            transport input ssh
```

Example:

```
Router# show running-config | include ssh
            ip ssh bulk-mode 13107245
            ip ssh rekey time 120
            transport input ssh
            transport input ssh
```

Step 3 Use the **show ssh rekey** command to monitor SSH rekey activity and current session information.

Example:

```
Router# show ssh rekey
            SSHv2 server sessions
            Connection  TimeTillRekey  DataLeftTillRekey TimeRekeys  VolumeRekeys
            0           0:00           98                0           0

            %No SSHv2 client connections running.
```

Example:

```
Router# show ssh rekey
            SSHv2 server sessions
            Connection  TimeTillRekey  DataLeftTillRekey TimeRekeys  VolumeRekeys
            0           0:00           99997             0           0

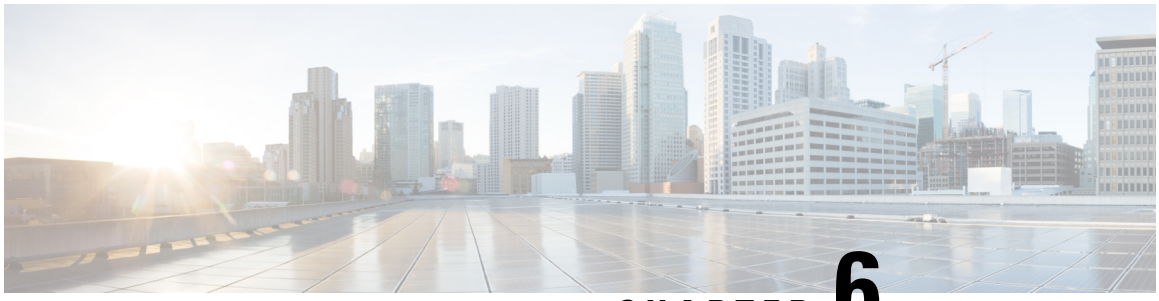
            SSHv2 client sessions
            Connection  TimeTillRekey  DataLeftTillRekey TimeRekeys  VolumeRekeys
            1           0:00           0                 0           0
```

Additional references

This sections provide references related to the SSH feature.

Related Topic	Document Title
Configuring Identity Control policies and Identity Service templates for Session Aware networking.	Session Aware Networking Configuration Guide, Cisco IOS XE Release 3SE: https://www.cisco.com/en/US/docs/ios-xml/ios/san/configuration/xe-3se/3850/san-xe-3se-3850-book.pdf

Related Topic	Document Title
Configuring RADIUS, TACACS+, Secure Shell, 802.1X and AAA.	Secure Shell Configuration Guide, Cisco IOS XE Gibraltar 16.11.x: https://www.cisco.com/c/en/us/td/docs/switches/lan/catalyst9500/software/release/16-11/configuration_guide/sec/b_1611_sec_9500_cg/configuring_secure_shell__ssh_.html



CHAPTER 6

Inline Tagging

- SGT Inline Tagging, on page 77
- Guidelines for Inline Tagging, on page 78
- Configuring SGT Static Inline Tagging, on page 79
- Example: Configuring SGT Static Inline Tagging, on page 80
- Troubleshoot the Security Group Tagging Configuration, on page 80

SGT Inline Tagging

Each security group in a Cisco TrustSec domain is assigned a unique 16 bit tag called the Security Group Tag (SGT). The SGT is a single label indicating the privileges of the source within the entire network. It is in turn propagated between network hops allowing any intermediary devices (switches, routers) to enforce polices based on the identity tag.

Cisco TrustSec-capable devices have built-in hardware capabilities that can send and receive packets with SGT embedded in the MAC (L3) layer. This feature is called Layer 3 (L3)-SGT Imposition. It allows ethernet interfaces on the device to be enabled for L3-SGT imposition so that the device can insert an SGT in the packet to be carried to its next hop ethernet neighbor. SGT-over-Ethernet is a method of hop-by-hop propagation of SGT embedded in clear-text (unencrypted) ethernet packets. The inline identity propagation is scalable, provides near line-rate performance, and avoids control plane overhead.

The Cisco TrustSec with SGT Exchange Protocol V4 (SXPv4) feature supports Cisco TrustSec metadata-based L3-SGT. When a packet enters a Cisco TrustSec-enabled interface, the IP-SGT mapping database (with dynamic entries built by SXP and/or static entries built by configuration commands) is analyzed to learn the SGT corresponding to the source IP address of the packet, which is then inserted into the packet and carried throughout the network within the Cisco TrustSec header.

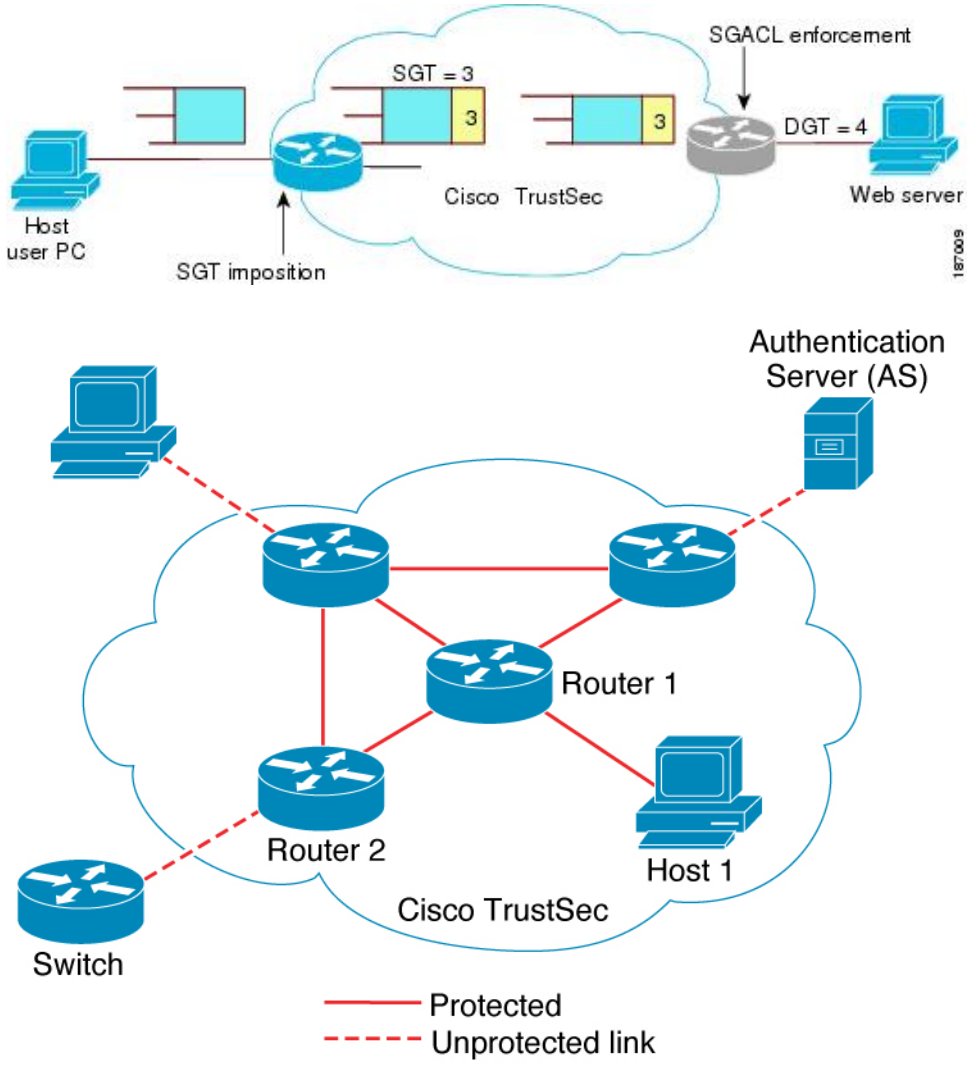
Cisco TrustSec enforces role-based access control using Security Group Tags (SGTs) and Destination Group Tags (DGTs). An SGT is assigned to traffic at the ingress point or can be propagated from a trusted peer device, identifying the security group of the source. At the egress point, the Destination Group Tag (DGT) is determined. Access decisions are made using Security Group Access Control Lists (SGACLs), which define permitted interactions between source and destination groups.

A network device at the ingress of Cisco TrustSec cloud needs to determine the SGT of the packet entering the Cisco TrustSec cloud so that it can tag the packet with that SGT when it forwards it into the Cisco TrustSec cloud. The SGT of a packet can be determined with these methods:

- SGT field on Cisco TrustSec header: If a packet is coming from a trusted peer device, it is assumed that the Cisco TrustSec header carries the correct SGT field. This situation applies to a network that is not the first network device in the Cisco TrustSec cloud for the packet.
- SGT lookup based on source IP address: In some cases, the administrator may manually configure a policy to decide the SGT of a packet based upon the source IP address. An IP address to SGT table can also be populated by the SXP protocol.

The following figures explains the topologies:

Figure 23: Cisco TrustSec Network



Guidelines for Inline Tagging

This principle defines the restrictions and supported features for CTS (Cisco TrustSec) functionality on Pure L3 interfaces and convertible switch ports operating in L3 mode.

Restrictions and Supported Features for L2 and L3 Interfaces

- Traffic Switching: Inter-VLAN communication is not supported on Layer 2 ports.
- CTS Trust and Propagation: Convertible switch ports functioning as L3 interfaces are enabled for CTS trust and propagation. Conversely, switch ports acting as L2 interfaces do not support CTS functionality.
- Supported only on IPv4 hosts.
- Static classification (IP-SGT and SVI-SGT) is supported on L2/L3 interfaces.
- Enforcement: CTS enforcement is not supported.

Configuring SGT Static Inline Tagging

Procedure

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**interface gigabitethernet** *port* **Example:** `Device(config)# interface gigabitethernet 0/0/0`	Configures the interface on which Cisco TrustSec SGT authorization and forwarding is enabled, and enters interface configuration mode.
Step 4	**cts manual** **Example:** `Device(config-if)# cts manual`	Enables Cisco TrustSec SGT authorization and forwarding on the interface, and enters Cisco TrustSec manual interface configuration mode.
Step 5	**propagate sgt** **Example:** `Device(config-if-cts-manual)# propagate sgt`	Enables Cisco TrustSec SGT propagation on an interface. **Note** Use this command in situations where the peer device is not capable of receiving SGT over Ethernet packets (that is, when a peer device

	Command or Action	Purpose
		does not support Cisco Ethertype CMD 0x8909 frame format).
Step 6	**policy static sgt** *tag* [**trusted**] **Example:** `Device(config-if-cts-manual)# policy static sgt 77 trusted`	Configures a static SGT ingress policy on the interface and defines the trustworthiness of an SGT received on the interface. **Note** The **trusted** keyword indicates that the interface is trustworthy for Cisco TrustSec. The SGT value received in the Ethernet packet on this interface is trusted and will be used by the device for any SG-aware policy enforcement or for the purpose of egress-tagging.
Step 7	**exit** **Example:** `Device(config-if-cts-manual)# exit`	Exits Cisco TrustSec manual interface configuration mode and enters interface configuration mode.
Step 8	**end** **Example:** `Device(config-if)# end`	Exits interface configuration mode and enters privileged EXEC mode.

Example: Configuring SGT Static Inline Tagging

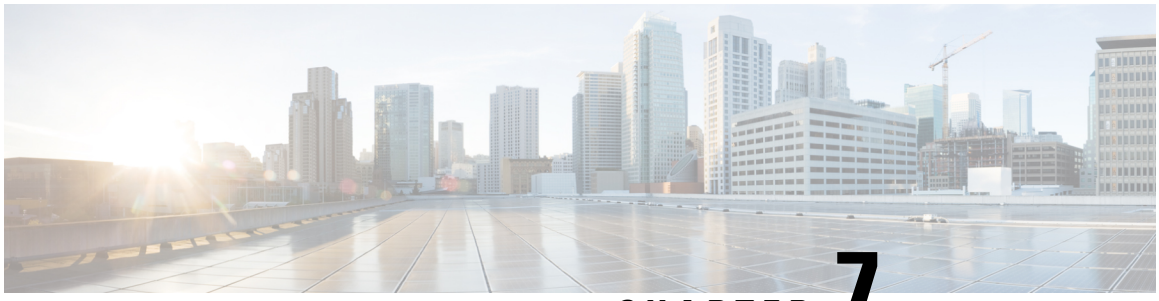
This is an example showing how to enable an interface connecting to a non-trustsec capable endpoint.

```
Device# configure terminal
Device(config)# interface gigabitethernet 0/0/0
Device(config-if)# cts manual
Device(config-if-cts-manual)# propagate sgt
Device(config-if-cts-manual)# policy static sgt 77 trusted
```

Troubleshoot the Security Group Tagging Configuration

You can use the following commands to troubleshoot the Cisco TrustSec configuration:

- **debug cts all**



CHAPTER 7

NTP Timing Based on GPS Clock

This chapter contains the following sections:

- Configure NTP using GPS Time, on page 81

Configure NTP using GPS Time

From IOS XE release 17.6.1, you can set up Network Time Protocol (NTP) to use GPS time as the reference clock using the **ntp refclock gps** command.

The cellular modem-based GPS works only with cellular and not with the GPS on Dead Reckoning module.

Before you begin

- Enable the cellular modem-based GPS and verify that it is working. To know more about cellular modem-based GPS, see Configuring GPS in the Cellular Pluggable Interface Module Configuration Guide.
- Verify that you have a location fix for this feature to work using the **show cellular** *cellualr slot* **gps** command.
- Enable National Marine Electronics Association (NMEA) streaming. For more details on enabling NMEA, see the Enable NMEA Data Streaming section in the Cellular Pluggable Interface Module Configuration Guide.
- Set up GPS time as a stratum 0 source and configure the Cisco IOS NTP server as a stratum 1 device to provide accurate time to NTP clients.

Procedure

Step 1 Use the **configure terminal** command to enter global configuration mode.

Example:

```
Router# configure terminal
```

Step 2 Use the **controller cellular** comamnd to configure cellular location.

Example:

```
Router(config)# controller cellular 0/4/0
```

Step 3 Use the **lte gps nmea** command to enable National Marine Electronics Association (NMEA) streaming.

Example:

```
Router(config-controller)# lte gps nmea
```

Step 4 Use the **end** command to exit NMEA configuration mode and return to privileged EXEC mode.

Example:

```
Router(config-controller-nmea)# end
```

Step 5 Use the **configure terminal** command to enter global configuration mode.

Example:

```
Router# configure terminal
```

Step 6 Use the **ntp refclock gps interface Cellular 0/4/0** commad to configure the NTP reference clock as GPS.

Example:

```
Router(config)# ntp refclock gps interface Cellular 0/4/0
```

Step 7 Use the **exit** command to exit from the configuration prompt.

Example:

```
Router(config)# exit
```

Step 8 (Optional) Use the **show cellular** *cellualr slot* **gps** command to view the GPS information.

Example:

```
Router# show cellular 0/4/0 gps
GPS Feature = enabled
GPS Mode Configured = standalone
Current Constellation Configured = gps
GPS Port Selected = Dedicated GPS port
GPS Status = GPS coordinates acquired
Last Location Fix Error = Offline [0x0]
Latitude = 50 Deg 36 Min 44.9907 Sec North
Longitude = 5 Deg 35 Min 12.0240 Sec East
Timestamp (GMT) = Tue Dec 17 17:06:05 2024

Fix type = 2D, Height = 64m
Satellite Info
----------------
Satellite #4, elevation 8, azimuth 329, SNR 31 *
Satellite #5, elevation 9, azimuth 88, SNR 37 *
Satellite #18, elevation 29, azimuth 170, SNR 48 *
Satellite #20, elevation 14, azimuth 54, SNR 30 *
Satellite #25, elevation 43, azimuth 111, SNR 40 *
Satellite #26, elevation 38, azimuth 292, SNR 34 *
Satellite #28, elevation 55, azimuth 227, SNR 44 *
Satellite #29, elevation 76, azimuth 67, SNR 38 *
Satellite #31, elevation 51, azimuth 281, SNR 29 *
Satellite #49, elevation 31, azimuth 180, SNR 46  **
Satellite #9, elevation 0, azimuth 0, SNR 0
Satellite #12, elevation 5, azimuth 106, SNR 0 *
Satellite #16, elevation 6, azimuth 288, SNR 0
```

Step 9 (Optional) Use the **show ntp associations** command to view the network time protocol associations.

Example:

```
Router# show ntp associations
  address         ref clock       st   when   poll reach  delay  offset   disp
x~127.127.5.1     .GPS.            0     58     64     3  0.000  -0.031 3938.4
*~45.87.78.35     244.220.219.218  2     51     64     3 13.985 495.081 64.377
 * sys.peer, # selected, + candidate, - outlyer, x falseticker, ~ configured
```

Step 10 (Optional) Use the **show clock** command to display the current time and clock details.

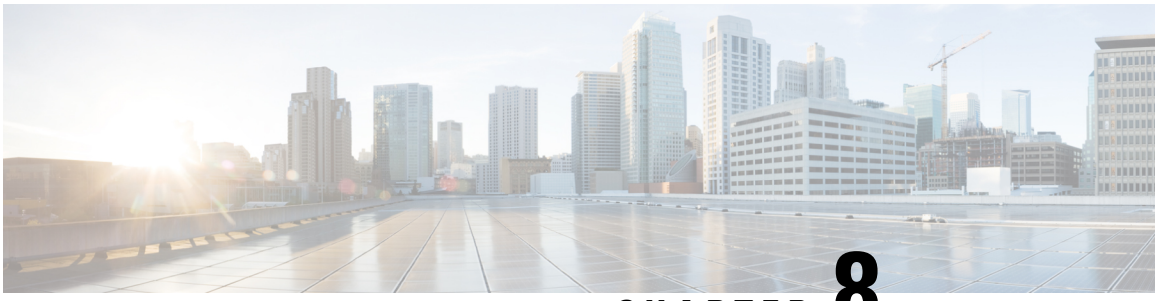
Example:

```
Router# show clock
17:12:56.870 UTC Tue Dec 17 2024
```

Step 11 (Optional) Use the **debug ntp refclock** command to troubleshoot the configuration.

Example:

```
Router# debug ntp refclock
*Sep 24 19:58:43.045 GMT: GPS: Poll Requested
*Sep 24 19:58:43.045 GMT: GPS (19:58:43.056 GMT Thu Sep 24 2020)
*Sep 24 19:58:43.045 GMT: Valid time rcvd from GPS: 2020/09/24 19:58:43.056 (frac =
0x0E560440)
*Sep 24 19:58:43.045 GMT: RTS poll timestamp (local clock) was 0xE31778F3.0B851ED8
*Sep 24 19:58:43.045 GMT: GPS timestamp is 0xE31778F3.0E560440
*Sep 24 19:58:43.045 GMT: NTP Core(NOTICE): ntpd PPM
*Sep 24 19:58:43.046 GMT: NTP Core(NOTICE): trans state : 5
*Sep 24 19:58:43.046 GMT: NTP Core(NOTICE): Clock is synchronized.
```



CHAPTER 8

Managing Configuration Files

This chapter contains the following sections:

- Understanding Configuration Files, on page 85
- Finding the Software Version, on page 86
- Managing and Configuring a Consolidated Package Using copy and boot Commands, on page 86
- Upgrading the Router Image through the WebUI, on page 88

Understanding Configuration Files

Configuration files contain the Cisco IOS XE software commands used to customize the functionality of your Cisco routing device (router, access server, switch, and so on). Commands are parsed (translated and executed) by the Cisco IOS XE software when the system is booted (from the startup-config file) or when you enter commands at the CLI in a configuration mode.

Types of Configuration Files

Startup configuration files (startup-config) are used during system startup to configure the software. Running configuration files (running-config) contain the current configuration of the software. The two configuration files can be different. For example, you may want to change the configuration for a short time period rather than permanently. In this case, you would change the running configuration using the configure terminal EXEC command but not save the configuration using the copy running-config startup-config EXEC command.

To change the running configuration, use the configure terminal command. As you use the Cisco IOS XE configuration modes, commands generally are executed immediately and are saved to the running configuration file either immediately after you enter them or when you exit a configuration mode.

To change the startup configuration file, you can either save the running configuration file to the startup configuration using the copy running-config startup-config EXEC command or copy a configuration file from a file server to the startup configuration.

Location of Configuration Files

Configuration files can be stored in the following locations:

- The running configuration is stored in RAM.
- The startup configuration is stored in the location specified by the CONFIG_FILE environment variable.

  The CONFIG_FILE variable defaults to NVRAM and can be a file in the following file systems:

- nvram: (NVRAM)
- bootflash: (Internal Flash memory)
- usbflash0: (external USB media)

Finding the Software Version

The package files for the Cisco IOS XE software can be found on the system board flash device (flash:) orone of the external devices previously mentioned.

You can use the **show version** privileged EXEC command to see the software version that is running on your device.

Note Although the **show version** output always shows the software image running on the device, the model name shown at the end of this display is the factory configuration and does not change if you upgrade the software license.

You can also use the **dir** *filesystem*: privileged EXEC command to see the directory names of other software images that you might have stored in flash memory.

Managing and Configuring a Consolidated Package Using copy and boot Commands

To upgrade a consolidated package, copy the consolidated package to the bootflash: directory on the router. After making this copy of the consolidated package, configure the router to boot using the consolidated package file.

The following example shows the consolidated package file being copied to the bootflash: file system. The config register is then set to boot using boot system commands, and the commands instruct the router to boot using the consolidated package stored in the bootflash: file system. The new configuration is then saved using the copy running-config startup-config command, and the system is then reloaded to complete the process.

Display the contents of the bootflash directory.

```
Router# dir bootflash:
Directory of bootflash:/
13      drwx           278528  May 19 2022 05:20:04 +00:00  tracelogs
11      drwx             4096  May 17 2022 14:24:54 +00:00  .installer
84      drwx            20480  May 17 2022 14:22:00 +00:00  license_evlog
83      -rw-               30  May 17 2022 14:21:41 +00:00  throughput_monitor_params
12      drwx             4096  May 17 2022 14:21:39 +00:00  .prst_sync
22      -rw-              335  May 17 2022 14:20:50 +00:00  boothelper.log
14      -rwx            41040  May 17 2022 14:20:39 +00:00  mode_event_log
259     -rw-        682679541  May 17 2022 12:54:32 +00:00
ir1800-universalk9.17.07.01.SPA.bin
```

Copy the new image into the bootflash: directory.

Note In order to use secure copy (scp), you must first set up an SSH configuration. See Configuring Secure Shell.

```
Router# copy scp: bootflash:
Address or name of remote host []? 192.168.1.2
Source username [xxxxx]?Enter
Source filename []? /auto/users/IR1800-universalk9.17.08.01.SPA.bin
Destination filename [IR1800-universalk9.17.08.01.SPA.bin]?

This is a Cisco managed device to be used only for authorized purposes.
Your use is monitored for security, asset protection, and policy compliance.

Password: <your-password>
Sending file modes: C0644 208904396 IR1800-universalk9.17.08.01.SPA.bin
...........
[OK - 208904396 bytes]
208904396 bytes copied in 330.453 secs (632176 bytes/sec)
```

Display the contents of the bootflash: directory.

```
Router# dir bootflash:
Directory of bootflash:/
13      drwx           278528  May 19 2022 05:20:04 +00:00  tracelogs
11      drwx             4096  May 17 2022 14:24:54 +00:00  .installer
84      drwx            20480  May 17 2022 14:22:00 +00:00  license_evlog
83      -rw-               30  May 17 2022 14:21:41 +00:00  throughput_monitor_params
12      drwx             4096  May 17 2022 14:21:39 +00:00  .prst_sync
22      -rw-              335  May 17 2022 14:20:50 +00:00  boothelper.log
14      -rwx            41040  May 17 2022 14:20:39 +00:00  mode_event_log
259     -rw-        682679541  May 17 2022 12:54:32 +00:00
ir1800-universalk9.17.07.01.SPA.bin
12      -rw-        208904396  May 17 2022 16:17:34 -07:00
ir1800-universalk9.17.08.01.SPA.bin
```

Configure the router to boot using the consolidated package file.

```
Router# configure terminal
Enter configuration commands, one per line. End with CNTL/Z.
Router(config)# boot system bootflash:ir1800-universalk9.17.08.01.SPA.bin
Router(config)# exit
```

Verify the configuration change.

```
Router# show run | include boot
boot-start-marker
boot system bootflash:IR1800-universalk9.17.08.01.SPA.bin
boot-end-marker
```

Copy the running configuration and save it. Then when reloading the router, it restarts with the saved configuration.

```
Router# copy running-config startup-config
Destination filename [startup-config]? <enter>
Building configuration...
[OK]

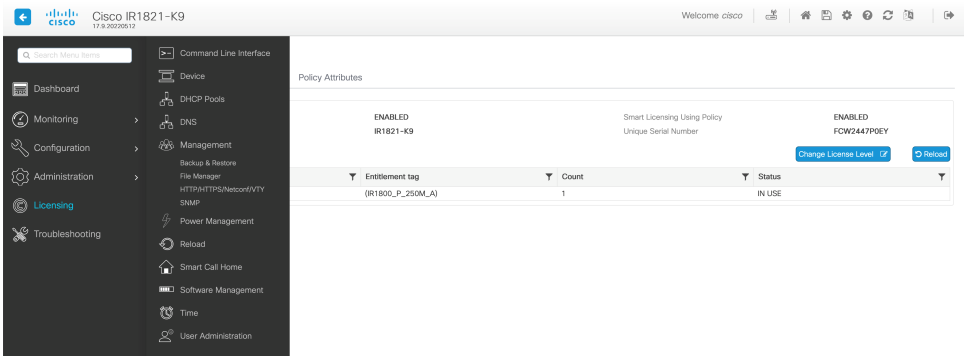
Router# reload
Proceed with reload? [confirm] <enter>
Dec 04 17:42:54.445 R0/0: %PMAN-5-EXITACTION: Process manager is exiting: process exit with
 reload

Initializing Hardware ...
```

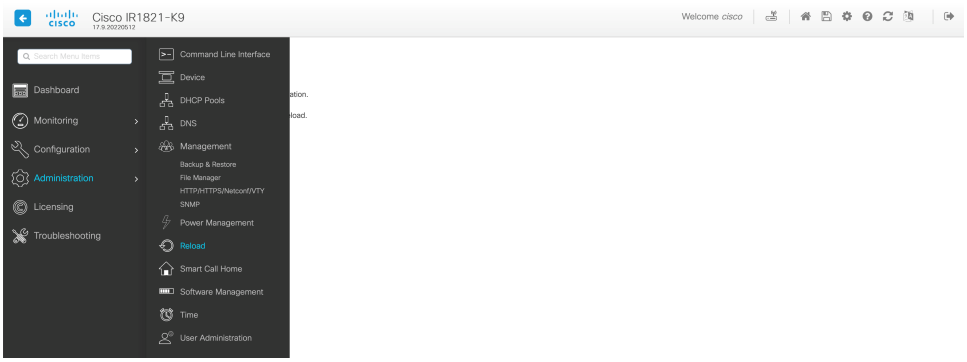
Upgrading the Router Image through the WebUI

The router can also be upgraded through the Web User Interface (WebUI). Further information on using the WebUI can be found in the Web User Interface (WebUI) chapter.

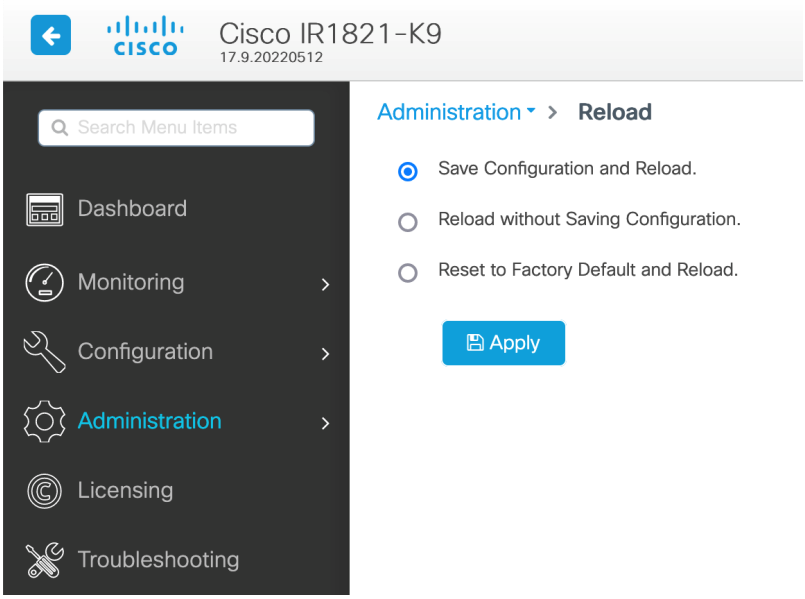
After you launch the WebUI, go to the **Administration** tab.



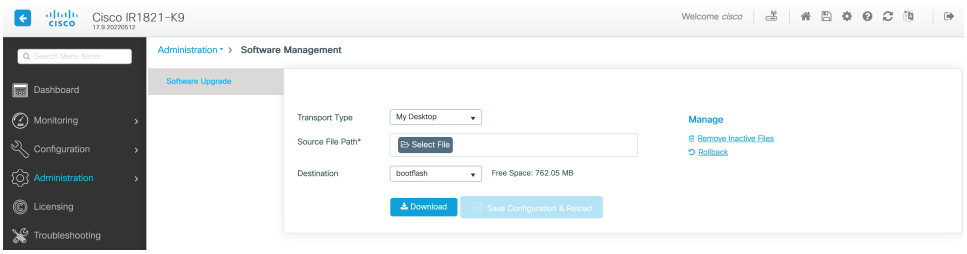
Reload the router by selecting **Administration > Reload**.



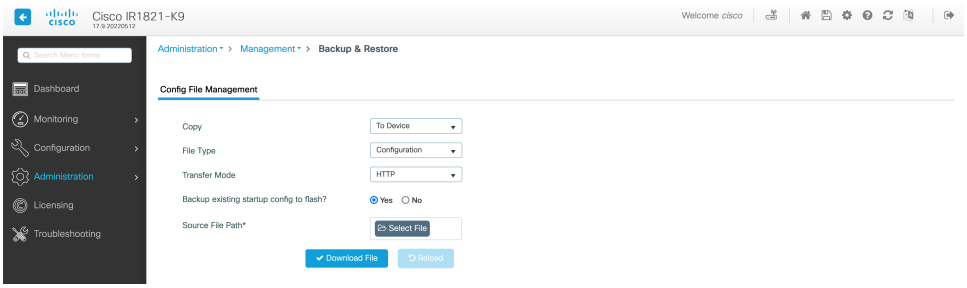
Select your option from the choices, then click **Apply**.



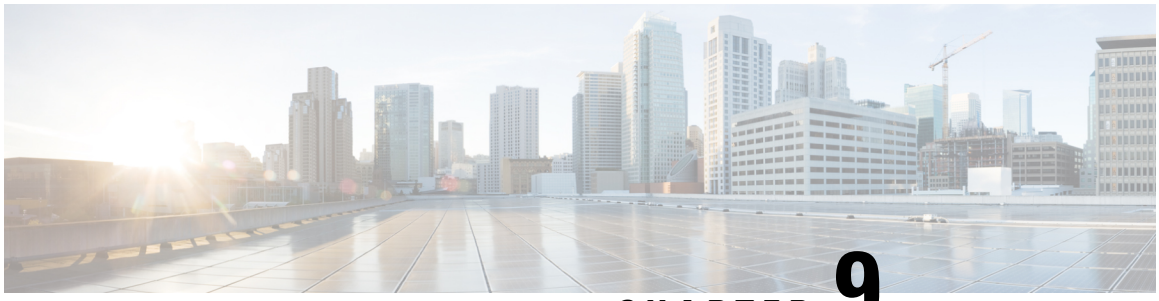
Select **Software Management** under the **Administration** tab. Browse to the location of the new IOS XE image file on your PC.



Select **Administration > Management > Backup & Restore**. Copy the image file from the laptop to your router. This example uses HTTP as transport.



Save the configuration by clicking on the floppy drive icon at the top of the WebUI.



CHAPTER 9

Using Cisco IOS XE Software

This chapter contains the following sections:

- Command modes, on page 91
- Use keyboard shortcuts, on page 93
- No and default command forms, on page 93
- Use the history buffer to recall commands, on page 94
- Configuration files management, on page 94
- Save configuration changes, on page 95
- Filtering output from show and more commands, on page 95
- Cisco feature navigator, on page 96
- Find support information for platforms and Cisco software images, on page 96
- Get help, on page 97
- Find command options, on page 97
- Software advisor, on page 101
- Use software release notes, on page 101

Command modes

The command modes available in Cisco IOS XE are the same as those available in traditional Cisco IOS. Use the CLI to access Cisco IOS XE software. Because the CLI is divided into many different modes, the commands available to you at any given time depend on the mode that you are currently in. Entering a question mark (**?**) at the CLI prompt allows you to obtain a list of commands available for each command mode.

When you log in to the CLI, you are in user EXEC mode. User EXEC mode contains only a limited subset of commands. To have access to all commands, you must enter privileged EXEC mode, normally by using a password. From privileged EXEC mode, you can issue any EXEC command—user or privileged mode—or you can enter global configuration mode. Most EXEC commands are one-time commands. For example, **show** commands show important status information, and **clear** commands clear counters or interfaces. The EXEC commands are not saved when the software reboots.

Configuration modes allow you to make changes to the running configuration. If you later save the running configuration to the startup configuration, these changed commands are stored when the software is rebooted. To enter specific configuration modes, you must start at global configuration mode. From global configuration mode, you can enter interface configuration mode and a variety of other modes, such as protocol-specific modes.

ROM monitor mode is a separate mode used when the Cisco IOS XE software cannot load properly. If a valid software image is not found when the software boots or if the configuration file is corrupted at startup, the software might enter ROM monitor mode.

Table 5: Accessing and exiting command modes

Command Mode	Access Method	Prompt	Exit Method
User EXEC	Log in.	`Router>`	Use the **logout** command.
Privileged EXEC	From user EXEC mode, use the **enable** command.	`Router#`	To return to user EXEC mode, use the **disable** command.
Global configuration	From privileged EXEC mode, use the **configure termina**l command.	`Router(config)#`	To return to privileged EXEC mode from global configuration mode, use the **exit** or **end** command.
Interface configuration	From global configuration mode, specify an interface using an **interface** command.	`Router(config-if)#`	To return to global configuration mode, use the **exit** command. To return to privileged EXEC mode, use the **end** command.
Diagnostic	The router boots up or accesses diagnostic mode in these scenarios: • In some cases, diagnostic mode will be reached when the Cisco IOS process or processes fail. In most scenarios, however, the router will reload. • A user-configured access policy is configured using the **transport-map** command that directs a user into diagnostic mode. • A break signal (**Ctrl-C**, **Ctrl-Shift-6**, or the **send break** command) is entered and the router is configured to go to diagnostic mode when the break signal is received.	`Router(diag)#`	If failure of the Cisco IOS process is the reason for entering diagnostic mode, the Cisco IOS problem must be resolved and the router rebooted to get out of diagnostic mode. If the router is in diagnostic mode because of a transport-map configuration, access the router through another port or by using a method that is configured to connect to the Cisco IOS CLI.
ROM monitor	From privileged EXEC mode, use the **reload** EXEC command. Press the **Break** key during the first 60 seconds while the system is booting.	`rommon#>`	To exit ROM monitor mode, manually boot a valid image or perform a reset with autoboot set so that a valid image is loaded.

Use keyboard shortcuts

Commands are not case sensitive. You can abbreviate commands and parameters if the abbreviations contain enough letters to be different from any other currently available commands or parameters.

Table 6: Keyboard shortcuts

Key Name	Purpose
Ctrl-B or the **Left Arrow** key	Move the cursor back one character.
Ctrl-F or the **Right Arrow** key	Move the cursor forward one character.
Ctrl-A	Move the cursor to the beginning of the command line.
Ctrl-E	Move the cursor to the end of the command line.
Esc B	Move the cursor back one word.
Esc F	Move the cursor forward one word.

No and default command forms

No and default command forms are CLI command variations that

- provide options to disable, enable, or reset configuration functions
- apply to almost every configuration command in Cisco IOS software, and
- allow administrators to modify command behavior without entering complete command syntax.

Command form functions

The **no** form is used to disable a function. Use the command without the **no** keyword to re-enable a disabled function or to enable a function that is disabled by default. For example, IP routing is enabled by default. To disable IP routing, use the **no IP routing** command; to re-enable IP routing, use the **IP routing** command. The Cisco IOS software command reference publications provide the complete syntax for the configuration commands and describe what the **no** form of a command does.

By issuing the **<command> default** command-name, you can configure the command to its default setting. The Cisco IOS software command reference publications describe the function from a **default** form of the command when the **default** form performs a different function than the plain and **no** forms of the command. To see what default commands are available on your system, enter **default ?** in the appropriate command mode.

Use the history buffer to recall commands

The history buffer stores the last 20 commands you entered. History substitution allows you to access these commands without retyping them, by using special abbreviated commands.

Table 7: History substitution commands

Command	Purpose
Ctrl-P or the **Up Arrow** key	Recalls commands in the history buffer, beginning with the most recent command. Repeat the key sequence to recall successively older commands.
Ctrl-N or the **Down Arrow** key[1]	Returns to more recent commands in the history buffer after recalling commands with **Ctrl-P** or the **Up Arrow** key.
Router# show history	While in EXEC mode, lists the last few commands you entered.

Configuration files management

Managing configuration files is a router maintenance process that:

- stores startup configuration files in the nvram: file system and running configuration files in the system: file system
- provides encryption of configuration files through IOS XE features, and
- enables backup of configuration files to prevent data loss.

Configuration file storage and management

This configuration file storage setup is also used on several other Cisco router platforms.

IOS XE provides encryption of the configuration file. Encryption is discussed in length in the IOS XE hardening device guide which can be found here:https://www.cisco.com/c/en/us/support/docs/ip/access-lists/13608-21.html

As a matter of routine maintenance on any Cisco router, users should back up the startup configuration file by copying the startup configuration file from NVRAM to one of the router's other file systems and, additionally, to a network server. Backing up the startup configuration file provides an easy method of recovering the startup configuration file if the startup configuration file in NVRAM becomes unusable for any reason.

The **copy** command can be used to back up startup configuration files.

For more detailed information on managing configuration files, see the "Managing Configuration Files" section in the Cisco IOS XE Configuration Fundamentals Configuration Guide.

Save configuration changes

Save configuration changes to the startup configuration so that the changes will not be lost if the software reloads or a power outage occurs.

This task saves the configuration to the NVRAM.

Procedure

Use the **copy running-config startup-config** command to save your configuration changes to the startup configuration.

Example:

```
Router# copy running-config startup-config
Destination filename [startup-config]? enter
Building configuration...
[OK]
Router#
*Sep 24 08:50:26.666: %SYS-6-PRIVCFG_ENCRYPT_SUCCESS: Successfully encrypted private config
 file
```

Note
It may take a few minutes to save the configuration.

The configuration is saved to the NVRAM and changes persist through software reloads and power outages.

Filtering output from show and more commands

Filtering output from show and more commands is a functionality that

- searches and filters the output of **show** and **more** commands
- helps you sort through large amounts of output, and
- excludes output that you need not see.

Command syntax

To use this functionality, enter a **show** or **more** command followed by the "pipe" character (|); one of the keywords **begin**, **include**, or **exclude**; and a regular expression on which you want to search or filter (the expression is case sensitive):

show *command* | **{append | begin | exclude | include | redirect | section | tee}** *regular-expression*

The output matches certain lines of information in the configuration file.

The example for show interface command with include protocol modifier is as given.

In this example, a modifier of the **show interface** command (**include protocol**) is used to provide only the output lines in which the expression **protocol** is displayed:

```
Router# show interface | include protocol
GigabitEthernet0/0/0 is administratively down, line protocol is down (disabled)
0 unknown protocol drops
GigabitEthernet0/1/0 is down, line protocol is down (notconnect)
0 unknown protocol drops
GigabitEthernet0/1/1 is down, line protocol is down (notconnect)
0 unknown protocol drops
GigabitEthernet0/1/2 is down, line protocol is down (notconnect)
0 unknown protocol drops
GigabitEthernet0/1/3 is down, line protocol is down (notconnect)
0 unknown protocol drops
GigabitEthernet0/0/5 is up, line protocol is up (connected)
0 unknown protocol drops
Cellular0/4/0 is up, line protocol is up
0 unknown protocol drops
Cellular0/4/1 is administratively down, line protocol is down
0 unknown protocol drops
Cellular0/5/0 is up, line protocol is up
0 unknown protocol drops
Cellular0/5/1 is administratively down, line protocol is down
0 unknown protocol drops
Async0/2/0 is up, line protocol is down
0 unknown protocol drops
Vlan1 is up, line protocol is up , Autostate Enabled
0 unknown protocol drops
Vlan172 is up, line protocol is down , Autostate Enabled
0 unknown protocol drops
Vlan175 is down, line protocol is down , Autostate Enabled
0 unknown protocol drops
IR1800#
```

Cisco feature navigator

Cisco Feature Navigator is a tool that:

- enables you to determine which Cisco IOS XE software images support a specific software release, feature set, or platform
- provides information about platform support and software image support, and
- does not require an account on Cisco.com to use.

Find support information for platforms and Cisco software images

The Cisco IOS XE software is packaged in feature sets consisting of software images that support specific platforms. The group of feature sets that are available for a specific platform depends on which Cisco software images are included in a release.

All of the Cisco IOS-XE configuration guides can be found here: https://www.cisco.com/c/en/us/support/ios-nx-os-software/ios-xe-17/series.html.

To identify the set of software images available in a specific release or to find out if a feature is available in a given Cisco IOS XE software image, you can use Cisco Feature Navigator or see the https://www.cisco.com/c/en/us/support/ios-nx-os-software/ios-xe-17/series.html.

Get help

Entering a question mark (**?**) at the CLI prompt displays a list of commands available for each command mode. You can also get a list of keywords and arguments associated with any command by using the context-sensitive help feature.

To get help that is specific to a command mode, a command, a keyword, or an argument, use one of these commands.

Command	Purpose
help	Provides a brief description of the help system in any command mode.
abbreviated-command-entry ?	Provides a list of commands that begin with a particular character string. **Note** There is no space between the command and the question mark.
abbreviated-command-entry <Tab>	Completes a partial command name.
?	Lists all the commands that are available for a particular command mode.
command ?	Lists the keywords or arguments that you must enter next on the command line. **Note** There is a space between the command and the question mark.

Find command options

This section provides information about how to display the syntax for a command. The syntax can consist of optional or required keywords and arguments. To display keywords and arguments for a command, enter a question mark (**?**) at the configuration prompt or after entering a part of a command followed by a space. The Cisco IOS XE software displays a list and brief descriptions of the available keywords and arguments. For example, if you are in global configuration mode and want to see all the keywords and arguments for the **arap** command, you should type **arap ?**.

The <**cr**> symbol in command help output stands for carriage return. On older keyboards, the carriage return key is the **Return** key. On most modern keyboards, the carriage return key is the **Enter** key. The <**cr**> symbol at the end of command help output indicates that you have the option to press **Enter** to complete the command and that the arguments and keywords in the list preceding the <**cr**> symbol are optional. The <**cr**> symbol by itself indicates that no more arguments or keywords are available, and that you must press **Enter** to complete the command.

Table 8: Finding command options

Command	Comment

`Router> enable`
`Password: <password>`
`Router#`
	Enter the **enable** command and password to access privileged EXEC commands. You are in privileged EXEC mode when the prompt changes to a " # " from the " > ", for example, **Router>** to **Router#**

`Router# configure terminal`
`Enter configuration commands, one per line. End with CNTL/Z.`
`Router(config)#`
	Enter the **configure terminal** privileged EXEC command to enter global configuration mode. You are in global configuration mode when the prompt changes to **Router (config)#**

`Router(config)# interface GigabitEthernet ?`
`<0-0> GigabitEthernet interface number`
 	
`Router(config)# interface GigabitEthernet 0/?`
`<0-5> Port Adapter number`
 	
`Router (config)# interface GigabitEthernet 0/0/?`
`<0-63> GigabitEthernet interface number`
 	
`Router (config)# interface GigabitEthernet 0/0/0?`
`.  <0-71>`
 	
`Router(config-if)#`
	Enter interface configuration mode by specifying the interface that you want to configure, using the **interface GigabitEthernet** global configuration command.
	
	Enter **?** to display what you must enter next on the command line.
	
	When the <cr> symbol is displayed, you can press **Enter** to complete the command.
	
	You are in interface configuration mode when the prompt changes to **Router(config-if)#**

Command	Comment

Router(config-if)# **?**
Interface configuration commands:
  .
  .
  .
  ip Interface Internet Protocol
  config commands
  keepalive Enable keepalive
  lan-name LAN Name command
  llc2 LLC2 Interface Subcommands
 	
  load-interval Specify interval for load calculation
  for an interface
  locaddr-priority Assign a priority group
  logging Configure logging for interface
  loopback Configure internal loopback on an
  interface
  mac-address Manually set interface MAC address
  mls mls router sub/interface commands
  mpoa MPOA interface configuration commands
  mtu Set the interface
  Maximum Transmission Unit (MTU)
  netbios Use a defined NETBIOS access list
  or enable
  name-caching
  no Negate a command or set its defaults
  nrzi-encoding Enable use of NRZI encoding
  ntp Configure NTP
  .
  .
  .
Router(config-if)#
	Enter **?** to display a list of all the interface configuration commands available for the interface. This example shows only some of the available interface configuration commands.

Command	Comment

`Router(config-if)# ip ?`
`Interface IP configuration subcommands:`
`  access-group  Specify access control for packets`
`  accounting  Enable IP accounting on this interface`
`  address  Set the IP address of an interface`
`  authentication  authentication subcommands`
`  bandwidth-percent  Set EIGRP bandwidth limit`
`  broadcast-address  Set the broadcast address of an interface`
`  cgmp  Enable/disable CGMP`
`  directed-broadcast  Enable forwarding of directed broadcasts`
`  dvmrp  DVMRP interface commands`
`  hello-interval  Configures IP-EIGRP hello interval`
`  helper-address  Specify a destination address for UDP broadcasts`
`  hold-time  Configures IP-EIGRP hold time`
`  .`
`  .`
`  .`
`Router(config-if)# ip`
	Enter the command that you want to configure for the interface. This example uses the **IP** command.
	Enter **?** to display what you must enter next on the command line. This example shows only some of the available interface IP configuration commands.

`Router(config-if)# ip address ?`
`  A.B.C.D  IP address`
`  negotiated  IP Address negotiated over PPP`
`Router(config-if)# ip address`
	Enter the command that you want to configure for the interface. This example uses the **IP address** command.
	Enter **?** to display what you must enter next on the command line. In this example, you must enter an IP address or the **negotiated** keyword.
	A carriage return (<cr>) is not displayed. Therefore, you must enter additional keywords or arguments to complete the command.

`Router(config-if)# ip address 172.16.0.1 ?`
`  A.B.C.D  IP subnet mask`
`Router(config-if)# ip address 172.16.0.1`
	Enter the keyword or argument that you want to use. This example uses the 172.16.0.1 IP address.
	Enter **?** to display what you must enter next on the command line. In this example, you must enter an IP subnet mask.
	<cr> is not displayed. Therefore, you must enter additional keywords or arguments to complete the command.

Command	Comment
Router(config-if)# ip address 172.16.0.1 255.255.255.0 ? secondary Make this IP address a secondary address <cr> Router(config-if)# ip address 172.16.0.1 255.255.255.0	Enter the IP subnet mask. This example uses the 255.255.255.0 IP subnet mask. Enter **?** to display what you must enter next on the command line. In this example, you can enter the **secondary** keyword, or you can press **Enter**. <cr> is displayed. Press **Enter** to complete the command, or enter another keyword.
Router(config-if)# ip address 172.16.0.1 255.255.255.0 Router(config-if)#	Press **Enter** to complete the command.

Software advisor

Software Advisor is a Cisco tool that:

- checks if a feature is supported in a Cisco IOS XE release
- locates the software document for that feature, and
- verifies the minimum software requirements of Cisco IOS XE software with the hardware installed on your router.

Software advisor access requirements

You must be a registered user on Cisco.com to access this tool. Cisco maintains the Software Advisor tool. See Tools and Resources.

Use software release notes

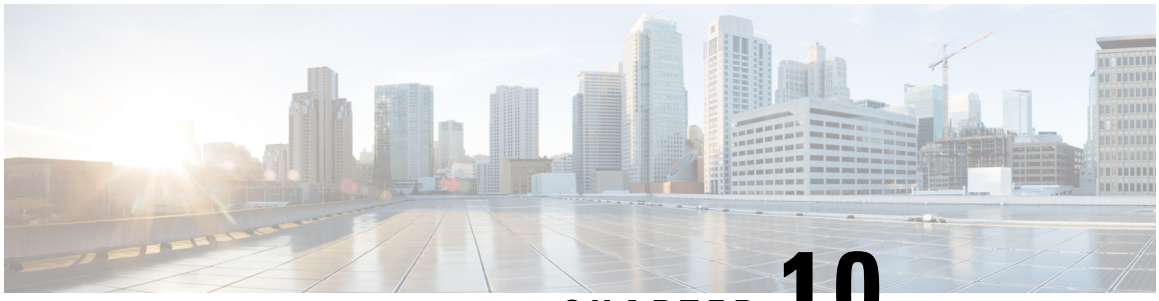
Software release notes provide essential information about the current release. This reference lists the types of information available in release notes and explains how to access cumulative feature information from previous releases.

Release notes contain information about these topics:

- Product overview
- Open and resolved severity 1 and 2 caveats
- Software image names
- New features
- Known limitations

Release notes are intended to be release-specific for the most current release, and the information provided in these documents may not be cumulative in providing information about features that first appeared in

previous releases. For cumulative feature information, refer to the Cisco Feature Navigator at: http://www.cisco.com/go/cfn/.



CHAPTER 10

Cisco IOS XE installation methods

Provides an overview of the supported procedures for installing Cisco IOS XE software on network devices.

- Bundle mode versus install mode, on page 103
- Installing software using install commands, on page 104
- Restrictions for installing the software using install commands, on page 104
- Install mode support, on page 104
- Information about installing the software using install commands, on page 105
- Configuration examples, on page 112
- Troubleshoot software installation using install commands, on page 119

Bundle mode versus install mode

Bundle mode and install mode are BOOT methods that:

- determine how Cisco IOS XE software loads on IoT routers
- use different file formats and storage locations for the operating system, and
- provide different levels of control over the software installation process.

BOOT mode characteristics

Bundle mode uses a single compressed image with the typical naming convention <product>-universalk9.<release>.SPA.bin. This mode provides a consolidated BOOT process, using local (hard disk, flash) or remote (TFTP) .bin image. Booting via a .bin image means that the router would first have to uncompress the image before booting from it. This led to a longer period of time for the router to BOOT.

To upgrade the router to a new version of IOS XE, you would point the "BOOT system" to a new software image. This method is well known and details are available in your products configuration guide.

Starting with IOS XE release 17.9.1, install mode has been added to the IoT routers. Install mode uses packages loaded into bootflash, which are read by a packages.conf file. This method provides more control over the software installation process.

Install mode requires more room in bootflash: for the files. The packages are slightly larger than the .bin images, and they vary per product in size.

Note If the `BOOT` variable points to `flash:packages.conf`, the device is in install mode. If it points to a `.bin` file, it is in bundle mode.

Installing software using install commands

Installing software using install commands is a software management method that enables users to boot the platform and upgrade or downgrade to Cisco IOS XE software versions using a set of **install** commands on Cisco IoT routers shipped in install mode by default from Cisco IOS XE 17.9.1.

Restrictions for installing the software using install commands

Be aware of these restrictions when installing software using install commands:

- Install mode requires a reboot of the system.
- SMU installation was supported in both bundle boot and install mode. From Cisco IOS XE Release 17.9.x, SMU installation will be stopped if the router is booted up in bundle mode. If the router is booted up in install mode, SMU installation will keep working as it is in previous releases.

Install mode support

This table describes the differences between Bundle mode and Install mode.

Cisco IOS XE running on IoT routers has typically made use of the Bundle boot mode. Bundle boot mode is also known as Consolidated boot, and uses a single compressed image. The typical naming convention is <product>-universalk9.<release>.SPA.bin.

This mode provides a consolidated boot process, using local (hard disk, flash) or remote (TFTP) .bin image. Booting via a .bin image means that the router would first have to uncompress the image before booting from it. This led to a longer period of time for the router to boot.

To upgrade the router to a new version of IOS XE, you would point the "boot system" to a new software image. This method is well known and details are available in your products configuration guide.

Starting with IOS XE release 17.9.1, a new boot mode called Install mode has been added to the IoT routers. Install mode uses packages loaded into bootflash, which are read by a packages.conf file. This method provides more control over the software installation process.

Note SMU installation was supported in both bundle boot and install mode. From Cisco IOS XE Release 17.9.x, SMU installation will be stopped if the router is booted up in bundle mode. If the router is booted up in install mode, SMU installation will keep working as it is in previous releases.

Table 9: 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.
CLI: `Router(config)#boot system bootflash:<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.
Image Auto-Upgrade: When a new Field-Replaceable Unit (FRU) is inserted in a modular chassis, manual intervention is required to get the new FRU running with the same version as the active FRUs.	Image Auto-Upgrade: When a new FRU is inserted in a modular chassis, the joining FRU is auto-upgraded to the image version in sync with the active FRUs.
Rollback: Rollback to the previous image with multiple Software Maintenance Updates (SMUs) may require multiple reloads.	Rollback: Enables rollback to an earlier version of Cisco IOS XE software, including multiple patches in single reload.

For additional information, please see Cisco IOS XE Installation Methods.

Information about installing the software using install commands

From the Cisco IOS XE 17.9.1 release, IoT routers will be shipped in install mode instead of bundle mode. So any new router from the factory will boot up in install mode.

Existing installations using previous releases of IOS XE have the option to continue to use their device in Bundle mode if they wish to. Or they can convert their device to Install mode.

Install mode is applicable to both autonomous mode and controller mode.

A new release can be installed in Install mode using vManage.

Table 10: 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: `Router(config)#boot system bootflash:<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.
Image Auto-Upgrade: When a new Field-Replaceable Unit (FRU) is inserted in a modular chassis, manual intervention is required to get the new FRU running with the same version as the active FRUs.	Image Auto-Upgrade: When a new FRU is inserted in a modular chassis, the joining FRU is auto-upgraded to the image version in sync with the active FRUs.
Rollback: Rollback to the previous image with multiple Software Maintenance Updates (SMUs) may require multiple reloads.	Rollback: Enables rollback to an earlier version of Cisco IOS XE software, including multiple patches in single reload.

Installing software using install mode commands

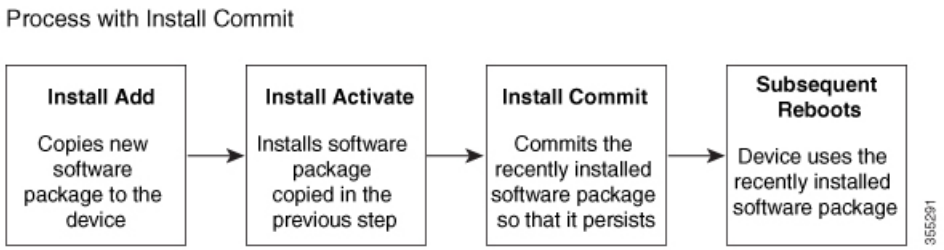
Summary

The key components involved in the install mode process are:

- **install add**: Copies software package from local or remote location to platform and validates the image file
- **install activate**: Performs validations and provisions packages previously added, triggering system reload
- **install commit**: Confirms packages previously activated and makes updates persistent over reloads

The install mode process flow comprises three commands to perform installation and upgrade of software on platforms. Installing an update replaces any previously installed software image, with only one image installed at any time.

Workflow



These stages describe how the install mode process works:

1. The system copies and validates the software package using the **install add** command.
   - 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

The **install add** command copies the software package from a local or remote location to the platform. 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 location of the software package can be in several places, as shown in the output of this command:

```
Router# install add file ?
bootflash: Package name
crashinfo: Package name
flash: Package name
ftp: Package name
http: Package name
https: Package name
pram: Package name
rcp: Package name
scp: Package name
sftp: Package name
tftp: Package name
webui: Package name
```

2. The system activates the package using the **install activate** command and triggers a reload.
   - 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

   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 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.

3. The system commits the activated package using the **install commit** command to make changes persistent.
   - Use the **show install summary** command to see which image is uncommitted. This image will get committed

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

Boot 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.

You can see how your device is set up to boot by using the **show romvar** and **show bootvar** commands.

```
Router# show romvar
ROMMON variables:
```

```
PS1 = rommon ! >
CM = IR1800
DEVICE_MANAGED_MODE = autonomous
LICENSE_SUITE =
RET_2_RTS =
THRPUT = 250
BOOT = flash:packages.conf,12;
LICENSE_BOOT_LEVEL = network-advantage,all:IR1800;
BSI = 0
RET_2_RCALTS =
RANDOM_NUM = 212626522
Router#

Router# show bootvar
BOOT variable = flash:packages.conf,12;
CONFIG_FILE variable does not exist
BOOTLDR variable does not exist
Configuration register is 0x2102

Standby not ready to show bootvar

Router#
```

Convert from bundle mode to install mode or upgrade the platform using one-step installation

Use the one-step install procedure 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, you can also use the one-step install procedure 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.

Note

- All the CLI actions (for example, add and activate) are executed.
- The configuration save prompt appears if an unsaved configuration is detected.
- The reload prompt appears 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.

Follow these steps to convert from bundle mode to install mode, or to upgrade the platform using the one-step installation.

Procedure

Step 1 Use the **enable** command to enable privileged EXEC mode. Enter your password if prompted.

Example:

```
Router> enable
```

Step 2 Use the **install add file** *location*: *filename* [**activate commit**] command to copy the software install package to the platform and extract its components, validate platform and image compatibility, activate the package, and commit it to make it persistent across reloads.

Example:

```
Router# install add file bootflash:<router_image>.SSA.bin activate commit
```

Note

- You can copy the software install package from a local or remote location (through FTP, HTTP, HTTPS, or TFTP).
- This command extracts the individual components of the .package file into subpackages and packages.conf files.
- The platform reloads after this command is run.

Step 3 Use the **exit** command to exit privileged EXEC mode and return to user EXEC mode.

Example:

```
Router# exit
```

The platform converts from bundle boot mode to install boot mode and reboots. The platform is now ready for future upgrades using the one-step install procedure.

Three-step installation

This procedure provides more flexibility and control during software installation by using the individual `install add`, `install activate`, and `install commit` commands to install a software package and upgrade the platform to a new version.

Note

- You can use the three-step installation procedure only after the platform is in install mode.
- All the CLI actions (for example, add and activate) are executed.
- The configuration save prompt appears if an unsaved configuration is detected.
- The reload prompt appears after the `install activate` step in this workflow. Use the `prompt-level none` keyword to automatically ignore the confirmation prompts.

Follow these steps to perform a three-step installation.

Procedure

Step 1 Use the **enable** command to enable privileged EXEC mode. Enter your password if prompted.

Example:

```
Router> enable
```

Step 2 Use the **install add file** *location*: *filename* command to copy the software install package to the platform and extract the individual components of the .package file into subpackages and packages.conf files.

Example:

```
Router# install add file bootflash:<router_image>.SSA.bin
```

Note
You can copy the software install package from a remote location (through FTP, HTTP, HTTPS, or TFTP).

Step 3 (Optional) Use the **show install summary** command to display an overview of the image versions and their corresponding install state.

Example:

```
Router# show install summary
```

Step 4 Use the **install activate auto-abort-timer** *time* command to activate the previously added package and reload the platform.

Example:

```
Router# install activate auto-abort-timer 120
```

Note

- When you do a full software install, do not provide a package filename.
- In the three-step variant, the 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 (Optional) Use the **install abort** command to terminate the software install activation and return the platform to the last committed version.

Example:

```
Router# install abort
```

Note
Use this command only when the image is in the activated state, not when the image is in the committed state.

Step 6 Use the **install commit** command to commit the new package installation and make the changes persistent across reloads.

Example:

```
Router# install commit
```

Step 7 (Optional) Use the **install rollback to committed** command to roll back the platform to the last committed state.

Example:

```
Router# install rollback to committed
```

Step 8 (Optional) Use the **install remove** {**file** *filesystem*: *filename* | **inactive**} command to delete software installation files.

Example:

```
Router# install remove inactive
```

Note

- `file` — Deletes a specific file.
- `inactive` — Deletes all the unused and inactive installation files.

Step 9 (Optional) Use the **show install summary** command to display information about the current state of the system.

Example:

```
Router# show install summary
```

Note
The output of this command varies according to the install commands run prior to this command.

a) Use the **exit** command to exit privileged EXEC mode and return to user EXEC mode.

Example:

```
Router# exit
```

The software package is successfully installed and activated on the platform. The installation is committed and persistent across reloads.

Upgrading in install mode

Use either the one-step installation or the three-step installation to upgrade the platform in install mode.

Downgrading in install mode

Downgrading in install mode is a platform management process that

- uses the **install rollback** command to revert the platform to a previous version by pointing it to the appropriate image
- requires that the target image was previously installed in install mode
- reloads the platform and boots it with the previous image.

Downgrade requirements and alternatives

The **install rollback** command has specific requirements and alternatives:

- The 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

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

Software installation termination

Software installation termination is a process that:

- stops the activation of a software package before completion,
- returns the platform to the previously running software version, and
- prevents the new software image from being committed to the system.

Termination methods

You can terminate the activation of a software package in these 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

This section shows examples of using install commands.

One step installation

This example demonstrates the one-step installation process or converting from bundle mode to install mode.

```
Router#
install add file flash:ir1800-universalk9.SSA.bin activate commit



install_add_activate_commit: START Mon May 30 20:45:11 UTC 2022
install_add: Adding IMG
--- Starting initial file syncing ---
Copying flash:ir1800-universalk9.SSA.bin from  R0 to  R0
Info: Finished copying to the selected
Finished initial file syncing

--- Starting Add ---
Performing Add on all members
 [1] Finished Add package(s) on  R0
Checking status of Add on [R0]
Add: Passed on [R0]
Finished Add

Image added. Version: 17.09.01.0.157857

install_activate: Activating IMG
Following packages shall be activated:
/flash/ir1800-mono-universalk9.SSA.pkg
```

```
/flash/ir1800-rpboot.SSA.pkg

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


--- Starting Activate ---
Performing Activate on all members
Building configuration...
[OK] [1] Activate package(s) on  R0
 [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
 [1] Finished Commit on  R0
Checking status of Commit on [R0]
Commit: Passed on [R0]
Finished Commit operation

SUCCESS: install_add_activate_commit Mon May 30 20:48:01 UTC 2022
%PMAN-5-EXITACTION: R0/0: pvp: Process manager is exiting: reload action requested
watchdog: watchdog0: watchdog did not stop!
reboot: Restarting system


System Bootstrap, Version 3.3(REL), RELEASE SOFTWARE
Copyright (c) 1994-2021  by cisco Systems, Inc.

IR1800-K9 platform with 4169728 Kbytes of main memory

MCU Version - Bootloader: 4, App: 6
MCU is in application mode.


........


Loading: bootflash:packages.conf
#

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

%BOOT-5-OPMODE_LOG: R0/0: binos: System booted in AUTONOMOUS mode
Press RETURN to get started!

Router# 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    17.09.01.0.157857

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

Complete three-step software installation

The three-step installation process provides a controlled method for upgrading router software by allowing verification at each stage before proceeding to the next step. Use this installation method when you need to upgrade router software with the ability to verify and roll back at each stage if needed.

Follow these steps to complete the three-step software installation.

Procedure

Step 1 Use the **install add file** *filesystem:filename* command to add the software image to the router.

Example:

```
Router# install add file flash:ir1101-universalk9.17.09.01.SPA.bin
```

Step 2 (Optional) Use the **show install summary** command to verify that the image is added and in the Inactive (I) state.

Example:

```
Router# 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   I    17.09.01.0.1
           --------------------------------------------------------------------------------

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

Step 3 Use the **install activate** command to activate the software image. When prompted, enter **y** to proceed with the reload.

Example:

```
Router# install activate
```

Step 4 (Optional) After the system reloads, use the **show install summary** command to verify that the image is in the Activated & Uncommitted (U) state.

Example:

```
Router# 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   U    17.09.01.0.1
           --------------------------------------------------------------------------------
```

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

Step 5 Use the **install commit** command to commit the software installation.

Example:

```
Router# install commit
```

Step 6 (Optional) Use the **show install summary** command to verify that the image is in the Activated & Committed (C) state.

Example:

```
Router# 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    17.09.01.0.1
           ---------------------------------------------------------------------------

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

The software image is successfully installed and committed. The install summary shows the image state as "C" (Activated & Committed), confirming that the installation is complete and permanent.

Example

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

This example shows adding the image for various platforms:

```
install add file flash:ir1101-universalk9.17.09.01.SPA.bin
        install add file flash:ir1800-universalk9.17.09.01.SPA.bin
        install add file flash:ir8140-universalk9.17.09.01.SPA.bin
        install add file flash:ir8340-universalk9.17.09.01.SPA.bin
        install add file flash:c6300-universalk9.17.09.01.SPA.bin
```

This example shows the output of the add operation:

```
install_add: START Tue May 31 01:35:40 UTC 2022
        install_add: Adding IMG
        --- Starting initial file syncing ---
        Copying flash:ir1101ir1800ir8140ir8340c6300-universalk9.17.09.01.SPA.bin from R0
to R0
        Info: Finished copying to the selected
        Finished initial file syncing

        --- Starting Add ---
        Performing Add on all members
        [1] Finished Add package(s) on R0
        Checking status of Add on [R0]
        Add: Passed on [R0]
        Finished Add

        Image added. Version: 17.09.01.0.1
```

```
        SUCCESS: install_add
/flash1/ir1101ir1800ir8140ir8340c6300-universalk9.17.09.01.SPA.bin Tue May 31 01:37:10 UTC
 2022
        Router#
```

This example shows the output of the activate operation:

```
install_activate: START Tue May 31 01:37:14 UTC 2022
        install_activate: Activating IMG
        Following packages shall be activated:
        /flash/ir1101ir1800ir8140ir8340c6300-mono-universalk9_iot.17.09.01.SPA.pkg
        /flash/ir1101ir1800ir8140ir8340c6300-rpboot.17.09.01.SPA.pkg

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

        --- Starting Activate ---
        Performing Activate on all members
        [1] Activate package(s) on R0
        [1] Finished Activate on R0
        Checking status of Activate on [R0]
        Activate: Passed on [R0]
        Finished Activate

        SUCCESS: install_activate Tue May 31 01:41:03 UTC 2022
        Router#
        May 31 01:41:08.684: %PMAN-5-EXITACTION: R0/0: pvp: Process manager is exiting:
reload action requested

        watchdog: watchdog0: watchdog did not stop!
        reboot: Restarting system

        System Bootstrap, Version 3.3(REL), RELEASE SOFTWARE
        Copyright (c) 1994-2021 by cisco Systems, Inc.

        IR1101IR1800IR8140IR8340C6300-K9 platform with 4169728 Kbytes of main memory

        MCU Version - Bootloader: 4, App: 6
        MCU is in application mode.

        ........

        Loading: bootflash:packages.conf
        #

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

        Press RETURN to get started!
```

This example shows the output of the commit operation:

```
install_commit: START Tue May 31 01:47:56 UTC 2022
        --- Starting Commit ---
        Performing Commit on all members
        [1] Commit packages(s) on R0
        [1] Finished Commit packages(s) on R0
        Checking status of Commit on [R0]
        Commit: Passed on [R0]
        Finished Commit operation

        SUCCESS: install_commit Tue May 31 01:48:04 UTC 2022
```

Show the installed packages

Use these commands to display detailed information about installed packages on your router and identify which package is currently active.

Use the following command to display detailed information about an installed package:

```
Router#
show install package flash:ir1800-universalk9.17.09.01.SPA.bin


  Package: ir1800-universalk9.17.09.01.SPA.bin
    Size: 674114352
    Timestamp:
  Canonical path: /flash1/ir1800-universalk9.17.09.01.SPA.bin

    Raw disk-file SHA1sum:
      e54ba5a59824156af7515eaf4367ebe51b920316
  Header size:     1148 bytes
  Package type:    30000
  Package flags:   0
  Header version:  3

  Internal package information:
    Name: rp_super
    BuildTime: 2022-04-27_00.47
    ReleaseDate: 2022-04-27_07.05
    BootArchitecture: arm64
    RouteProcessor: IR1800
    Platform: IR1800
    User: mcpre
    PackageName: universalk9
    Build: 17.09.01
    CardTypes:

  Package is bootable from media and tftp.
  Package contents:

  Package: ir1800-mono-universalk9_iot.17.09.01.SPA.pkg
    Size: 673776700
    Timestamp:

    Raw disk-file SHA1sum:

    Header size:     1084 bytes
    Package type:    30000
    Package flags:   0
    Header version:  3

    Internal package information:
      Name: mono
      BuildTime: 2022-04-27_00.47
      ReleaseDate: 2022-04-27_07.05
      BootArchitecture: arm64
      RouteProcessor: IR1800
      Platform: IR1800
      User: mcpre
      PackageName: mono-universalk9_iot
      Build: 17.09.01
      CardTypes:

    Package is bootable from media and tftp.
```

```
    Package contents:
```

You can determine which package is active using the **show install active** command.

```
Router# 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 17.09.01.0.1193

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

Show committed and uncommitted packages

These two show commands provide information on which packages are committed and uncommited.

```
Router# 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    17.09.01.0.1

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

Router# 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
```

Remove inactive packages

This command will remove unused installation files (.conf/.pkg/.bin) from installation media.



Note This command is used to clean up the boot directory of unused installation files. This will not remove the bootable image.

```
Router#install remove inactive
install_remove: START Tue May 31 01:49:10 UTC 2022
install_remove: Removing IMG
Cleaning up unnecessary package files
No path specified, will use booted path /bootflash/packages.conf

Cleaning /flash
```

```
  Scanning boot directory for packages ... done.
  Preparing packages list to delete ...
    [R0]: /flash/packages.conf File is in use, will not delete.
    [R0]: /flash/ir1800-mono-universalk9_iot.17.09.01.SPA.pkg File is in use, will not
delete.
    [R0]: /flash/ir1800-universalk9.17.09.01.SPA.conf File is in use, will not delete.
    [R0]: /flash/ir1800-rpboot.17.09.01.SPA.pkg File is in use, will not delete.

The following files will be deleted:
    [R0]: /flash/ir1800-universalk9.17.09.01.SPA.bin
    [R0]: /flash/ir1800-mono-universalk9_iot.SSA.pkg
    [R0]: /flash/ir1800-universalk9.SSA.conf
    [R0]: /flash/ir1800-rpboot.SSA.pkg

Do you want to remove the above files? [y/n]y


Deleting file /flash/ir1800-universalk9.17.09.01.SPA.bin ... done.
Deleting file /flash/ir1800-mono-universalk9_iot.SSA.pkg ... done.
Deleting file /flash/ir1800-universalk9.SSA.conf ... done.
Deleting file /flash/ir1800-rpboot.SSA.pkg ... done.
Deleting /bootflash/.images/17.09.01.0.1.1651045630 ... done.
SUCCESS: Files deleted.

--- Starting Post_Remove_Cleanup ---
Performing REMOVE_POSTCHECK on all members
Finished Post_Remove_Cleanup
SUCCESS: install_remove Tue May 31 01:49:14 UTC 2022

Router#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
```

Troubleshoot software installation using install commands

Troubleshooting the software installation and resolving other installation issues.

Software installation troubleshooting commands

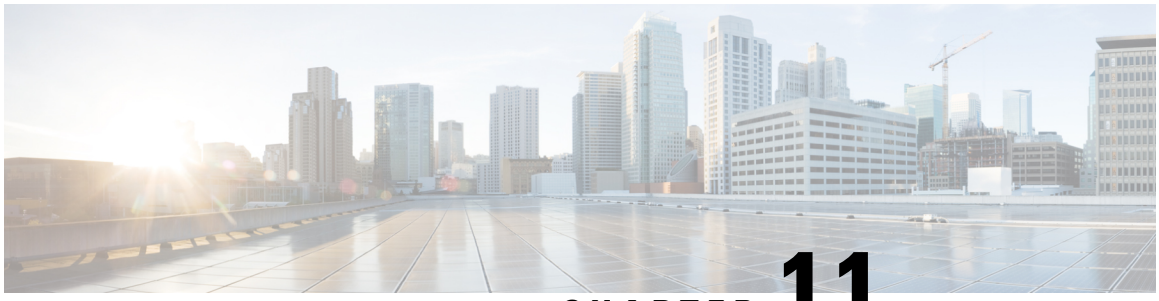
Use these show commands to view installation summary, logs, and software versions.

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

Other installation issue resolution

Use these 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 11

Installing the Software

This chapter contains the following sections:

- Installing the Software, on page 121
- IOS XE downgrade warning, on page 128
- Enable the secure data wipe capability, on page 128

Installing the Software

Installing software on the router involves installing a consolidated package (bootable image). This consists of a bundle of subpackages (modular software units), with each subpackage controlling a different set of functions.

It is better to upgrade software in a planned period of maintenance when an interruption in service is acceptable. The router needs to be rebooted for a software upgrade to take effect.

Licensing

This section contains the following:

Cisco Software Licensing

Cisco software licensing consists of processes and components to activate Cisco IOS software feature sets by obtaining and validating Cisco software licenses.

You can enable licensed features and store license files in the bootflash of your router. Licenses pertain to consolidated packages, technology packages, or individual features.

The IR1800 uses Enhanced Smart Licensing, which is discussed in detail in the next chapter.

Consolidated Packages

To obtain software images for the router, go to: https://software.cisco.com/download/home/286200112

Note All of the IOS-XE feature set may not apply to the IR1800. Some features may not have been implemented yet, or are not appropriate for this platform.

An image-based license is used to help bring up all the subsystems that correspond to a license. This license is enforced only at boot time.

One of the following image-based licenses can be pre-installed on the IR1800 router:

- Network-Essentials
- Network-Advantage

Note Details of the Network-Essentials and Network-Advantage contents can be found in the product data sheet.

Network-Essentials

The **Network-Essentials** technology package includes the baseline features. It also supports security features.

The **Network-Essentials_npe** technology package (npe = No Payload Encryption) includes all the features in the Network-Essentials technology package without the payload encryption functionality. This is to fulfill export restriction requirements. The Network-Essentials_npe is available only in the Network-Essentials_npe image. The difference in features between the Network-Essentials package and the Network-Essentials_npe package is therefore the set of payload encryption features such as IPsec and Secure VPN.

Network-Advantage

The **Network-Advantage** technology package includes all crypto features.

The **Network-Advantage_npe** package (npe = No Payload Encryption) includes all the features in the **Network-Advantage** technology package without the payload-encryption functionality. This is to fulfill export restriction requirements. The **Network-Advantage_npe** package is available only in the **Network-Advantage_npe** image. The difference in features between the **Network-Advantage** package and the **Network-Advantage_npe** package is therefore the set of payload-encryption-enabling features such as IPsec and Secure VPN.

How to Install the Software for Cisco IOS XE

To install the software, use one of the following methods described in this section to use the software from a consolidated package or an individual package.

Installing the Cisco IOS XE Release

When the device boots up with Cisco IOS XE image for the first time, the device checks the installed version of the ROMMON, and upgrades if the system is running an older version. During the upgrade, do not power cycle the device. The system automatically power cycles the device after the new ROMMON is installed. After the installation, the system will boot up with the Cisco IOS XE image as normal.

Note When the device boots up for first time and if the device requires an upgrade, the entire boot process may take several minutes. This process will be longer than a normal boot due to the ROMMON upgrade.

The following example illustrates the boot process of a consolidated package:

```
Router# configure terminal
 Router(config)#boot system bootflash:/ir1800-universalk9.17.06.01prd18.SPA.bin
Router(config)#config-register 0x2102
Router(config)#exit
*Nov  7 00:07:06.784: %SYS-5-CONFIG_I: Configured from console by console

Router#
Router#show run | inc license
license udi pid IR1800-K9 sn FCW2150TH0F
license boot level network-advantage
Router#

Router#reload ?
  /noverify  Don't verify file signature before reload.
  /verify    Verify file signature before reload.
  at         Reload at a specific time/date
  cancel     Cancel pending reload
  in         Reload after a time interval
  pause      Pause during reload
  reason     Reload reason
  <cr>       <cr>

Router#reload /verify

System configuration has been modified. Save? [yes/no]: yes
Building configuration...

[OK]
*Nov  7 00:08:48.101: %SYS-2-PRIVCFG_ENCRYPT: Successfully encrypted private config file
Verifying file integrity of bootflash:/ir1800-universalk9.17.06.01prd18.SPA.bin.....
.................................

Embedded Hash   SHA1 : B0315BDC4F545D624BB128CE0FFAA468E6EF7587
Computed Hash   SHA1 : B0315BDC4F545D624BB128CE0FFAA468E6EF7587
Starting image verification
Hash Computation:    100%Done!
Computed Hash   SHA2: 03febcc07fbeadeed664f2f5ef87f6c3
                      5b343e6f7aecdd70e50e5203909aec8f
                      3d276529d2a6af6859d4c77237f812d5
                      0da93678edc942c8874edca2d5224101

Embedded Hash   SHA2: 03febcc07fbeadeed664f2f5ef87f6c3
                      5b343e6f7aecdd70e50e5203909aec8f
                      3d276529d2a6af6859d4c77237f812d5
                      0da93678edc942c8874edca2d5224101

Digital signature successfully verified in file bootflash:/ir1800-universalk9.16.10.01.SPA.bin
Signature Verified

Proceed with reload? [confirm]<Enter>

*Jul  9 06:43:37.910: %SYS-5-RELOAD: Reload requested by console. Reload Reason: Reload
Command. Jul  9 14:43:59.134: %PMAN-5-EXITACTION: R0/0: pvp: Process manager is exiting:
process exit with reload chassis code

watchdog watchdog0: watchdog did not stop!
reboot: Restarting system



Press RETURN to get started!
```

ROMMON Images

A ROMMON image is a software package used by ROM Monitor (ROMMON) software on a router. The software package is separate from the consolidated package normally used to boot the router.

An independent ROMMON image (software package) may occasionally be released and the router can be upgraded with the new ROMMON software. For detailed instructions, see the documentation that accompanies the ROMMON image.

Note A new version of the ROMMON image is not necessarily released at the same time as a consolidated package for a router.

File Systems

The following table provides a list of file systems that can be seen on the Cisco IR1800 router.

Table 11: Router File Systems

File System	Description
bootflash:	Boot flash memory file system.
flash:	Alias to the boot flash memory file system above.
cns:	Cisco Networking Services file directory.
nvram:	Router NVRAM. You can copy the startup configuration to NVRAM or from NVRAM.
obfl:	File system for Onboard Failure Logging (OBFL) files.
system:	System memory file system, which includes the running configuration.
tar:	Archive file system.
tmpsys:	Temporary system files file system.
usbflash0:	The Universal Serial Bus (USB) flash drive file systems. **Note** The USB flash drive file system is visible only if a USB drive is installed in the usb port. **Note** Only Cisco supported USB flash drives may be used. A list of supported devices are found in the Hardware Installation Guide

Use the **?** help option if you find a file system that is not listed in the table above.

Option to Enable or Disable USB Access

USB flash drives offer inexpensive and easy storage space for the routers to store the images, configuration files and other files.

Note The IR1800 supports ext2 and vfat file systems for USB flash drives. Only Cisco approved USB Flash drives may be used.

The IR1800 supports hot plug/unplug of USB flash drives. To access the USB flash drive, insert the device into Router's USB interface. Once the USB is recognized, an alert message is seen on the console:

```
Aug  1 11:08:53.198 PDT: %IOSD_INFRA-6-IFS_DEVICE_OIR: Device usbflash0 added
```

After this message is seen, the USB flash drive is accessible. Users can access the USB contents using the **dir usbflash0:** command:

```
Device#dir usbflash0:
Directory of usbflash0:/
    5  drwx              512  Aug 23 2019 10:42:18 -07:00  System Volume Information
    6  -rwx               35  Aug 27 2019 17:40:38 -07:00  test.txt
206472192 bytes total (206470144 bytes free)
Device#
```

Contents can be copied to and from the USB flash drive using the copy command. Once the copy is complete, a log message showing number of bytes copied is displayed.

```
Device#copy flash:test.txt usbflash0:
Destination filename [test.txt]? <Enter>
Copy in progress...C
35 bytes copied in 0.020 secs (1750 bytes/sec)
Device#
```

While hot plug/unplug of a USB flash drive is supported, the functionality comes with security vulnerabilities. To prevent users from copying sensitive information to the USB flash drive, USB enable/disable functionality has been added.

By default, the USB flash drive is enabled. If a user wishes to disable USB, they can do so using the disable command:

```
Device# config terminal
Device(config)#platform usb disable

Device(config)#end
```

Once the USB flash drive has been disabled, the file system is not shown on the Device and syslog messages will not be displayed when the USB is inserted. Users will not be able to access the contents of the USB.

For example:

```
Device#dir usbflash0:
dir usbflash0:
    ^
% Invalid input detected at '^' marker.
Device#
```

The USB is enabled by issuing a '**no**' with the disable command:

```
Device#config terminal

Device(config)#no platform usb disable
Device(config)#end
```

The USB status can be displayed using the following command:

```
Device#show platform usb status
USB enabled
Device#
```

The USB port could be considered a potential security risk. If you wish to disable the USB port, use these steps:

```
Configure terminal
platform usb disable
exit

show platform usb
```

Autogenerated File Directories and Files

This section discusses the autogenerated files and directories that can be created, and how the files in these directories can be managed.

Table 12: Autogenerated Files

File or Directory	Description
crashinfo files	Crashinfo files may appear in the bootflash: file system. These files provide descriptive information of a crash and may be useful for tuning or troubleshooting purposes. However, the files are not part of router operations, and can be erased without impacting the functioning of the router.
core directory	The storage area for .core files. If this directory is erased, it will automatically regenerate itself at bootup. The .core files in this directory can be erased without impacting any router functionality, but the directory itself should not be erased.
managed directory	This directory is created on bootup if a system check is performed. Its appearance is completely normal and does not indicate any issues with the router.
tracelogs directory	The storage area for trace files. Trace files are useful for troubleshooting. If the Cisco IOS process fails, for instance, users or troubleshooting personnel can access trace files using diagnostic mode to gather information related to the Cisco IOS failure. Trace files, however, are not a part of router operations, and can be erased without impacting the router's performance.

Important Notes About Autogenerated Directories

Important information about autogenerated directories include:

- Autogenerated files on the bootflash: directory should not be deleted, renamed, moved, or altered in any way unless directed by Cisco customer support.



Note Altering autogenerating files on the bootflash: may have unpredictable consequences for system performance.

- Crashinfo files and files in the core and tracelogs directory can be deleted.

Flash Storage

Subpackages are installed to local media storage, such as flash. For flash storage, use the **dir bootflash:** command to list the file names.



Note Flash storage is required for successful operation of a router.

LED Indicators

For information on LEDs on the router, see "LED Indicators" in the "Product Overview" section of the Hardware Installation Guide.

To monitor the LED status of the system, the alarm and interface ports, the show LED command line is supported in IOS mode.

```
Router# show LED
SYSTEM LED : Green

GigabitEthernet0/0/0 LED : On
GigabitEthernet0/1/0 LED : Off
GigabitEthernet0/1/1 LED : Off
GigabitEthernet0/1/2 LED : Off
GigabitEthernet0/1/3 LED : Off

*Cellular 0/4*
LTE module Enable LED : Green
LTE module SIM 0 LED : Green
LTE module SIM 1 LED : Yellow
LTE module GPS LED : Off
LTE module RSSI 0 LED : On
LTE module RSSI 1 LED : On
LTE module RSSI 2 LED : On
LTE module RSSI 3 LED : On

*Cellular 0/5*
LTE module Enable LED : Green
LTE module SIM 0 LED : Green
LTE module SIM 1 LED : Off
LTE module GPS LED : Off
LTE module RSSI 0 LED : On
```

```
LTE module RSSI 1 LED : On
LTE module RSSI 2 LED : On
LTE module RSSI 3 LED : Off
Router#
```

Related Documentation

For further information on software licenses, see the Smart Licensing chapter.

IOS XE downgrade warning

An IOS XE downgrade warning is a system feature that

- presents a warning when issuing a **boot system flash** command followed by a file name of an image which has a version number lower than the one of the running image
- allows the downgrade operation by ignoring the warning message presented to the user
- permits booting an image with the same or higher version of the running image without warning, and
- applies only to images already loaded on the bootflash of the router using the **boot system flash**<*file_name*> CLI (excluding other sources/devices like ftp, mop, rpc, tftp, rom).

Version comparison examples

The system compares versions using these rules:

When comparing two version numbers as follows:

- 17.7.1
- 17.7.1c

The version with the letter (17.7.1c) will be considered the most updated one.

When comparing two version numbers as follows:

- 17.7.3a
- 17.7.3f

The comparison will be made taking into consideration the alphabetical order. In the case above 17.7.3f will be considered the most updated one.

Enable the secure data wipe capability

Use this task to configure secure data wipe on IoT platforms to ensure storage devices are properly purged using NIST SP 800-88r1 compliant secure erase commands.

Secure data wipe is a Cisco-wide initiative to ensure that storage devices on all IOS XE based platforms are properly purged using NIST SP 800-88r1 compliant secure erase commands. Whenever possible, IoT platforms leverage the corresponding ENG design and implementation available on their platforms.

This feature is supported on the following IoT platforms:

- IR1101
- IR1800
- IR8140
- ESR6300

When secure data wipe is executed, the following are wiped out:

- IR1101, IR1800, IR8140: NVRAM, rommon variables, and bootflash.
- ESR6300: NVRAM, rommon variables, and bootflash.

Procedure

Step 1 Use the **factory-reset all secure** command to execute the secure data wipe. When prompted to confirm, enter **Y**.

Example:

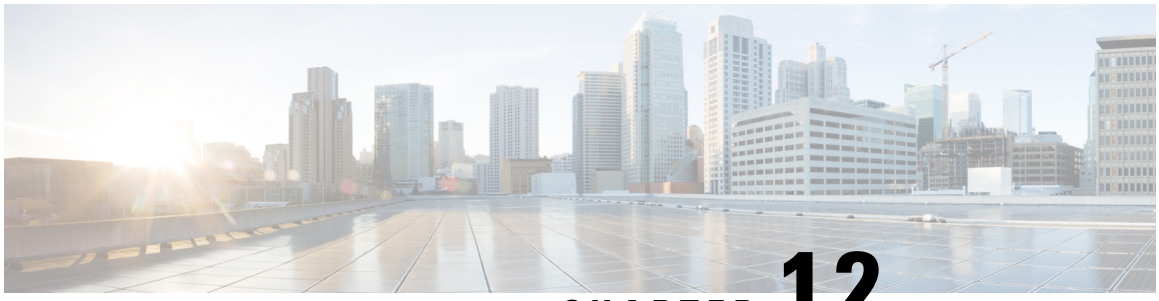
```
Router# factory-reset all secure
The factory reset operation is irreversible for securely reset all. Are you sure? [confirm]Y
This operation may take hours. Please do not power cycle.
```

Step 2 Use the **show platform software factory-reset secure log** command after the command is executed and IOS XE boots up, to check the log.

Example:

```
Router# show platform software factory-reset secure log
Factory reset log:
#CISCO DATA SANITIZATION REPORT:# IR1800
Purge ACT2 chip at 12-08-2022, 15:17:28
ACT2 chip Purge done at 12-08-2022, 15:17:29
mtd and backup flash wipe start at 12-08-2022, 15:17:29
mtd and backup flash wipe done at 12-08-2022, 15:17:29.
```

The router is in the rommon prompt with default factory settings (baud rate 9600) after the command is executed. The bootflash is not formatted until booting with an IOS image through usbflash or TFTP download, if the platform is supported.



CHAPTER 12

Software Maintenance Upgrade (SMU)

This section contains the following:

- Software Maintenance Upgrade (SMU), on page 131
- SMU Work-flow and Basic Requirements, on page 132
- SMU Example, on page 132
- Installing a Patch Image, on page 132
- Uninstalling the Patch Image, on page 135

Software Maintenance Upgrade (SMU)

The Software Maintenance Upgrade (SMU) is a package that can be installed on a system to provide a patch fix or security resolution to a released image for a specific defect in order to respond to immediate issues. It does not contain new features.

Note SMU installation was supported in both bundle boot and install mode. From Cisco IOS XE Release 17.9.x, SMU installation will be stopped if the router is booted up in bundle mode. If the router is booted up in install mode, SMU installation will keep working as it is in previous releases.

Some of the caveats of the SMU are:

- Provided on a per release, per component basis and is specific to the platform. SMU versions are synchronized to the package major, minor, and maintenance versions they upgrade.
- SMUs are not an alternative to maintenance releases. All defects fixed by SMUs are then automatically integrated into the upcoming maintenance releases.
- The Cisco IOS XE platform internally validates the SMU compatibility and does not allow you to install non-compatible SMUs. This is based on rules/limitations for a SMU change-set.
- An SMU provides a significant benefit over classic IOS software as it allows you to address the network issue quickly while reducing the time and scope of the testing required.
- SMU is a method to fix bugs in an existing release, and allows the application of a PSIRT fix in an existing release.
- SMU is NOT an upgrade path from release X to maintenance release X.1

- SMU is NOT an upgrade path from release X to release Y

The device only supports “Hot Patching”. This means:

- The running image is modified in-place or in-service
- This avoids downtime and interruption of service
- The updated code to fix the defect is written in a different location, and where the patch redirects the program run

SMU Work-flow and Basic Requirements

The work-flow for the patch requires that you complete the following sequence of operation in exec mode:

1. Addition of the SMU to the file system
2. Activation of the SMU onto the system
3. Committing the SMU change
4. Removal and Uninstallation of the SMU

The basic requirements for SMU are:

- The image where the defect was discovered
- The patch file that contains the fix for the defect must be formatted as ir1800-image_name.release_version.CSCxxyyyyy.SPA.smu.bin

SMU Example

This section shows an example of a patch created as a test. Your patch will have a name associated with a CDET to be installed as a fix.

Installing a Patch Image

Perform the following steps to install the patch image:

Procedure

Step 1 Show a standard command.

```
Router#show power
Main PSU :
    Total Power Consumed: 11.37 Watts
    Configured Mode : N/A
    Current runtime state same : N/A
    PowerSupplySource : External PS
POE Module :
```

```
    Configured Mode : N/A
    Current runtime state same : N/A
    Total power available : 30 Watts
Router#
```

Step 2 Add the image.

```
Router# install add file
bootflash:ir1800-universalk9.2020-08-06_10.38.0.CSCxx12345.SSA.smu.bin
install_add: START Thu Aug  6 11:52:52 PDT 2020
cat: /tmp/patch/patch.sta: No such file or directory
install_add: Adding SMU
install_add: Checking whether new add is allowed ....

--- Starting SMU Add operation ---
Performing SMU_ADD on Active/Standby
  [1] SMU_ADD package(s) on R0
  [1] Finished SMU_ADD on R0
Checking status of SMU_ADD on [R0]
SMU_ADD: Passed on [R0]
Finished SMU Add operation

SUCCESS: install_add  Thu Aug  6 11:53:31 PDT 2020

Router#
```

Step 3 Activate the patch image.

```
Router# install activate file
bootflash:ir1800-universalk9.2020-08-06_10.38.0.CSCxx12345.SSA.smu.bin
install_activate: START Thu Aug  6 11:53:59 PDT 2020

System configuration has been modified.
Press Yes(y) to save the configuration and proceed.
Press No(n) for proceeding without saving the configuration.
Press Quit(q) to exit, you may save configuration and re-enter the command. [y/n/q] y
Building configuration...
[OK]Modified configuration has been saved
install_activate: Activating SMU
Executing pre scripts....
Executing pre sripts done.

--- Starting SMU Activate operation ---
Performing SMU_ACTIVATE on Active/Standby
/usr/sbin/kgv_update: kgv_update [
/flash1/ir1800-universalk9.2020-08-06_10.38.0.CSCxx12345.SSA.smu.bin, NOT slot local is ics
 ] continuing ....
/usr/sbin/kgv_update: Signature validated for
/flash1/ir1800-universalk9.2020-08-06_10.38.0.CSCxx12345.SSA.smu.bin
/usr/sbin/kgv_update: TAM hash len:32
val:4407CBB447F0EEE3B12120D902F48FBA1C0D4900EED1FB614441198BE2302934
/usr/sbin/kgv_update: PCR8 before extend ctr:2
0817449B454BF036AF9D593D726D94D8942C50A9FFE93278FDA78EA62F2989F2
/usr/sbin/kgv_update: PCR8 after extend ctr:3
EF5F579FCDFA989D044296F0584B99F719F2B6215895524B5E8AD55DF5671560
/usr/sbin/kgv_update: PCR extend successful
/usr/sbin/kgv_update: Chasfs updated for
name:ir1800-universalk9.2020-08-06_10.38.0.CSCxx12345.SSA.smu.bin
hash:975352C1562A492D582D09D5BB91230863F6CC18E6F9C0EB512AF27CC0C77E2C05F29596AD34AD7808C9B39EC23D4412F0D3AFA707BC906FE03D554A845E42D4
/usr/sbin/kgv_update: Update successful for
ir1800-universalk9.2020-08-06_10.38.0.CSCxx12345.SSA.smu.bin
  [1] SMU_ACTIVATE package(s) on R0
```

```
  [1] Finished SMU_ACTIVATE on R0
Checking status of SMU_ACTIVATE on [R0]
SMU_ACTIVATE: Passed on [R0]
Finished SMU Activate operation
SUCCESS: install_activate /flash1/ir1800-universalk9.2020-08-06_10.38.0.CSCxx12345.SSA.smu.bin
 Thu Aug  6 11:55:14 PDT 2020
Router#
```

Step 4 Commit the installation.

```
Router# install commit
install_commit: START Thu Aug  6 11:55:29 PDT 2020
install_commit: Committing SMU
Executing pre scripts....
Executing pre sripts done.
--- Starting SMU Commit operation ---
Performing SMU_COMMIT on Active/Standby
  [1] SMU_COMMIT package(s) on R0
  [1] Finished SMU_COMMIT on R0
Checking status of SMU_COMMIT on [R0]
SMU_COMMIT: Passed on [R0]
Finished SMU Commit operation

SUCCESS: install_commit /flash1/ir1800-universalk9.2020-08-06_10.38.0.CSCxx12345.SSA.smu.bin
 Thu Aug  6 11:56:08 PDT 2020
Router#
```

Step 5 Show the status summary of the installation procedure.

```
Router# 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    17.04.01.0.118999
SMU   C    flash:ir1800-universalk9.2020-08-06_10.38.0.CSCxx12345.SSA.smu.bin

--------------------------------------------------------------------------------
Auto abort timer: inactive
--------------------------------------------------------------------------------
Router#
```

Step 6 Verify the result of the patch by showing the same command.

```
Router#show power
Main PSU :
    Total Power Consumed: 11.04 Watts
Device HOT SMU works!

    Configured Mode : N/A
    Current runtime state same : N/A
    PowerSupplySource : External PS
POE Module :
    Configured Mode : N/A
    Current runtime state same : N/A
    Total power available : 0 Watts
Router#
```

Uninstalling the Patch Image

There are two methods to remove or uninstall the patch image.

- Restoring the image to its original version by using the following command:
  - **install rollback to base**
- Specific removal of a patch by using the following commands in sequence:
  - **install deactivate file flash:**<*file*>
  - **install commit**
  - **install remove file flash:**<*file*>

Uninstalling the Patch Image Using Rollback

This section shows an example of using the rollback method.

Show what patches are installed:

```
Router# 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    17.04.01.0.118999
SMU   C    flash:ir1800-universalk9.2020-08-06_10.38.0.CSCxx12345.SSA.smu.bin

--------------------------------------------------------------------------------
Auto abort timer: inactive
--------------------------------------------------------------------------------
Router#
```

The following commands are available:

```
Router# install ?
  abort             Abort the current install operation
  activate          Activate an installed package
  add               Install a package file to the system
  auto-abort-timer  Install auto-abort-timer
  commit            Commit the changes to the loadpath
  deactivate        Deactivate an install package
  label             Add a label name to any installation point
  prepare           Prepare package for operation
  remove            Remove installed packages
  rollback          Rollback to a previous installation point
Router# install rollback to ?
  base       Rollback to the base image
  committed  Rollback to the last committed installation point
  id         Rollback to a specific install point id
  label      Rollback to a specific install point label
```

The **install rollback to base** command removes the entire patch and returns to the base image version with the found defect.

```
Router# install rollback to base
install_rollback: START Thu Aug  6 12:04:04 PDT 2020
install_rollback: Rolling back SMU
Executing pre scripts....
Executing pre sripts done.

--- Starting SMU Rollback operation ---
Performing SMU_ROLLBACK on Active/Standby
  [1] SMU_ROLLBACK package(s) on R0
  [1] Finished SMU_ROLLBACK on R0
Checking status of SMU_ROLLBACK on [R0]
SMU_ROLLBACK: Passed on [R0]
Finished SMU Rollback operation

CSCxx12345:SUCCESS
SUCCESS: install_rollback
/flash1/ir1800-universalk9.2020-08-06_10.38_shchang2.0.CSCxx12345.SSA.smu.bin Thu Aug  6
12:04:57 PDT 2020
Router#
```

Show what patches are installed:

```
Router# 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    17.04.01.0.118999

--------------------------------------------------------------------------------
Auto abort timer: inactive
--------------------------------------------------------------------------------
Router#
```

Note In the above command output, the patch has been removed and the device returns to the base image version prior to the upgrade.

Verify the result of the patch by showing the same command.

```
Router#show power
Main PSU :
    Total Power Consumed: 11.98 Watts
    Configured Mode : N/A
    Current runtime state same : N/A
    PowerSupplySource : External PS
POE Module :
    Configured Mode : N/A
    Current runtime state same : N/A
    Total power available : 30 Watts
Router#
```

Uninstalling the Patch Image Using Deactivate, Commit, and Remove

In the following sequence, there are two patches installed on the device. CSCvq11111 and CSCvt22222 Only CSCvt22222 will be removed.

Show what patches are installed.

```
Router# 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
--------------------------------------------------------------------------------
SMU   C    /flash1/ir1800-universalk9.<release>.CSCvq11111.SPA.smu.bin
SMU   C    /flash1/ir1800-universalk9.<release>.CSCvt22222.SPA.smu.bin
IMG   C    17.04.1
```

Procedure

Step 1 Deactivate the patch.

```
Router# install deactivate file flash:ir1800-universalk9.<release>.CSCvt22222.SPA.smu.bin
install_deactivate: START Fri Apr 24 22:54:10 UTC 2020
install_deactivate: Deactivating SMU
Executing pre scripts....
Executing pre sripts done.

--- Starting SMU Deactivate operation ---
Performing SMU_DEACTIVATE on Active/Standby
  [R0] SMU_DEACTIVATE package(s) on R0
  [R0] Finished SMU_DEACTIVATE on R0
Checking status of SMU_DEACTIVATE on [R0]
SMU_DEACTIVATE: Passed on [R0]
Finished SMU Deactivate operation

SUCCESS: install_deactivate /flash1/ir1800-universalk9.<release>.CSCvt22222.SPA.smu.bin Fri
 Apr 24 22:54:49 UTC 2020
```

Show what patches are installed:

```
Router# 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
--------------------------------------------------------------------------------
SMU   C    /flash1/ir1800-universalk9.<release>.CSCvt11111.SPA.smu.bin
SMU   D    /flash1/ir1800-universalk9.<release>.CSCvt22222.SPA.smu.bin
IMG   C    17.01.1
```

Step 2 Commit the action.

```
Router# install commit
install_commit: START Fri Apr 24 22:56:11 UTC 2020
install_commit: Committing SMU
```

```
*Apr 24 22:56:15.169: %INSTALL-5-INSTALL_START_INFO: R0/0: install_engine: Started install
 commitExecuting pre scripts....
Executing pre sripts done.
--- Starting SMU Commit operation ---
Performing SMU_COMMIT on Active/Standby
  [R0] SMU_COMMIT package(s) on R0
  [R0] Finished SMU_COMMIT on R0
Checking status of SMU_COMMIT on [R0]
SMU_COMMIT: Passed on [R0]
Finished SMU Commit operation

SUCCESS: install_commit /flash1/ir1800-universalk9.<release>.CSCvt22222.SPA.smu.bin Fri Apr
 24 22:56:32 UTC 2020

*Apr 24 22:56:33.342: %INSTALL-5-INSTALL_COMPLETED_INFO: R0/0: install_engine: Completed
install commit SMU
```

Show what patches are installed:

```
Router# 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
--------------------------------------------------------------------------------
SMU   C    /flash1/ir1800-universalk9.<release>.CSCvt11111.SPA.smu.bin
SMU   I    /flash1/ir1800-universalk9.<release>.CSCvt22222.SPA.smu.bin
IMG   C    <release>
```

Step 3 Remove the patch.

```
Router# install remove file flash:ir1800-universalk9.<release>.CSCvt22222.SPA.smu.bin
install_remove: START Fri Apr 24 22:57:17 UTC 2020

*Apr 24 22:57:20.775: %INSTALL-5-INSTALL_START_INFO: R0/0: install_engine: Started install
 remove flash:ir1800-universalk9.<release>.CSCvt22222.SPA.smu.bininstall_remove: Removing
SMU
Executing pre scripts....
Executing pre scripts done.

--- Starting SMU Remove operation ---
Performing SMU_REMOVE on Active/Standby
  [R0] SMU_REMOVE package(s) on R0
  [R0] Finished SMU_REMOVE on R0
Checking status of SMU_REMOVE on [R0]
SMU_REMOVE: Passed on [R0]
Finished SMU Remove operation

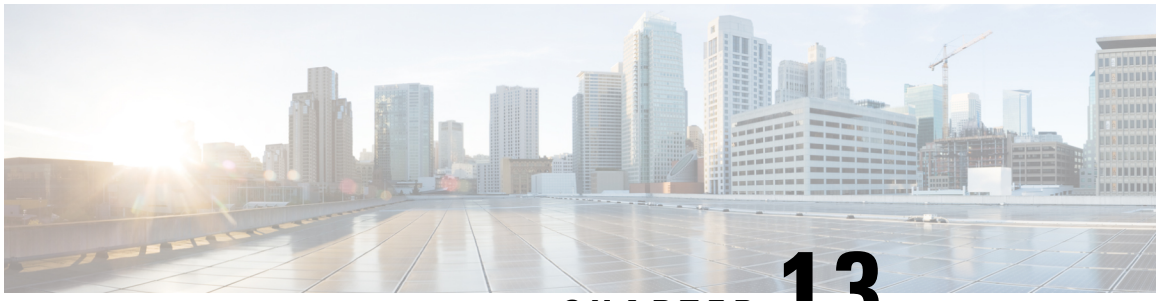
SUCCESS: install_remove /flash1/ir1800-universalk9.<release>.CSCvt22222.SPA.smu.bin Fri Apr
 24 22:57:34 UTC 2020

*Apr 24 22:57:34.902: %INSTALL-5-INSTALL_COMPLETED_INFO: R0/0: install_engine: Completed
install remove flash:ir1800-universalk9.<release>.CSCvt22222.SPA.smu.bin
```

Show what patches are installed:

```
Router# 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
--------------------------------------------------------------------------------
SMU   C    /flash1/ir1800-universalk9.<release>.CSCvt11111.SPA.smu.bin
IMG   C    <release>
```



CHAPTER 13

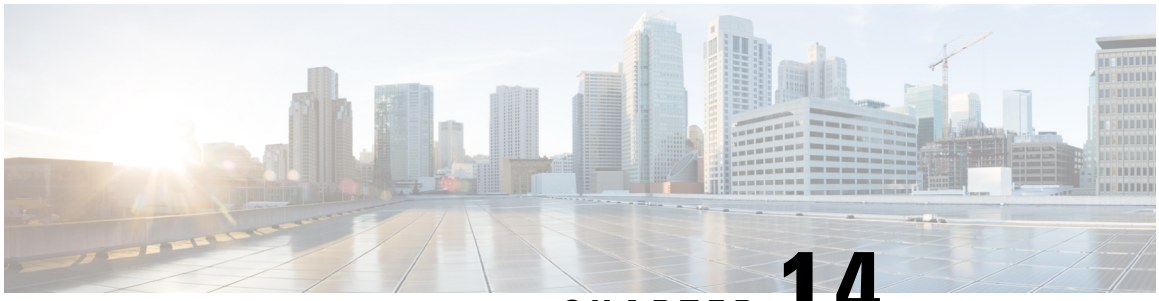
Smart Licensing Using Policy

This reference provides information regarding the implementation and management of Smart Licensing Using Policy.

- Smart licensing using policy on Cisco routers, on page 141

Smart licensing using policy on Cisco routers

The Smart Licensing Using Policy chapter has been replaced by a standalone guide called Smart Licensing Using Policy on the Cisco Catalyst IR1101, IR1800, IR8140, IR8340 and ESR6300 Routers.



CHAPTER 14

Configuring Ethernet Switch Ports

This chapter contains the following sections:

- Configuring VLANs, on page 143
- VLAN Trunking Protocol (VTP), on page 144
- Configuring 802.1x Authentication, on page 144
- Configuring Spanning Tree Protocol, on page 145
- Configuring MAC Address Table Manipulation, on page 147
- Configuring Switch Port Analyzer, on page 148
- Configuring IGMP Snooping, on page 149
- Configuring EVPN VXLAN VLAN-Aware Service, on page 149
- Multi-Homing Configuration in BGP EVPN VXLAN Fabric, on page 169

Configuring VLANs

A VLAN is a switched network that is logically segmented by function, project team, or application, without regard to the physical locations of the users. VLANs have the same attributes as physical LANs, but you can group end stations even if they are not physically located on the same LAN segment. Any switch port can belong to a VLAN, and unicast, broadcast, and multicast packets are forwarded and flooded only to end stations in the VLAN. Each VLAN is considered a logical network, and packets destined for stations that do not belong to the VLAN must be forwarded through a router.

Note

- IR1800 routers have reserved a set of VLANs (2350 to 2449) for additional usage. It is not allowed to add the reserved VLANs. You must ensure that these VLANs are not used in the network.
- Jumbo frames on L2 interfaces are not supported.

The following is an example of a vlan configuration:

```
IR1800#show vlan
VLAN Name                             Status    Ports
---- -------------------------------- --------- -------------------------------
1    default                          active    Ge0/1/0, Ge0/1/1, Ge0/1/2, Ge0/1/3
1002 fddi-default                     act/unsup
1003 token-ring-default               act/unsup
1004 fddinet-default                  act/unsup
1005 trnet-default                    act/unsup
```

```
VLAN Type  SAID       MTU   Parent RingNo BridgeNo Stp  BrdgMode Trans1 Trans2
---- ----- ---------- ----- ------ ------ -------- ---- -------- ------ ------
1    enet  100001     1500  -      -      -        -    -        0      0
1002 fddi  101002     1500  -      -      -        -    -        0      0
1003 tr    101003     1500  -      -      -        -    -        0      0
1004 fdnet 101004     1500  -      -      -        ieee -        0      0
1005 trnet 101005     1500  -      -      -        ibm  -        0      0

Primary Secondary Type              Ports
------- --------- ----------------- ------------------------------------------

IR1800#
```

You can assign a given port to a vlan by following these steps:

```
interface GigabitEthernet0/1/0
switchport access vlan 4

interface vlan 4
ip v4 address ...
ipv6 address autoconf
```

VLAN Trunking Protocol (VTP)

VTP is a Layer 2 messaging protocol that maintains VLAN configuration consistency by managing the addition, deletion, and renaming of VLANs on a network-wide basis. VTP minimizes misconfigurations and configuration inconsistencies that can cause several problems, such as duplicate VLAN names, incorrect VLAN-type specifications, and security violations.

Before you create VLANs, you must decide whether to use VTP in your network. Using VTP, you can make configuration changes centrally on one or more switches and have those changes automatically communicated to all the other switches in the network. Without VTP, you cannot send information about VLANs to other switches.VTP is designed to work in an environment where updates are made on a single switch and are sent through VTP to other switches in the domain. It does not work well in a situation where multiple updates to the VLAN database occur simultaneously on switches in the same domain, which would result in an inconsistency in the VLAN database.

Further information about configuring VTP can be found here:http://www.cisco.com/c/en/us/td/docs/routers/access/interfaces/software/feature/guide/geshwic_cfg.html#wp1046901

Configuring 802.1x Authentication

IEEE 802.1x port-based authentication defines a client-server-based access control and authentication protocol to prevent unauthorized clients from connecting to a LAN through publicly accessible ports.The authentication server authenticates each client connected to a switch port before allowing access to any switch or LAN services. Until the client is authenticated, IEEE 802.1x access control allows only Extensible Authentication Protocol over LAN (EAPOL), Cisco Discovery Protocol (CDP), and Spanning Tree Protocol (STP) traffic through the port to which the client is connected. After authentication, normal traffic passes through the port.

With IEEE 802.1x authentication, the devices in the network have specific roles:

- Supplicant—Device (workstation) that requests access to the LAN and switch services and responds to requests from the router. The workstation must be running IEEE 802.1x-compliant client software such

as that offered in the Microsoft Windows XP operating system. (The supplicant is sometimes called the client.)

- Authentication server—Device that performs the actual authentication of the supplicant. The authentication server validates the identity of the supplicant and notifies the router whether or not the supplicant is authorized to access the LAN and switch services. The Network Access Device transparently passes the authentication messages between the supplicant and the authentication server, and the authentication process is carried out between the supplicant and the authentication server. The particular EAP method used will be decided between the supplicant and the authentication server (RADIUS server). The RADIUS security system with EAP extensions is available in Cisco Secure Access Control Server Version 3.0 or later. RADIUS operates in a client and server model in which secure authentication information is exchanged between the RADIUS server and one or more RADIUS clients.
- Authenticator—Router that controls the physical access to the network based on the authentication status of the supplicant. The router acts as an intermediary between the supplicant and the authentication server, requesting identity information from the supplicant, verifying that information with the authentication server, and relaying a response to the supplicant. The router includes the RADIUS client, which is responsible for encapsulating and decapsulating the EAP frames and interacting with the authentication server.

For detailed information on how to configure 802.1x port-based authentication, see the following link:

http://www.cisco.com/c/en/us/td/docs/ios-xml/ios/sec_usr_8021x/configuration/15-mt/sec-user-8021x-15-mt-book/config-ieee-802x-pba.html

Example: Enabling IEEE 802.1x and AAA on a Switch Port

This example shows how to configure an IR1800 router as 802.1x authenticator:

```
Router> enable
Router# configure terminal
Router(config)# dot1x system-auth-control
Router(config)# aaa new-model
Router(config)# aaa authentication dot1x default group radius
Router(config)# interface GigabitEthernet 0/1/0
Router(config-if)# switchport mode access
Router(config-if)# access-session port-control auto
Router(config-if)# dot1x pae authenticator
Router(config-if)# access-session closed
Router(config-if)# access-session host-mode single-host
Router(config-if)# end
```

Configuring Spanning Tree Protocol

Spanning Tree Protocol (STP) is a Layer 2 link management protocol that provides path redundancy while preventing loops in the network. For a Layer 2 Ethernet network to function properly, only one active path can exist between any two stations. Multiple active paths among end stations cause loops in the network. If a loop exists in the network, end stations might receive duplicate messages. Switches might also learn end-station MAC addresses on multiple Layer 2 interfaces. These conditions result in an unstable network. Spanning-tree operation is transparent to end stations, which cannot detect whether they are connected to a single LAN segment or a switched LAN of multiple segments.

The STP uses a spanning-tree algorithm to select one switch of a redundantly connected network as the root of the spanning tree. The algorithm calculates the best loop-free path through a switched Layer 2 network by assigning a role to each port based on the role of the port in the active topology:

- Root—A forwarding port elected for the spanning-tree topology
- Designated—A forwarding port elected for every switched LAN segment
- Alternate—A blocked port providing an alternate path to the root bridge in the spanning tree
- Backup—A blocked port in a loopback configuration

The switch that has all of its ports as the designated role or as the backup role is the root switch. The switch that has at least one of its ports in the designated role is called the designated switch. Spanning tree forces redundant data paths into a standby (blocked) state. If a network segment in the spanning tree fails and a redundant path exists, the spanning-tree algorithm recalculates the spanning-tree topology and activates the standby path. Switches send and receive spanning-tree frames, called bridge protocol data units (BPDUs), at regular intervals. The switches do not forward these frames but use them to construct a loop-free path. BPDUs contain information about the sending switch and its ports, including switch and MAC addresses, switch priority, port priority, and path cost. Spanning tree uses this information to elect the root switch and root port for the switched network and the root port and designated port for each switched segment.

When two ports on a switch are part of a loop, the spanning-tree port priority and path cost settings control which port is put in the forwarding state and which is put in the blocking state. The spanning-tree port priority value represents the location of a port in the network topology and how well it is located to pass traffic. The path cost value represents the media speed.

For detailed configuration information on STP see the following link:

http://www.cisco.com/c/en/us/td/docs/routers/access/interfaces/NIM/software/configuration/guide/4_8PortGENIM.html#pgfId-1079138



Important If the router is factory-defaulted, write erased, or config-reset, the vlan database gets deleted. Even though the configuration takes effect, interfaces need to be removed and re-applied.

Example: Spanning Tree Protocol Configuration

The following example shows configuring spanning-tree port priority of a Gigabit Ethernet interface. If a loop occurs, spanning tree uses the port priority when selecting an interface to put in the forwarding state.

```
Router# configure terminal
Router(config)# interface GigabitEthernet 0/1/0
Router(config-if)# spanning-tree vlan 1 port-priority 64
Router(config-if)# end
```

The following example shows how to change the spanning-tree port cost of a Gigabit Ethernet interface. If a loop occurs, spanning tree uses cost when selecting an interface to put in the forwarding state.

```
Router#configure terminal
Router(config)# interface GigabitEthernet 0/1/0
Router(config-if)# spanning-tree cost 18
Router(config-if)# end
```

The following example shows configuring the bridge priority of VLAN 10 to 33792:

```
Router# configure terminal
Router(config)# spanning-tree vlan 10 priority 33792
Router(config)# end
```

The following example shows configuring the hello time for VLAN 10 being configured to 7 seconds. The hello time is the interval between the generation of configuration messages by the root switch.

```
Router# configure terminal
Router(config)# spanning-tree vlan 10 hello-time 7
Router(config)# end
```

The following example shows configuring forward delay time. The forward delay is the number of seconds an interface waits before changing from its spanning-tree learning and listening states to the forwarding state.

```
Router# configure terminal
Router(config)# spanning-tree vlan 10 forward-time 21
Router(config)# end
```

The following example shows configuring maximum age interval for the spanning tree. The maximum-aging time is the number of seconds a switch waits without receiving spanning-tree configuration messages before attempting a reconfiguration.

```
Router# configure terminal
Router(config)# spanning-tree vlan 20 max-age 36
Router(config)# end
```

The following example shows the switch being configured as the root bridge for VLAN 10, with a network diameter of 4.

```
Router# configure terminal
Router(config)# spanning-tree vlan 10 root primary diameter 4
Router(config)# exit
```

Configuring MAC Address Table Manipulation

The MAC address table contains address information that the switch uses to forward traffic between ports. All MAC addresses in the address table are associated with one or more ports. The address table includes these types of addresses:

- Dynamic address: a source MAC address that the switch learns and then drops when it is not in use. You can use the aging time setting to define how long the switch retains unseen addresses in the table.
- Static address: a manually entered unicast address that does not age and that is not lost when the switch resets.

The address table lists the destination MAC address, the associated VLAN ID, and port associated with the address and the type (static or dynamic).

Port security is supported, as is sticky MAC addresses.

See the “Example: MAC Address Table Manipulation” for sample configurations for enabling secure MAC address, creating a statc entry, set the maximum number of secure MAC addresses and set the aging time.

For detailed configuration information on MAC address table manipulation see the following link:

http://www.cisco.com/c/en/us/td/docs/routers/access/interfaces/software/feature/guide/geshwic_cfg.html#wp1048223

Example: MAC Address Table Manipulation

The following example shows creating a static entry in the MAC address table.

```
Router# configure terminal
Router(config)# mac address-table static 0002.0003.0004 interface GigabitEthernet 0/1/0
vlan 3
Router(config)# end
```

The following example shows setting the aging timer.

```
Router# configure terminal
Router(config)# mac address-table aging-time 300
Router(config)# end
```

Configuring Switch Port Analyzer

The Cisco IR1800 supports local SPAN only, and up to one SPAN session. You can analyze network traffic passing through ports by using SPAN to send a copy of the traffic to another port on the switch or on another switch that has been connected to a network analyzer or other monitoring or security device. SPAN copies (or mirrors) traffic received or sent (or both) on source ports to a destination port for analysis. SPAN does not affect the switching of network traffic on the source ports. You must dedicate the destination port for SPAN use. Except for traffic that is required for the SPAN or RSPAN session, destination ports do not receive or forward traffic.

Only traffic that enters or leaves source ports or traffic that enters or leaves source can be monitored by using SPAN; traffic routed to a source cannot be monitored. For example, if incoming traffic is being monitored, traffic that gets routed from another source cannot be monitored; however, traffic that is received on the source and routed to another can be monitored.

For detailed information on how to configure a switched port analyzer (SPAN) session, see the following web link:

http://www.cisco.com/c/en/us/td/docs/switches/lan/catalyst3750/software/release/15-0_2_se/configuration/guide/scg3750/swspan.html

Example: SPAN Configuration

The following example shows how to configure a SPAN session to monitor bidirectional traffic from a Gigabit Ethernet source interface:

```
Router# configure terminal
Router(config)# monitor session 1 source GigabitEthernet 0/1/0
Router(config)# end
```

The following example shows how to configure a gigabit ethernet interface as the destination for a SPAN session:

```
Router# configure terminal
Router(config)# monitor session 1 destination GigabitEthernet 0/1/0
Router(config)# end
```

The following example shows how to remove gigabit ethernet as a SPAN source for SPAN session 1:

```
Router# configure terminal
Router(config)# no monitor session 1 source GigabitEthernet 0/1/0
Router(config)# end
```

Show Monitor Example

```
Router(config)#monitor session 1 source interface gi0/1/0
Router(config)#monitor session 1 destination interface gi0/1/1
Router#sh monitor session 1
Session 1
---------
Type : Local Session
Source Ports :
Both : Gi0/1/0
Destination Ports : Gi0/1/1
```

Example of ERSPAN

```
Router#show monitor session 1
Session 1
---------
Type                      : ERSPAN Source Session
Status                    : Admin Disabled
Source Ports              :
    RX Only               : Gi0/0/0
Destination IP Address    : 172.5.5.200
MTU                       : 1464
Destination ERSPAN ID     : 100
Origin IP Address         : 172.5.6.2
IPv6 DSCP                 : 0
IPV6 TTL                  : 0
```

Configuring IGMP Snooping

IGMP snooping constrains the flooding of multicast traffic by dynamically configuring Layer 2 interfaces so that multicast traffic is forwarded to only those interfaces associated with IP multicast devices. As the name implies, IGMP snooping requires the LAN switch to snoop on the IGMP transmissions between the host and the router and to keep track of multicast groups and member ports. When the switch receives an IGMP report from a host for a particular multicast group, the switch adds the host port number to the forwarding table entry; when it receives an IGMP Leave Group message from a host, it removes the host port from the table entry. It also periodically deletes entries if it does not receive IGMP membership reports from the multicast clients.

The multicast router sends out periodic general queries to all VLANs. All hosts interested in this multicast traffic send join requests and are added to the forwarding table entry.

Use the **ip igmp snooping enable** command to configure IGMP Snooping on the IR1800.

By default, IGMP snooping is globally enabled in the IR1800.

MLD snooping is also supported on the IR1800, and further information can be found in this documentation set: https://www.cisco.com/c/en/us/td/docs/switches/lan/catalyst3850/software/release/16-1/configuration_guide/b_161_consolidated_3850_cg/b_161_consolidated_3850_cg_chapter_01100.html

Configuring EVPN VXLAN VLAN-Aware Service

A VLAN-Aware EVPN instance (EVI) allows multiple subnets (L2VNI) to be mapped to a single EVI. When this feature is configured, the MAC-VRF is identified by the combination of the route-target and the ethernet-tag. The EVPN routes that require the identification of a specific bridge-table within a MAC-VRF will advertise

these routes with the Ethernet Tag field set to a value that allows such identification. This feature enhances the router's inter-operability.

There are two methods to configure VLAN-Aware:

1. Profile based configuration
2. Manual configuration

Guidelines and Limitations

- You may configure either the Profile based configuration, or Manual Configuration method on an EVI. Both methods cannot be used to configure VLAN-aware on the same EVI on the device.
- This feature is supported only on Layer 2. IRB and L3 are not supported.

Profile based Configuration of EVPN VXLAN VLAN-Aware Service

Configure EVPN VXLAN VLAN-Aware instance

Follow this procedure to configure a VLAN-aware EVPN instance (EVI):

Procedure

	Command or Action	Purpose
Step 1	configure terminal **Example:** `Router#configure terminal`	Enters Global Configuration Mode
Step 2	l2vpn evpn profile <profile-name> <service-type> **Example:** `Router(config)#l2vpn evpn profile evpn_va vlan-aware`	Configures an EVPN Vxlan Vlan-Aware profile.
Step 3	evi-id <id> **Example:** `Router(config-evpn-prof)#evi-id 1`	Configures the EVPN Instance (EVI) id.
Step 4	ethernet-tag [auto-vni|auto-vlan] **Example:** `Router(config)#ethernet-tag auto-vlan`	Enables the autogeneration of the Ethernet-Tag based on the L2VNI or on the VLAN value. The default is *auto-vni*.
Step 5	(Optional) l2vni-base <l2vni-base> **Example:** `Router(config-evpn-prof)#l2vni-base 50000`	(Optional) Configures the L2 Virtual Network Identifier (VNI) base id.

	Command or Action	Purpose
Step 6	(Optional) replication-type {*ingress*|*static* {[*ipv4_mcast_addr*|*ipv4_mcast_prefix*] **Example:** `Router(config-evpn-prof)#replication-type ingress`	(Optional) Configures the replication-type.
Step 7	(Optional) encapsulation vxlan **Example:** `Router(config-evpn-prof)#encapsulation vxlan`	(Optional) Configures the encapsulation type.
Step 8	(Optional) default-gateway advertise {enable|disable} **Example:** `Router(config)#default-gateway advertise enable`	(Optional) Enable or Disable default gateway advertising.
Step 9	(Optional) multicast advertise enable **Example:** `Router(config)#multicast advertise enable`	(Optional) Enable or Disable multicast advertising.
Step 10	(Optional) ip local-learning {enable|disable} **Example:** `Router(config)#ip local-learning enable`	(Optional) Enable or Disable IP based local learning.
Step 11	(Optional) flooding-suppression address-resolution {enable|disable} **Example:** `Router(config)#flooding-suppression address-resolution enable`	(Optional) Enable or Disable flooding suppression based address resolution.
Step 12	(Optional) re-originate route-type5 **Example:** `Router(config)#re-originate route-type5`	(Optional) Enable re-origination of type 5 routes.

Apply the configuration on a Bridge-Domain

Follow the procedure to apply the configuration on a bridge-domain on your device:

Procedure

	Command or Action	Purpose
Step 1	bridge-domain <id> **Example:**	Applies the configuration to the specified bridge domain.

	Command or Action	Purpose
	`Router(config)#bridge-domain 12`	
Step 2	member Vlan12 service-instance 12 **Example:** `Router(config-bdomain)#member Vlan12 service-instance 12`	Specifies the service instance for the VLAN member.
Step 3	member evpn-instance profile <va-profile-name> **Example:** `Router(config-bdomain)#member evpn-instance profile evpn_va`	Add an EVI member to the bridge-domain.

Manual configuration of EVPN VXLAN VLAN-Aware Service

To manually configure EVPN VXLAN VLAN-Aware service on your device, complete these tasks:

- Static Configuration of VXLAN VLAN-Aware EVPN Instance, on page 152
- Applying Configuration on the Bridge Domain, on page 153
- Configure Ingress Replication for EVPN VXLAN VLAN-Aware, on page 153
- Configure Static Replication for VLAN-Aware, on page 155

Static Configuration of VXLAN VLAN-Aware EVPN Instance

Follow the procedure for static configuration of the EVPN instance:

Procedure

	Command or Action	Purpose
Step 1	configure terminal **Example:** `Router#configure terminal`	Enters Global Configuration Mode
Step 2	l2vpn evpn instance 1 vlan-aware **Example:** `Router(config)#l2vpn evpn instance 1 vlan-aware`	Configure a VLAN-Aware EVPN instance.
Step 3	encapsulation vxlan **Example:** `Router(config-evpn-evi)#encapsulation vxlan`	Configures the encapsulation type to VXLAN.

Example

```
l2vpn evpn instance 1 vlan-aware
 encapsulation vxlan
!
```

Applying Configuration on the Bridge Domain

Follow the procedure to apply the configuration on the bridge domain:

Procedure

	Command or Action	Purpose
Step 1	configure terminal **Example:** `Router#configure terminal`	Enters Global Configuration Mode
Step 2	bridge-domain 12 **Example:** `Router(config)bridge-domain 12`	Specifies the bridge domain for the configuration to be applied to.
Step 3	member Vlan12 service-instance 12 **Example:** `Router(config-bdomain)#member Vlan12 service-instance 12`	Specifies the service instance for the VLAN member.
Step 4	member evpn-instance 1 vni <vni> ethernet-tag <etag> **Example:** `Router(config-bdomain)#member evpn-instance 1 vni 30012 ethernet-tag 20012`	Configure a VLAN-aware EVI member under the bridge-domain

Example

```
bridge-domain 12
 member Vlan12 service-instance 12
 member evpn-instance 1 vni 30012 ethernet-tag 20012
```

Configure Ingress Replication for EVPN VXLAN VLAN-Aware

Follow this procedure to configure Ingress Replication for EVPN VXLAN VLAN-Aware:

Procedure

	Command or Action	Purpose
Step 1	configure terminal **Example:** `Router#configure terminal`	Enters Global Configuration Mode
Step 2	l2vpn evpn instance 1 vlan-aware **Example:** `Router(config)#l2vpn evpn instance 1 vlan-aware`	Configure a VLAN-Aware EVPN instance
Step 3	replication-type ingress **Example:** `Router(config-evpn-evi)#replication-type ingress`	Configures the replication type to ingress replication.
Step 4	interface nve1 **Example:** `Router(config)#interface nve1`	Configures the NVE interface and enters the interface configuration mode.
Step 5	no ip address **Example:** `Router(config-if)#no ip address`	Disables IP processing on the interface.
Step 6	source-interface Loopback0 **Example:** `Router(config-if)#source-interface Loopback0`	Specifies the source loopback for the interface.
Step 7	host-reachability protocol bgp **Example:** `Router(config-if)#host-reachability protocol bgp`	Specifies the BGP protocol for host reachability.
Step 8	member vni 30000 ingress replication **Example:** `Router(config-if)#member vni 30000 ingress replication`	Configures the L2VNI member with ingress replication.

Example

```
l2vpn evpn instance 1 vlan-aware
 encapsulation vxlan
 replication-type ingress
!
interface nve1
```

```
 no ip address
 source-interface Loopback0
 host-reachability protocol bgp
 member vni 30000 ingress-replication
 member vni 30012 ingress-replication
!
```

Configure Static Replication for VLAN-Aware

Follow this procedure to configure Static Replication for VLAN-Aware:

Procedure

	Command or Action	Purpose
Step 1	configure terminal **Example:** `Router#configure terminal`	Enters Global Configuration Mode
Step 2	l2vpn evpn instance 1 vlan-aware **Example:** `Router(config)#l2vpn evpn instance 1 vlan-aware`	Enables VLAN-Aware on the specified EVPN instance.
Step 3	replication-type static **Example:** `Router(config)#replication-type static`	Configures the replication type to static replication.
Step 4	interface nve1 **Example:** `Router(config)#interface nve1`	Configures the interface and enters the interface configuraiton mode.
Step 5	no ip address **Example:** `Router(config-if)#no ip address`	Disables IP processing on the interface.
Step 6	source-interface Loopback0 **Example:** `Router(config-if)#source-interface Loopback0`	Specifies the source loopback for the interface.
Step 7	host-reachability protocol bgp **Example:** `Router(config-if)#host-reachability protocol bgp`	Specifies the BGP protocol for host reachability.
Step 8	member vni 30000 mcast-group 209.165.1.1 **Example:**	Configures the VNI member and Multicast group

	Command or Action	Purpose
	`Router(config-if)#member vni 30000 mcast-group 209.165.1.1`	
Step 9	member vni 30012 mcast-group 209.165.1.1 **Example:** `Router(config-if)#member vni 30012 mcast-group 209.165.1.1`	Configures the VNI member and Multicast group

Example

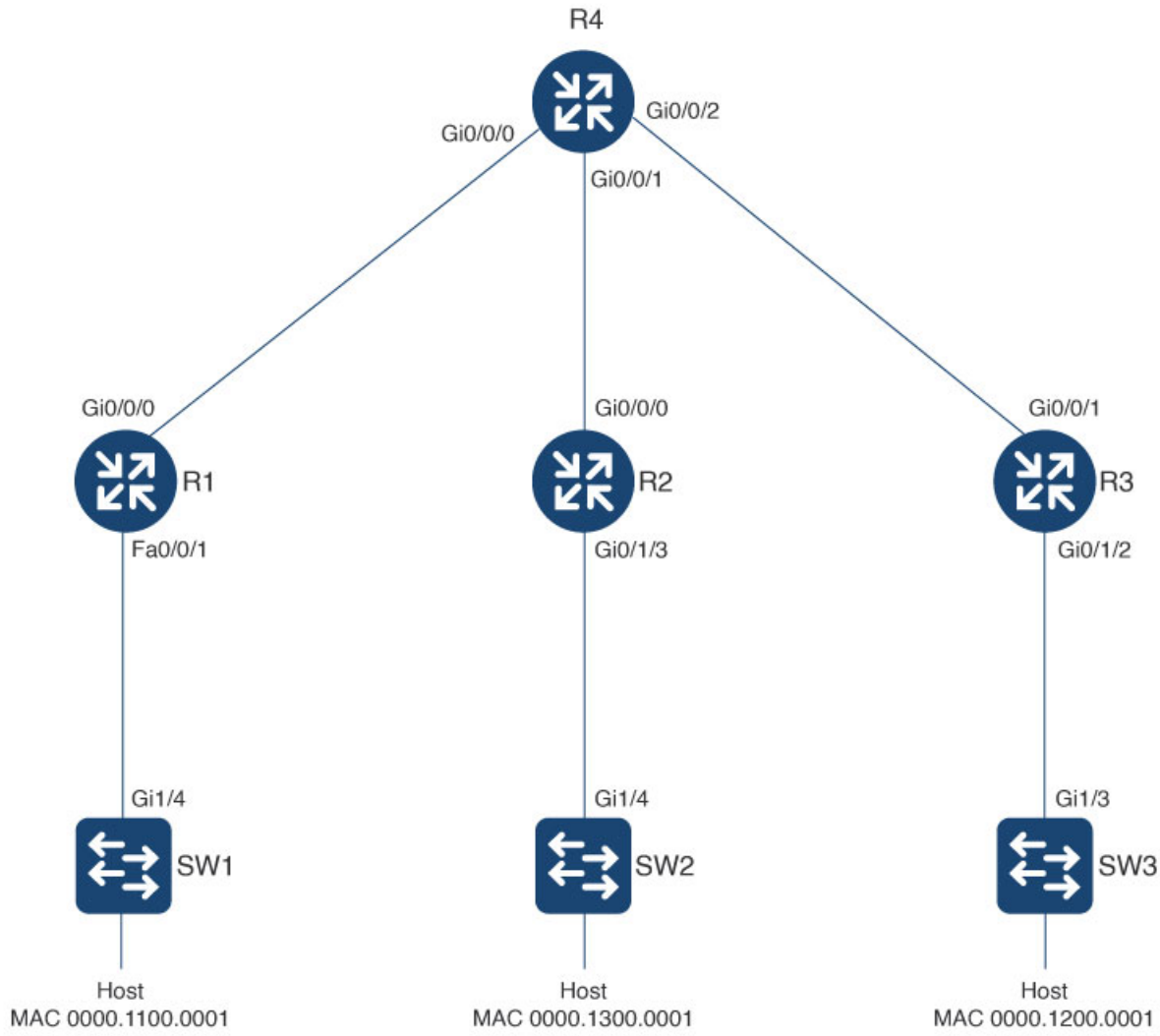
Example:

```
l2vpn evpn instance 1 vlan-aware
 encapsulation vxlan
 replication-type static
!
interface nve1
 no ip address
 source-interface Loopback0
 host-reachability protocol bgp
 member vni 30000 mcast-group 209.165.1.1
 member vni 30012 mcast-group 209.165.1.1
!
```

Configuration Example

Here is an example of configuring the VLAN aware feature over VXLAN EVPN in a network topology with three routers (R1, R2, and R3) as the VTEPs (Virtual Tunnel Endpoint), connected to three switchports (SW1, SW2, and SW3) which are connected to the host, while one router (R4) acts as the *spine* as shown in the figure.

Figure 24: Configuration example of 4 router topology



Here is the running configuration for Router 1 (R1)

```
l2vpn evpn
 replication-type ingress
!
l2vpn evpn profile evpn_va vlan-aware
 evi-id 3
 l2vni-base 50000
 ethernet-tag auto-vni
!
bridge-domain 12
 member Vlan12 service-instance 12
 member evpn-instance profile evpn_va
!
bridge-domain 22
 member Vlan22 service-instance 22
 member evpn-instance profile evpn_va

interface Loopback0
 ip address 192.0.2.1 255.255.255.255
```

```
!
interface GigabitEthernet0/0/0
 ip address 10.10.10.2 255.255.255.0
!
!
interface FastEthernet0/0/1
 switchport trunk allowed vlan 12,22
 switchport mode trunk
!
interface Vlan12
 no ip address
 service instance 12 ethernet
  encapsulation dot1q 12
 !
!
interface Vlan22
 no ip address
 service instance 22 ethernet
  encapsulation dot1q 22
 !
interface nve1
 no ip address
 source-interface Loopback0
 host-reachability protocol bgp
 member vni 30000 ingress-replication
!
router ospf 1
 router-id 192.0.2.1
 network 10.10.10.0 0.0.0.255 area 0
 network 192.0.2.1 0.0.0.0 area 0
!
router bgp 1
 bgp router-id 192.0.2.1
 bgp log-neighbor-changes
 neighbor 192.0.2.4 remote-as 1
 neighbor 192.0.2.4 update-source Loopback0
 !
 address-family ipv4
  neighbor 192.0.2.4 activate
  neighbor 192.0.2.4 send-community both
 exit-address-family
!
address-family l2vpn evpn
  neighbor 192.0.2.4 activate
  neighbor 192.0.2.4 send-community both
 exit-address-family
!
```

Here is the running configuration for Router 2 (R2)

```
l2vpn evpn
 replication-type ingress
!
l2vpn evpn profile evpn_va vlan-aware
 evi-id 3
 l2vni-base 50000
 ethernet-tag auto-vni
!
bridge-domain 12
 member Vlan12 service-instance 12
 member evpn-instance profile evpn_va
!
bridge-domain 22
 member Vlan22 service-instance 22
 member evpn-instance profile evpn_va
```

```
!
interface Loopback0
 ip address 192.0.2.2 255.255.255.255
!
interface GigabitEthernet0/0/0
 ip address 10.10.20.2 255.255.255.0
!
interface GigabitEthernet0/1/3
 switchport trunk allowed vlan 12,22
 switchport mode trunk
!
interface Vlan12
 no ip address
 service instance 12 ethernet
  encapsulation dot1q 12
 !
!
interface Vlan22
 no ip address
 service instance 22 ethernet
  encapsulation dot1q 22
 !
interface nve1
 no ip address
 source-interface Loopback0
 host-reachability protocol bgp
 member vni 30000 ingress-replication
!
router ospf 1
 router-id 192.0.2.2
 network 10.10.20.0 0.0.0.255 area 0
 network 192.0.2.2 0.0.0.0 area 0
!
router bgp 1
 bgp router-id 192.0.2.2
 bgp log-neighbor-changes
 neighbor 192.0.2.4 remote-as 1
 neighbor 192.0.2.4 update-source Loopback0
 !
 address-family ipv4
  neighbor 192.0.2.4 activate
  neighbor 192.0.2.4 send-community both
 exit-address-family
!
address-family l2vpn evpn
  neighbor 192.0.2.4 activate
  neighbor 192.0.2.4 send-community both
 exit-address-family
!
```

Here is the running configuration for Router 3 (R3)

```
l2vpn evpn
 replication-type ingress
!
l2vpn evpn profile evpn_va vlan-aware
 evi-id 3
 l2vni-base 50000
 ethernet-tag auto-vni
!
bridge-domain 12
 member Vlan12 service-instance 12
 member evpn-instance profile evpn_va
!
bridge-domain 22
```

```
 member Vlan22 service-instance 22
 member evpn-instance profile evpn_va

interface Loopback0
 ip address 192.0.2.3 255.255.255.255
!
interface GigabitEthernet0/0/1
 ip address 10.10.30.2 255.255.255.0
interface GigabitEthernet0/1/2
 switchport trunk allowed vlan 12,22
 switchport mode trunk
!
interface Vlan12
 no ip address
 service instance 12 ethernet
  encapsulation dot1q 12
 !
!
interface Vlan22
 no ip address
 service instance 22 ethernet
  encapsulation dot1q 22
 !
interface nve1
 no ip address
 source-interface Loopback0
 host-reachability protocol bgp
 member vni 30000 ingress-replication
!
router ospf 1
 router-id 192.0.2.3
 network 10.10.30.0 0.0.0.255 area 0
 network 192.0.2.3 0.0.0.0 area 0
!
router bgp 1
 bgp router-id 192.0.2.3
 bgp log-neighbor-changes
 neighbor 192.0.2.4 remote-as 1
 neighbor 192.0.2.4 update-source Loopback0
 !
 address-family ipv4
  neighbor 192.0.2.4 activate
  neighbor 192.0.2.4 send-community both
 exit-address-family
 !
 address-family l2vpn evpn
  neighbor 192.0.2.4 activate
  neighbor 192.0.2.4 send-community both
 exit-address-family
!
```

Here is the running configuration for the Router 4 (R4) which is acting as the *spine*.

```
interface Loopback0
 ip address 192.0.2.4 255.255.255.255
!
interface GigabitEthernet0/0/0
 ip address 10.10.10.1 255.255.255.0
 negotiation auto
!
interface GigabitEthernet0/0/1
 ip address 10.10.20.1 255.255.255.0
 negotiation auto
!
```

```
interface GigabitEthernet0/0/2
 ip address 10.10.30.1 255.255.255.0
 negotiation auto
!
router ospf 1
 router-id 192.0.2.4
 network 10.10.10.0 0.0.0.255 area 0
 network 10.10.20.0 0.0.0.255 area 0
 network 10.10.30.0 0.0.0.255 area 0
 network 192.0.2.4 0.0.0.0 area 0
!
router bgp 1
 bgp router-id 192.0.2.4
 bgp log-neighbor-changes
 neighbor 192.0.2.1 remote-as 1
 neighbor 192.0.2.1 update-source Loopback0
 neighbor 192.0.2.2 remote-as 1
 neighbor 192.0.2.2 update-source Loopback0
 neighbor 192.0.2.3 remote-as 1
 neighbor 192.0.2.3 update-source Loopback0
 !
address-family l2vpn evpn
  neighbor 192.0.2.1 activate
  neighbor 192.0.2.1 send-community both
  neighbor 192.0.2.1 route-reflector-client
  neighbor 192.0.2.2 activate
  neighbor 192.0.2.2 send-community both
  neighbor 192.0.2.2 route-reflector-client
  neighbor 192.0.2.3 activate
  neighbor 192.0.2.3 send-community both
 exit-address-family
!
```

Here is the running configuration for the switchport SW1, connected to Router R1.

```
interface GigabitEthernet1/4
 switchport trunk allowed vlan 12,22
 switchport mode trunk
!
interface GigabitEthernet1/7
 switchport trunk allowed vlan 12,22
 switchport mode trunk
!
```

Here is the running configuration for the switchport SW2, connected to Router R2.

```
interface GigabitEthernet1/4
 switchport trunk allowed vlan 12,22
 switchport mode trunk
!
interface GigabitEthernet1/7
 switchport trunk allowed vlan 12,22
 switchport mode trunk
!
```

Here is the running configuration for the switchport SW3, connected to Router R3.

```
 switchport trunk allowed vlan 12,22
 switchport mode trunk
!
interface GigabitEthernet1/7
 switchport trunk allowed vlan 12,22
 switchport mode trunk
!
```

Verify Configuration of EVPN VXLAN VLAN-Aware Service

Verify the configuration on the router R1

Use the **show l2vpn evpn evi 3 detail** command to verify the configuration of the EVI and bridge-domain, and to ensure the ethernet-tag value is configured.

```
Router1#show l2vpn evpn evi 3 detail
EVPN instance:       3 (VLAN Aware)
  Profile:           evpn_va
  RD:                10.10.10.2:32770 (auto)
  Import-RTs:        1:3
  Export-RTs:        1:3
  Per-EVI Label:     none
  State:             Established
  Replication Type:  Ingress (profile)
  Encapsulation:     vxlan (profile)
  IP Local Learn:    Enabled (global)
  Adv. Def. Gateway: Disabled (global)
  Re-originate RT5:  Disabled (profile)
  AR Flood Suppress: Enabled (global)
  Bridge Domain:     12
    Ethernet-Tag:    50012
    State:           Established
    Flood Suppress:  Attached
    Core If:
    Access If:
    NVE If:          nve1
    RMAC:            0000.0000.0000
    Core BD:         0
    L2 VNI:          50012
    L3 VNI:          0
    VTEP IP:         192.0.2.1
    Pseudoports:
      Vlan12 service instance 12
        Routes: 0 MAC, 0 MAC/IP
    Peers:
      192.0.2.2
        Routes: 0 MAC, 0 MAC/IP, 1 IMET, 0 EAD
      192.0.2.3
        Routes: 1 MAC, 0 MAC/IP, 1 IMET, 0 EAD
  Bridge Domain:     22
    Ethernet-Tag:    50022
    State:           Established
    Flood Suppress:  Attached
    Core If:
    Access If:
    NVE If:          nve1
    RMAC:            0000.0000.0000
    Core BD:         0
    L2 VNI:          50022
    L3 VNI:          0
    VTEP IP:         192.0.2.1
    Pseudoports:
      Vlan22 service instance 22
        Routes: 0 MAC, 0 MAC/IP
    Peers:
      192.0.2.2
        Routes: 0 MAC, 0 MAC/IP, 1 IMET, 0 EAD
      192.0.2.3
        Routes: 0 MAC, 0 MAC/IP, 1 IMET, 0 EAD
```

Use the **show l2route evpn imet** command to verify the IMET routes.

```
Router1#show l2route evpn imet
EVI       ETAG   Prod                              Router IP Addr  Type    Label
                 Tunnel ID Multicast Proxy
----- ---------- ------ ---------------------------------------- ----- --------
-------------------------------------- ---------------
    3      50012    BGP                                 10.10.20.2     6    50012
                   192.0.2.2             No
    3      50012    BGP                                 10.10.30.2     6    50012
                   192.0.2.3             No
    3      50012  L2VPN                                 10.10.10.2     6    50012
                   192.0.2.1             No
    3      50022    BGP                                 10.10.20.2     6    50022
                   192.0.2.2             No
    3      50022    BGP                                 10.10.30.2     6    50022
                   192.0.2.3             No
    3      50022  L2VPN                                 10.10.10.2     6    50022
                   192.0.2.1             No
```

Use the **show ip bgp l2vpn evpn all** command to verify the configuration.

```
Router1#show ip bgp l2vpn evpn all
BGP table version is 31, local router ID is 192.0.2.1
Status codes: s suppressed, d damped, h history, * valid, > best, i - internal,
              r RIB-failure, S Stale, m multipath, b backup-path, f RT-Filter,
              x best-external, a additional-path, c RIB-compressed,
              t secondary path, L long-lived-stale,
Origin codes: i - IGP, e - EGP, ? - incomplete
RPKI validation codes: V valid, I invalid, N Not found

     Network          Next Hop            Metric LocPrf Weight Path
Route Distinguisher: 192.0.2.1:32770
 *>   [2][192.0.2.1:32770][50012][48][000011000001][0][*]/20
                       0.0.0.0                                32768 ?
 *>i  [2][192.0.2.1:32770][50012][48][000012000001][0][*]/20
                       192.0.2.3              0    100      0 ?
 *>i  [2][192.0.2.1:32770][50012][48][000013000001][0][*]/20
                       192.0.2.2              0    100      0 ?
 *>i  [2][192.0.2.1:32770][50012][48][04BD9708512B][0][*]/20
                       192.0.2.3              0    100      0 ?
 *>   [2][192.0.2.1:32770][50022][48][000011000002][0][*]/20
                       0.0.0.0                                32768 ?
 *>i  [2][192.0.2.1:32770][50022][48][000012000002][0][*]/20
                       192.0.2.3              0    100      0 ?
 *>i  [2][192.0.2.1:32770][50022][48][000013000002][0][*]/20
                       192.0.2.2              0    100      0 ?
Route Distinguisher: 192.0.2.2:32770
 *>i  [2][192.0.2.2:32770][50012][48][000013000001][0][*]/20
                       192.0.2.2              0    100      0 ?
 *>i  [2][192.0.2.2:32770][50022][48][000013000002][0][*]/20
                       192.0.2.2              0    100      0 ?
Route Distinguisher: 192.0.2.3:32770
 *>i  [2][192.0.2.3:32770][50012][48][000012000001][0][*]/20
                       192.0.2.3              0    100      0 ?
 *>i  [2][192.0.2.3:32770][50012][48][04BD9708512B][0][*]/20
                       192.0.2.3              0    100      0 ?
 *>i  [2][192.0.2.3:32770][50022][48][000012000002][0][*]/20
                       192.0.2.3              0    100      0 ?
Route Distinguisher: 192.0.2.1:32770
 *>   [3][192.0.2.1:32770][50012][32][192.0.2.1]/17
                       0.0.0.0                                32768 ?
 *>i  [3][192.0.2.1:32770][50012][32][192.0.2.2]/17
                       192.0.2.2              0    100      0 ?
 *>i  [3][192.0.2.1:32770][50012][32][192.0.2.3]/17
                       192.0.2.3              0    100      0 ?
 *>   [3][192.0.2.1:32770][50022][32][192.0.2.1]/17
```

```
                     0.0.0.0                                32768 ?
 *>i  [3][192.0.2.1:32770][50022][32][192.0.2.2]/17
                     192.0.2.2              0    100       0 ?
 *>i  [3][192.0.2.1:32770][50022][32][192.0.2.3]/17
                     192.0.2.3              0    100       0 ?
Route Distinguisher: 192.0.2.2:32770
 *>i  [3][192.0.2.2:32770][50012][32][192.0.2.2]/17
                     192.0.2.2              0    100       0 ?
 *>i  [3][192.0.2.2:32770][50022][32][192.0.2.2]/17
                     192.0.2.2              0    100       0 ?
Route Distinguisher: 192.0.2.3:32770
 *>i  [3][192.0.2.3:32770][50012][32][192.0.2.3]/17
                     192.0.2.3              0    100       0 ?
 *>i  [3][192.0.2.3:32770][50022][32][192.0.2.3]/17
                     192.0.2.3              0    100       0 ?
```

Use the **show nve peers** command to verify the configuration.

```
Router1#show nve peers
'M' - MAC entry download flag  'A' - Adjacency download flag
'4' - IPv4 flag  '6' - IPv6 flag

Interface  VNI      Type Peer-IP          RMAC/Num_RTs   eVNI     state flags UP time
nve1       50012    L2CP 192.0.2.2    1              50012      UP   N/A  00:38:47
nve1       50012    L2CP 192.0.2.3    1              50012      UP   N/A  00:38:47
nve1       50022    L2CP 192.0.2.2    1              50022      UP   N/A  00:38:47
nve1       50022    L2CP 192.0.2.3    1              50022      UP   N/A  00:38:47
```

Use the **show l2vpn evpn mac** command to verify the configuration.

```
Router1#show l2vpn evpn mac
MAC Address    EVI   BD    ESI                       Ether Tag  Next Hop(s)
-------------- ----- ----- ------------------------ ---------- ---------------
0000.1100.0001 3     12    0000.0000.0000.0000.0000 50012      Vl12:12
0000.1200.0001 3     12    0000.0000.0000.0000.0000 50012      192.0.2.3
0000.1300.0001 3     12    0000.0000.0000.0000.0000 50012      192.0.2.2
04bd.9708.512b 3     12    0000.0000.0000.0000.0000 50012      192.0.2.3
0000.1100.0002 3     22    0000.0000.0000.0000.0000 50022      Vl22:22
0000.1200.0002 3     22    0000.0000.0000.0000.0000 50022      192.0.2.3
0000.1300.0002 3     22    0000.0000.0000.0000.0000 50022      192.0.2.2
```

Verify the configuration on the router R2

Use the **show l2vpn evpn evi 3 detail** command to verify the configuration of the EVI and bridge-domain, and to ensure the ethernet-tag value is configured.

```
Router2#show l2vpn evpn evi 3 detail

EVPN instance:       3 (VLAN Aware)
  Profile:           evpn_va
  RD:                10.10.20.2:32770 (auto)
  Import-RTs:        1:3
  Export-RTs:        1:3
  Per-EVI Label:     none
  State:             Established
  Replication Type:  Ingress (profile)
  Encapsulation:     vxlan (profile)
  IP Local Learn:    Enabled (global)
  Adv. Def. Gateway: Disabled (global)
  Re-originate RT5:  Disabled (profile)
  AR Flood Suppress: Enabled (global)
  Bridge Domain:     12
    Ethernet-Tag:    50012
    State:           Established
    Flood Suppress:  Attached
```

```
    Core If:
    Access If:
    NVE If:          nve1
    RMAC:            0000.0000.0000
    Core BD:         0
    L2 VNI:          50012
    L3 VNI:          0
    VTEP IP:         192.0.2.2
    Pseudoports:
      Vlan12 service instance 12
        Routes: 0 MAC, 0 MAC/IP
    Peers:
      192.0.2.1
        Routes: 0 MAC, 0 MAC/IP, 1 IMET, 0 EAD
      192.0.2.3
        Routes: 1 MAC, 0 MAC/IP, 1 IMET, 0 EAD
  Bridge Domain:     22
    Ethernet-Tag:    50022
    State:           Established
    Flood Suppress:  Attached
    Core If:
    Access If:
    NVE If:          nve1
    RMAC:            0000.0000.0000
    Core BD:         0
    L2 VNI:          50022
    L3 VNI:          0
    VTEP IP:         192.0.2.2
    Pseudoports:
      Vlan22 service instance 22
        Routes: 0 MAC, 0 MAC/IP
    Peers:
      192.0.2.1
        Routes: 0 MAC, 0 MAC/IP, 1 IMET, 0 EAD
      192.0.2.3
        Routes: 0 MAC, 0 MAC/IP, 1 IMET, 0 EAD
```

Use the **show l2route evpn imet** command to verify the IMET routes.

```
Router2#show l2route evpn imet
 EVI       ETAG   Prod                              Router IP Addr  Type    Label
                  Tunnel ID Multicast Proxy
----- ---------- ------ ---------------------------------------- ----- --------
--------------------------------------- ----------------
    3      50012    BGP                                  10.10.10.2     6    50012
                   192.0.2.1              No
    3      50012    BGP                                  10.10.30.2     6    50012
                   192.0.2.3              No
    3      50012  L2VPN                                  10.10.20.2     6    50012
                   192.0.2.2              No
    3      50022    BGP                                  10.10.10.2     6    50022
                   192.0.2.1              No
    3      50022    BGP                                  10.10.30.2     6    50022
                   192.0.2.3              No
    3      50022  L2VPN                                  10.10.20.2     6    50022
                   192.0.2.2              No
```

Use the **show ip bgp l2vpn evpn all** command to verify the configuration.

```
Router2#sh ip bgp l2vpn evpn all
BGP table version is 27, local router ID is 192.0.2.2
Status codes: s suppressed, d damped, h history, * valid, > best, i - internal,
              r RIB-failure, S Stale, m multipath, b backup-path, f RT-Filter,
              x best-external, a additional-path, c RIB-compressed,
```

```
              t secondary path, L long-lived-stale,
Origin codes: i - IGP, e - EGP, ? - incomplete
RPKI validation codes: V valid, I invalid, N Not found

     Network          Next Hop            Metric LocPrf Weight Path
Route Distinguisher: 192.0.2.1:32770
 *>i  [2][192.0.2.1:32770][50012][48][000011000001][0][*]/20
                      192.0.2.1                0    100      0 ?
 *>i  [2][192.0.2.1:32770][50022][48][000011000002][0][*]/20
                      192.0.2.1                0    100      0 ?
Route Distinguisher: 192.0.2.2:32770
 *>i  [2][192.0.2.2:32770][50012][48][000011000001][0][*]/20
                      192.0.2.1                0    100      0 ?
 *>i  [2][192.0.2.2:32770][50012][48][000012000001][0][*]/20
                      192.0.2.3                0    100      0 ?
 *>   [2][192.0.2.2:32770][50012][48][000013000001][0][*]/20
                      0.0.0.0                               32768 ?
 *>i  [2][192.0.2.2:32770][50012][48][04BD9708512B][0][*]/20
                      192.0.2.3                0    100      0 ?
 *>i  [2][192.0.2.2:32770][50022][48][000011000002][0][*]/20
                      192.0.2.1                0    100      0 ?
 *>i  [2][192.0.2.2:32770][50022][48][000012000002][0][*]/20
                      192.0.2.3                0    100      0 ?
 *>   [2][192.0.2.2:32770][50022][48][000013000002][0][*]/20
                      0.0.0.0                               32768 ?
Route Distinguisher: 192.0.2.3:32770
 *>i  [2][192.0.2.3:32770][50012][48][000012000001][0][*]/20
                      192.0.2.3                0    100      0 ?
 *>i  [2][192.0.2.3:32770][50012][48][04BD9708512B][0][*]/20
                      192.0.2.3                0    100      0 ?
 *>i  [2][192.0.2.3:32770][50022][48][000012000002][0][*]/20
                      192.0.2.3                0    100      0 ?
Route Distinguisher: 192.0.2.1:32770
 *>i  [3][192.0.2.1:32770][50012][32][192.0.2.1]/17
                      192.0.2.1                0    100      0 ?
 *>i  [3][192.0.2.1:32770][50022][32][192.0.2.1]/17
                      192.0.2.1                0    100      0 ?
Route Distinguisher: 192.0.2.2:32770
 *>i  [3][192.0.2.2:32770][50012][32][192.0.2.1]/17
                      192.0.2.1                0    100      0 ?
 *>   [3][192.0.2.2:32770][50012][32][192.0.2.2]/17
                      0.0.0.0                               32768 ?
 *>i  [3][192.0.2.2:32770][50012][32][192.0.2.3]/17
                      192.0.2.3                0    100      0 ?
 *>i  [3][192.0.2.2:32770][50022][32][192.0.2.1]/17
                      192.0.2.1                0    100      0 ?
 *>   [3][192.0.2.2:32770][50022][32][192.0.2.2]/17
                      0.0.0.0                               32768 ?
 *>i  [3][192.0.2.2:32770][50022][32][192.0.2.3]/17
                      192.0.2.3                0    100      0 ?
Route Distinguisher: 192.0.2.3:32770
 *>i  [3][192.0.2.3:32770][50012][32][192.0.2.3]/17
                      192.0.2.3                0    100      0 ?
 *>i  [3][192.0.2.3:32770][50022][32][192.0.2.3]/17
                      192.0.2.3                0    100      0 ?
```

Use the **show nve peers** command to verify the configuration.

```
Router2#show nve peers
'M' - MAC entry download flag  'A' - Adjacency download flag
'4' - IPv4 flag  '6' - IPv6 flag

Interface  VNI      Type Peer-IP          RMAC/Num_RTs   eVNI     state flags UP time
nve1       50012    L2CP 192.0.2.1    2              50012      UP   N/A  00:05:28
nve1       50012    L2CP 192.0.2.3    3              50012      UP   N/A  00:05:28
```

```
nve1       50022    L2CP 192.0.2.1    2              50022      UP   N/A  00:05:28
nve1       50022    L2CP 192.0.2.3    2              50022      UP   N/A  00:05:28
```

Use the **show l2vpn evpn mac** command to verify the configuration

```
Router2#show l2vpn evpn mac
MAC Address    EVI   BD    ESI                       Ether Tag  Next Hop(s)
-------------- ----- ----- ------------------------ ---------- ---------------
0000.1100.0001 3     12    0000.0000.0000.0000.0000 50012      192.0.2.1
0000.1200.0001 3     12    0000.0000.0000.0000.0000 50012      192.0.2.3
0000.1300.0001 3     12    0000.0000.0000.0000.0000 50012      Vl12:12
04bd.9708.512b 3     12    0000.0000.0000.0000.0000 50012      192.0.2.3
0000.1100.0002 3     22    0000.0000.0000.0000.0000 50022      192.0.2.1
0000.1200.0002 3     22    0000.0000.0000.0000.0000 50022      192.0.2.3
0000.1300.0002 3     22    0000.0000.0000.0000.0000 50022      Vl22:22
```

Verify the configuration on the router R3

Use the **show l2vpn evpn evi 3 detail** command to verify the configuration of the EVI and bridge-domain, and to ensure the ethernet-tag value is configured.

```
Router3#show l2vpn evpn evi 3 detail
EVPN instance:       3 (VLAN Aware)
  Profile:           evpn_va
  RD:                10.10.30.2:32770 (auto)
  Import-RTs:        1:3
  Export-RTs:        1:3
  Per-EVI Label:     none
  State:             Established
  Replication Type:  Ingress (profile)
  Encapsulation:     vxlan (profile)
  IP Local Learn:    Enabled (global)
  Adv. Def. Gateway: Disabled (global)
  Re-originate RT5:  Disabled (profile)
  AR Flood Suppress: Enabled (global)
  Bridge Domain:     12
    Ethernet-Tag:    50012
    State:           Established
    Flood Suppress:  Attached
    Core If:
    Access If:
    NVE If:          nve1
    RMAC:            0000.0000.0000
    Core BD:         0
    L2 VNI:          50012
    L3 VNI:          0
    VTEP IP:         192.0.2.3
    Pseudoports:
      Vlan12 service instance 12
        Routes: 1 MAC, 0 MAC/IP
    Peers:
      192.0.2.1
        Routes: 0 MAC, 0 MAC/IP, 1 IMET, 0 EAD
      192.0.2.2
        Routes: 0 MAC, 0 MAC/IP, 1 IMET, 0 EAD
  Bridge Domain:     22
    Ethernet-Tag:    50022
    State:           Established
    Flood Suppress:  Attached
    Core If:
    Access If:
    NVE If:          nve1
    RMAC:            0000.0000.0000
    Core BD:         0
```

```
    L2 VNI:           50022
    L3 VNI:           0
    VTEP IP:          192.0.2.3
    Pseudoports:
      Vlan22 service instance 22
        Routes: 0 MAC, 0 MAC/IP
    Peers:
      192.0.2.1
        Routes: 0 MAC, 0 MAC/IP, 1 IMET, 0 EAD
      192.0.2.2
        Routes: 0 MAC, 0 MAC/IP, 1 IMET, 0 EAD
```

Use the **show l2route evpn imet** command to verify the IMET routes.

```
Router3#show l2route evpn imet
  EVI       ETAG   Prod                              Router IP Addr  Type    Label
                   Tunnel ID Multicast Proxy
----- ---------- ------ ---------------------------------------- ----- --------
---------------------------------------- ---------------
    3      50012    BGP                                  10.10.10.2     6    50012
                   192.0.2.1              No
    3      50012    BGP                                  10.10.20.2     6    50012
                   192.0.2.2              No
    3      50012  L2VPN                                  10.10.30.2     6    50012
                   192.0.2.3              No
    3      50022    BGP                                  10.10.10.2     6    50022
                   192.0.2.1              No
    3      50022    BGP                                  10.10.20.2     6    50022
                   192.0.2.2              No
    3      50022  L2VPN                                  10.10.30.2     6    50022
                   192.0.2.3              No
```

Use the **show ip bgp l2vpn evpn all** command to verify the configuration.

```
Router3# sh ip bgp l2vpn evpn all
BGP table version is 30, local router ID is 192.0.2.3
Status codes: s suppressed, d damped, h history, * valid, > best, i - internal,
              r RIB-failure, S Stale, m multipath, b backup-path, f RT-Filter,
              x best-external, a additional-path, c RIB-compressed,
              t secondary path, L long-lived-stale,
Origin codes: i - IGP, e - EGP, ? - incomplete
RPKI validation codes: V valid, I invalid, N Not found

     Network          Next Hop            Metric LocPrf Weight Path
Route Distinguisher: 192.0.2.1:32770
 *>i  [2][192.0.2.1:32770][50012][48][000011000001][0][*]/20
                      192.0.2.1                0    100      0 ?
 *>i  [2][192.0.2.1:32770][50022][48][000011000002][0][*]/20
                      192.0.2.1                0    100      0 ?
Route Distinguisher: 192.0.2.2:32770
 *>i  [2][192.0.2.2:32770][50012][48][000013000001][0][*]/20
                      192.0.2.2                0    100      0 ?
 *>i  [2][192.0.2.2:32770][50022][48][000013000002][0][*]/20
                      192.0.2.2                0    100      0 ?
Route Distinguisher: 192.0.2.3:32770
 *>i  [2][192.0.2.3:32770][50012][48][000011000001][0][*]/20
                      192.0.2.1                0    100      0 ?
 *>   [2][192.0.2.3:32770][50012][48][000012000001][0][*]/20
                      0.0.0.0                            32768 ?
 *>i  [2][192.0.2.3:32770][50012][48][000013000001][0][*]/20
                      192.0.2.2                0    100      0 ?
 *>   [2][192.0.2.3:32770][50012][48][04BD9708512B][0][*]/20
                      0.0.0.0                            32768 ?
 *>i  [2][192.0.2.3:32770][50022][48][000011000002][0][*]/20
                      192.0.2.1                0    100      0 ?
```

```
 *>   [2][192.0.2.3:32770][50022][48][000012000002][0][*]/20
                      0.0.0.0                              32768 ?
 *>i  [2][192.0.2.3:32770][50022][48][000013000002][0][*]/20
                      192.0.2.2            0    100      0 ?
Route Distinguisher: 192.0.2.1:32770
 *>i  [3][192.0.2.1:32770][50012][32][192.0.2.1]/17
                      192.0.2.1            0    100      0 ?
 *>i  [3][192.0.2.1:32770][50022][32][192.0.2.1]/17
                      192.0.2.1            0    100      0 ?
Route Distinguisher: 192.0.2.2:32770
 *>i  [3][192.0.2.2:32770][50012][32][192.0.2.2]/17
                      192.0.2.2            0    100      0 ?
 *>i  [3][192.0.2.2:32770][50022][32][192.0.2.2]/17
                      192.0.2.2            0    100      0 ?
Route Distinguisher: 192.0.2.3:32770
 *>i  [3][192.0.2.3:32770][50012][32][192.0.2.1]/17
                      192.0.2.1            0    100      0 ?
 *>i  [3][192.0.2.3:32770][50012][32][192.0.2.2]/17
                      192.0.2.2            0    100      0 ?
 *>   [3][192.0.2.3:32770][50012][32][192.0.2.3]/17
                      0.0.0.0                              32768 ?
 *>i  [3][192.0.2.3:32770][50022][32][192.0.2.1]/17
                      192.0.2.1            0    100      0 ?
 *>i  [3][192.0.2.3:32770][50022][32][192.0.2.2]/17
                      192.0.2.2            0    100      0 ?
 *>   [3][192.0.2.3:32770][50022][32][192.0.2.3]/17
                      0.0.0.0                              32768 ?
```

Use the **show nve peers** command to verify the configuration.

```
Router3#show nve peers
'M' - MAC entry download flag  'A' - Adjacency download flag
'4' - IPv4 flag  '6' - IPv6 flag

Interface  VNI      Type Peer-IP          RMAC/Num_RTs   eVNI     state flags UP time
nve1       50012    L2CP 192.0.2.1    2              50012      UP   N/A  00:07:48
nve1       50012    L2CP 192.0.2.2    2              50012      UP   N/A  00:05:23
nve1       50022    L2CP 192.0.2.1    2              50022      UP   N/A  00:07:48
nve1       50022    L2CP 192.0.2.2    2              50022      UP   N/A  00:05:23
```

Use the **show l2vpn evpn mac** command to verify the configuration

```
Router3# show l2vpn evpn mac
MAC Address    EVI   BD    ESI                       Ether Tag  Next Hop(s)
-------------- ----- ----- ------------------------ ---------- ---------------
0000.1100.0001 3     12    0000.0000.0000.0000.0000 50012      192.0.2.1
0000.1200.0001 3     12    0000.0000.0000.0000.0000 50012      Vl12:12
0000.1300.0001 3     12    0000.0000.0000.0000.0000 50012      192.0.2.2
0000.1100.0002 3     22    0000.0000.0000.0000.0000 50022      192.0.2.1
0000.1200.0002 3     22    0000.0000.0000.0000.0000 50022      Vl22:22
0000.1300.0002 3     22    0000.0000.0000.0000.0000 50022      192.0.2.2
```

Multi-Homing Configuration in BGP EVPN VXLAN Fabric

Ethernet Virtual Private Network (EVPN) has gained prominence in contemporary networking due to its capacity to deliver scalable, flexible, and efficient Layer 2 and Layer 3 VPN services over an IP/MPLS backbone. EVPN is commonly integrated with Virtual Extensible LAN (VXLAN), a widely-used network virtualization overlay protocol that significantly enhances the Layer 2 network address space.

EVPN-VXLAN operates as an open standards technology addressing the constraints of traditional VLAN-based networks. It establishes network fabric that extends Layer 2 connectivity as an overlay on existing physical networks.

Why Multi-Homing?

Multi-homing is critical for effectively implementing the combined two-stage VLAN and Bridge Domain (BD) model in routing platforms. A traditional single-model procedure for EVPN is designed exclusively for either BD-only or VLAN-only configurations. Additionally, there are limited numbers of routing ports available, which means that switchports must be utilized to enable effective Multi-homing deployments.

By configuring EVPN Ethernet Segments on switchports and associating them with VLANs which are mapped to Switched Virtual Interfaces (SVIs) under the EVPN BD, multi-homing facilitates increased resilience and optimized network resource utilization. The Multi-Homing (MH) All-Active Ethernet Segment feature offers redundancy for connections between hosts or Layer 2 switches and the EVPN VXLAN network

Significance of DF Election and Split-Horizon in Multi-Homing Deployments

Designated Forwarder (DF) election and split-horizon techniques are essential in multi-homing deployments to prevent traffic loops and duplication. In scenarios where multiple switchports are configured as members of a multi-homing VLAN, ambiguity arises regarding the intended traffic destination. To maintain network stability, only one switchport can be actively supported to handle traffic within the multi-homing VLAN, while local switching is disabled.

The switchport functions as the access interface for the EVPN Ethernet Segment, while the associated SVI Ethernet Forwarding Port (EFP) represents the member pseudoport of the EVPN Ethernet Virtual Instance (EVI) or BD linked to the EVPN Ethernet Segment.

The implementation of DF election and split-horizon at the SVI EFP level ensures that traffic is efficiently managed, significantly reducing the risk of loops and optimizing network performance in multi-homing configurations.

Configuring Multi-Homing in a BGP EVPN VXLAN Fabric

To configure multi-homing with all-active redundancy in a BGP EVPN VXLAN fabric, perform the following set of procedures:

1. Configure Ethernet Segment and Redundancy in the Ethernet Segment
2. Configure Multi-homing VLAN and Associate EVPN Ethernet-Segment
   - Access Mode
   - Trunk Mode
3. Configure SVI Service Instance
4. Configure EVPN-Instance
   - Profile based Configuration
   - Manual Configuration
5. Apply the configuration on a Bridge-Domain

Removing a Multi-homing VLAN

Unconfiguring the EVPN Ethernet-segment on the switchport will remove the corresponding multihoming VLANs.

Configure Ethernet Segment and Redundancy in the Ethernet Segment

Follow these steps to configure redundancy on an ethernet segment on your router.

Procedure

Step 1 Enter the privileged EXEC mode and enter the password, if prompted.

Example:

```
Router#enable
```

Step 2 Enter the Global Configuration Mode.

Example:

```
Router#configure terminal
```

Step 3 Enter the Layer 2 VPN EVPN ethernet segment configuration mode.

Example:

```
Router(config)#l2vpn evpn ethernet-segment 1
```

Step 4 Configure the ethernet segment identifier type (ESI) and value for the ethernet segment.

Example:

```
Router(config-evpn-es)#l2vpn evpn ethernet-segment 1
```

The following ESI types are supported:

- Type **0**: This type indicates an arbitrary 9-octet ESI value. The format is 00 + 9-octets of ESI value
- Type **3**: This type indicates a MAC-based ESI Value. The format is 03 + system-mac (6 bytes) + value of MAC address (3 bytes).

Step 5 Configure the redundancy type for the ethernet segment.

Example:

```
Router(config-evpn-es)#redundancy all-active
```

Step 6 Exit the Layer 2 VPN EVPN ethernet segment configuration mode and enter privileged EXEC mode.

Example:

```
Router(config-evpn-es)#end
```

Configure Multi-Homing VLAN and Associate EVPN Ethernet-Segment

You can add a multi-homing VLAN through two modes:

- Access Mode: The switchport must be the only port in the Access VLAN.

If the switchport is not the only port in the access VLAN, the command line interface (CLI) request to configure EVPN Ethernet-segment will be rejected.

- Trunk Mode: The switchport must be the only port for all the VLANs which are allowed on the trunk switchport.

  If the switchport is not the only port associated with the VLANs which are allowed on the trunk, the CLI request to configure EVPN Ethernet-segment will be rejected.

Follow these steps to add a VLAN to switchport.

Procedure

Step 1 Enter privileged EXEC mode and enter password, if prompted.

Example:

```
Router#enable
```

Step 2 Enter Global Configuration Mode.

Example:

```
Router#configure terminal
```

Step 3 Specify the interface, and enter interface configuration mode.

Example:

```
Router(config)#interface FastEthernet0/0/1
```

Step 4 Select the switchport mode and add the multi-homing VLAN.

Example:

For Access mode:

```
Router(config-if)#switchport mode access
Router(config-if)#switchport access vlan 200
```

For Trunk mode:

```
Router(config-if)#switchport mode trunk
Router(config-if)#switchport trunk allowed vlan 200
```

Step 5 Associate the specified Ethernet segment with the interface. Each Ethernet segment is represented by a unique Ethernet segment ID.

Example:

```
Router(config-if)#evpn ethernet-segment 1
```

Note
Ensure that you configure a unique Ethernet segment ID on any interface. Ensure that you configure the same segment ID on the link that connects the second VTEP (Virtual Tunnel Endpoint) and the dual-homed device (the second link through the Ethernet segment).

Step 6 Exit the interface configuration mode and enter the privileged EXEC mode.

Example:

```
Router(config-if)#end
```

Configure SVI Service Instance

This task configures the SVI service instance.

Procedure

Step 1 Enter the Global Configuration Mode.

Example:

```
Router#configure terminal
```

Step 2 Configure the VLAN interface and enter the interface configuration mode.

Example:

```
Router(config)#interface Vlan200
```

Step 3 Disable the IP processing on the interface.

Example:

```
Router(config-if)#no ip address
```

Step 4 Specify the SVI service instance for the ethernet.

Example:

```
Router(config-if)#service instance 200 ethernet
```

Step 5 Configure the encapsulation type.

Example:

```
Router(config-if-svi-efp)#encapsulation dot1q 200
```

Configure EVPN-Instance

There are two methods to configure evpn-instance:

- Profile based configuration
- Manual configuration

Procedure

Step 1 Enter privileged EXEC mode and enter password, if prompted.

Example:

```
Router#enable
```

Step 2 Enter the Global Configuration Mode.

Example:

```
Router#configure terminal
```

Step 3 For profile method: Configure an EVPN profile instance.

```
Router(config)#l2vpn evpn profile <profile-name> <service-type>
```

Example:

```
Router(config)#l2vpn evpn profile evpn_va vlan-aware
```

Step 4 For manual method: Configure an EVPN instance.

Example:

```
Router(config)#l2vpn evpn instance 1 vlan-aware
```

Step 5 Configures the encapsulation type to VXLAN.

Example:

```
Router(config-evpn-evi)#encapsulation vxlan
```

Apply the Configuration on the Bridge-Domain

Follow these steps to apply the configuration on a bridge-domain.

Procedure

Step 1 Apply the configuration to the specified bridge domain.

Example:

```
Router(config)#bridge-domain 200
```

Step 2 Specify the service instance EFP to the bridge domain.

Example:

```
Router(config-bdomain)#member Vlan200 service-instance 200
```

Step 3 For profile based configuration: Add an EVPN instance to the bridge-domain.

Example:

```
Router(config-bdomain)#member evpn-instance profile evpn_va
```

Step 4 For manual configuration: Add an EVPN instance to the bridge-domain.

Example:

```
Router(config-bdomain)#member evpn-instance 1 vni 20011 ethernet-tag 20011
```

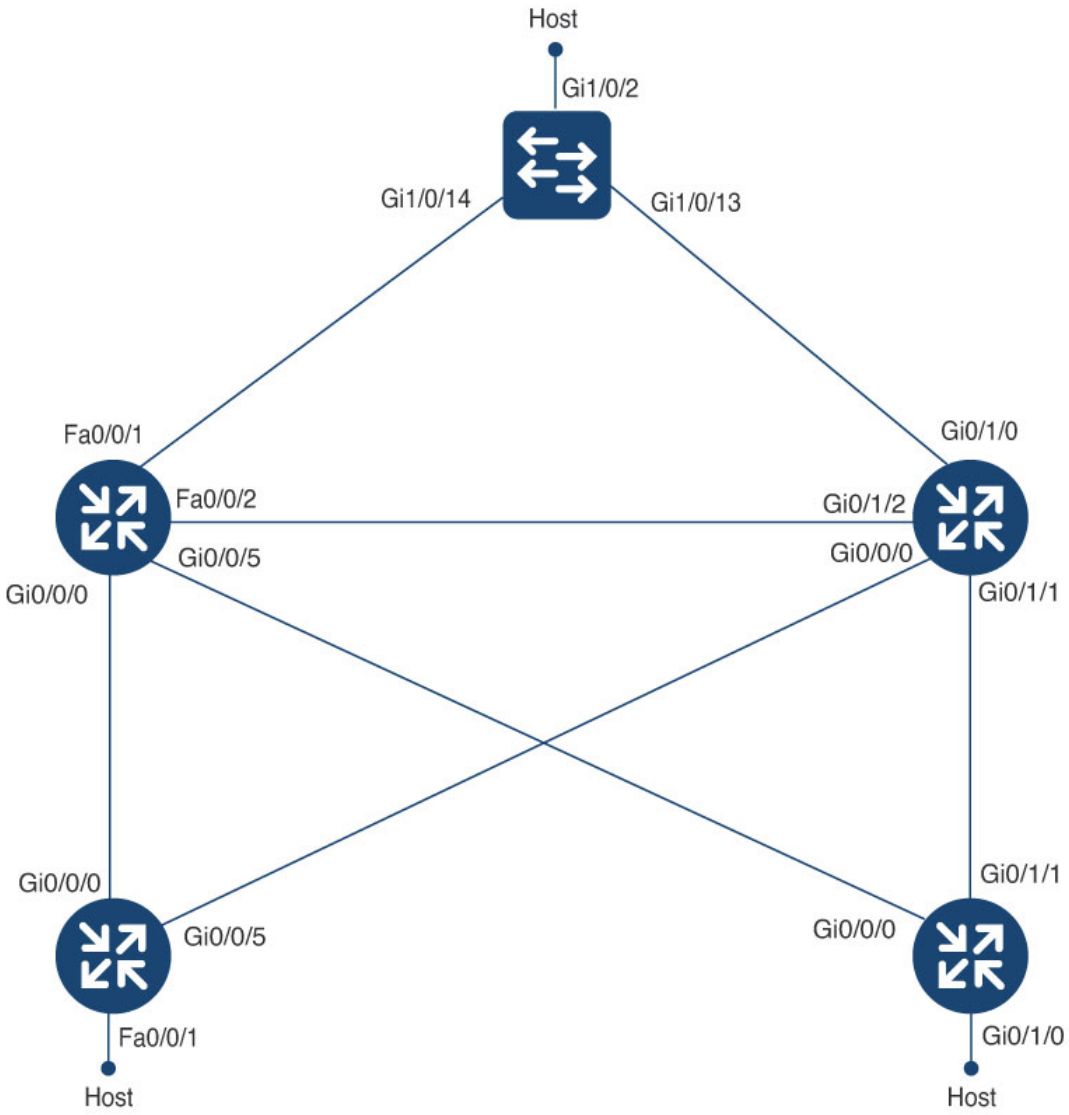
Note

For more information on different replication types, on the **nve**, see Configure Ingress Replication and Configure Static Replication

Configuration Example

Here is an example of configuring the Multi-Homing feature over BGP EVPN VXLAN in a network topology with four routers (R1, R2, R3, and R4) as the VTEPs (Virtual Tunnel Endpoint). The R1 and R2 are the VTEPs with Multi-Homing VLANs as shown in the figure.

Figure 25: Configuration example of 4 router and 1 switch topology



Running Configuration for Router 1 (R1)

```
l2vpn evpn ethernet-segment 1
 identifier type 3 system-mac 0012.0012.0012
 redundancy all-active
 df-election wait-time 1
!
l2vpn evpn
 replication-type ingress
!
l2vpn evpn profile evpn_va vlan-aware
 evi-id 3
 l2vni-base 50000
 ethernet-tag auto-vni
!
bridge-domain 200
 member Vlan200 service-instance 200
 member evpn-instance profile evpn_va
!
interface Loopback0
 ip address 1.1.1.1 255.255.255.255
 ip ospf network point-to-point
 ip ospf 1 area 0
!
interface Loopback1
 ip address 1.1.1.2 255.255.255.255
 ip ospf network point-to-point
 ip ospf 1 area 0
!
interface GigabitEthernet0/0/0
ip address 10.1.3.1 255.255.255.0
 ip ospf network point-to-point
 ip ospf bfd
 ip ospf 1 area 0
 media-type rj45
 bfd interval 50 min_rx 50 multiplier 3
!
interface FastEthernet0/0/1
 switchport trunk allowed vlan 200
 switchport mode trunk
evpn ethernet-segment 1
!
interface FastEthernet0/0/2
 switchport access vlan 12
 switchport mode access
!
interface GigabitEthernet0/0/5
 no switchport
ip address 10.1.4.1 255.255.255.0
ip ospf network point-to-point
 ip ospf bfd
 ip ospf 1 area 0
 bfd interval 50 min_rx 50 multiplier 3
!
interface Vlan12
 ip address 10.1.2.1 255.255.255.0
 ip ospf network point-to-point
 ip ospf bfd
 ip ospf 1 area 0
 bfd interval 999 min_rx 999 multiplier 3
!
interface Vlan200
 no ip address
 service instance 200 ethernet
```

```
  encapsulation dot1q 200
 !
!
interface nve1
 no ip address
 source-interface Loopback1
 host-reachability protocol bgp
 member vni 30000 ingress-replication
!
router ospf 1
 router-id 1.1.1.1
!
router bgp 65001
 bgp log-neighbor-changes
 neighbor 2.2.2.1 remote-as 65001
 neighbor 2.2.2.1 update-source Loopback0
 neighbor 2.2.2.1 fall-over bfd
 neighbor 3.3.3.1 remote-as 65001
 neighbor 3.3.3.1 update-source Loopback0
 neighbor 3.3.3.1 fall-over bfd
 neighbor 3.3.3.1 route-reflector-client
 neighbor 4.4.4.1 remote-as 65001
neighbor 4.4.4.1 update-source Loopback0
 neighbor 4.4.4.1 fall-over bfd
 neighbor 4.4.4.1 route-reflector-client
 !
address-family l2vpn evpn
  neighbor 2.2.2.1 activate
  neighbor 2.2.2.1 send-community both
  neighbor 3.3.3.1 activate
  neighbor 3.3.3.1 send-community both
  neighbor 3.3.3.1 route-reflector-client
  neighbor 4.4.4.1 activate
  neighbor 4.4.4.1 send-community both
  neighbor 4.4.4.1 route-reflector-client
 exit-address-family
!
```

Running Configuration for Router 2 (R2)

```
l2vpn evpn ethernet-segment 1
 identifier type 3 system-mac 0012.0012.0012
 redundancy all-active
 df-election wait-time 1
!
l2vpn evpn
 replication-type ingress
!
l2vpn evpn profile evpn_va vlan-aware
 evi-id 3
 l2vni-base 50000
 ethernet-tag auto-vni
!
bridge-domain 200
 member Vlan200 service-instance 200
 member evpn-instance profile evpn_va
!
interface Loopback0
 ip address 2.2.2.1 255.255.255.255
 ip ospf network point-to-point
 ip ospf 1 area 0
!
interface Loopback1
 ip address 2.2.2.2 255.255.255.255
```

```
 ip ospf network point-to-point
 ip ospf 1 area 0
!
interface GigabitEthernet0/0/0
ip address 10.2.3.2 255.255.255.0
 ip ospf network point-to-point
 ip ospf bfd
 ip ospf 1 area 0
 media-type rj45
 negotiation auto
 bfd interval 50
!
interface GigabitEthernet0/1/0
 switchport trunk allowed vlan 200
 switchport mode trunk
 evpn ethernet-segment 1
!
interface GigabitEthernet0/1/1
switchport access vlan 24
 switchport mode access
!
interface GigabitEthernet0/1/2
 switchport access vlan 12
 switchport mode access
!
interface Vlan12
 ip address 10.1.2.2 255.255.255.0
 ip ospf network point-to-point
 ip ospf bfd
 ip ospf 1 area 0
 bfd interval 999 min_rx 999 multiplier 3
!
interface Vlan24
ip address 10.2.4.2 255.255.255.0
 ip ospf network point-to-point
 ip ospf bfd
 ip ospf 1 area 0
 bfd interval 999 min_rx 999 multiplier 3
!
interface Vlan200
 no ip address
 service instance 200 ethernet
  encapsulation dot1q 200
 !
interface nve1
 no ip address
 source-interface Loopback1
 host-reachability protocol bgp
 member vni 30000 ingress-replication
!
router ospf 1
 router-id 2.2.2.1
!
router bgp 65001
 bgp log-neighbor-changes
 neighbor 1.1.1.1 remote-as 65001
 neighbor 1.1.1.1 update-source Loopback0
 neighbor 1.1.1.1 fall-over bfd
 neighbor 3.3.3.1 remote-as 65001
 neighbor 3.3.3.1 update-source Loopback0
 neighbor 3.3.3.1 fall-over bfd
 neighbor 3.3.3.1 route-reflector-client
 neighbor 4.4.4.1 remote-as 65001
 neighbor 4.4.4.1 update-source Loopback0
```

```
 neighbor 4.4.4.1 fall-over bfd
 neighbor 4.4.4.1 route-reflector-client
 !
 address-family l2vpn evpn
  neighbor 1.1.1.1 activate
  neighbor 1.1.1.1 send-community both
  neighbor 3.3.3.1 activate
  neighbor 3.3.3.1 send-community both
  neighbor 3.3.3.1 route-reflector-client
  neighbor 4.4.4.1 activate
  neighbor 4.4.4.1 send-community both
  neighbor 4.4.4.1 route-reflector-client
exit-address-family
!
```

Running Configuration for Router 3 (R3)

```
l2vpn evpn
 replication-type ingress
!
l2vpn evpn profile evpn_va vlan-aware
 evi-id 3
 l2vni-base 50000
 ethernet-tag auto-vni
!
bridge-domain 200
 member Vlan200 service-instance 200
 member evpn-instance profile evpn_va
!
interface Loopback0
 ip address 3.3.3.1 255.255.255.255
 ip ospf network point-to-point
 ip ospf 1 area 0
!
interface Loopback1
 ip address 3.3.3.2 255.255.255.255
 ip ospf network point-to-point
 ip ospf 1 area 0
!
interface GigabitEthernet0/0/0
ip address 10.1.3.3 255.255.255.0
 ip ospf network point-to-point
 ip ospf bfd
 ip ospf 1 area 0
 media-type rj45
 bfd interval 50 min_rx 50 multiplier 3
!
interface FastEthernet0/0/1
 switchport trunk allowed vlan 200
 switchport mode trunk
!
interface GigabitEthernet0/0/5
 no switchport
ip address 10.2.3.3 255.255.255.0
 ip ospf network point-to-point
 ip ospf bfd
 ip ospf 1 area 0
 bfd interval 50 min_rx 50 multiplier 3
!
interface Vlan200
no ip address
 service instance 200 ethernet
  encapsulation dot1q 200
 !
```

```
!
interface nve1
 no ip address
 source-interface Loopback1
 host-reachability protocol bgp
member vni 30000 ingress-replication
!
router ospf 1
 router-id 3.3.3.1
!
router bgp 65001
 bgp log-neighbor-changes
 neighbor 1.1.1.1 remote-as 65001
 neighbor 1.1.1.1 update-source Loopback0
 neighbor 1.1.1.1 fall-over bfd
 neighbor 2.2.2.1 remote-as 65001
 neighbor 2.2.2.1 update-source Loopback0
 neighbor 2.2.2.1 fall-over bfd
 !
 address-family l2vpn evpn
  neighbor 1.1.1.1 activate
  neighbor 1.1.1.1 send-community both
  neighbor 2.2.2.1 activate
  neighbor 2.2.2.1 send-community both
 exit-address-family
!
```

Running Configuration for Router 4 (R4)

```
l2vpn evpn
 replication-type ingress
!
l2vpn evpn profile evpn_va vlan-aware
 evi-id 3
 l2vni-base 50000
 ethernet-tag auto-vni
!
bridge-domain 200
 member Vlan200 service-instance 200
 member evpn-instance profile evpn_va
!
interface Loopback0
 ip address 4.4.4.1 255.255.255.255
 ip ospf network point-to-point
 ip ospf 1 area 0
!
interface Loopback1
 ip address 4.4.4.2 255.255.255.255
 ip ospf network point-to-point
 ip ospf 1 area 0
!
interface GigabitEthernet0/0/0
 ip address 10.1.4.4 255.255.255.0
 ip ospf network point-to-point
 ip ospf bfd
 ip ospf 1 area 0
 media-type rj45
 negotiation auto
 bfd interval 50 min_rx 50 multiplier 3
!
interface GigabitEthernet0/1/0
 switchport trunk allowed vlan 200
 switchport mode trunk
!
```

```
interface GigabitEthernet0/1/1
 switchport access vlan 24
 switchport mode access
!
interface Vlan24
ip address 10.2.4.4 255.255.255.0
 ip ospf network point-to-point
 ip ospf bfd
 ip ospf 1 area 0
 bfd interval 999 min_rx 999 multiplier 3
!
interface Vlan200
 no ip address
 service instance 200 ethernet
  encapsulation dot1q 200
 !
!
interface nve1
 no ip address
 source-interface Loopback1
 host-reachability protocol bgp
 member vni 30000 ingress-replication
!
router ospf 1
 router-id 4.4.4.1
!
router bgp 65001
 bgp log-neighbor-changes
 neighbor 1.1.1.1 remote-as 65001
 neighbor 1.1.1.1 update-source Loopback0
 neighbor 1.1.1.1 fall-over bfd
 neighbor 2.2.2.1 remote-as 65001
 neighbor 2.2.2.1 update-source Loopback0
 neighbor 2.2.2.1 fall-over bfd
 !
address-family l2vpn evpn
  neighbor 1.1.1.1 activate
  neighbor 1.1.1.1 send-community both
  neighbor 2.2.2.1 activate
  neighbor 2.2.2.1 send-community both
 exit-address-family
!
```

Running Configuration for Switch

```
interface GigabitEthernet1/0/13
 switchport trunk allowed vlan 200
 switchport mode trunk
 carrier-delay 0
 speed 100
 channel-group 1 mode on
end
!
interface GigabitEthernet1/0/14
 switchport access vlan 500
 switchport trunk allowed vlan 200
 switchport mode trunk
 carrier-delay 0
 channel-group 1 mode on
end
!
interface Port-channel1
 switchport trunk allowed vlan 200
 switchport mode trunk
```

```
end
!
```

Verify the Multi-Homing Configuration

Verify the Configuration on Router 1 (R1)

Use the **show l2vpn evpn evi detail** command to verify the configuration of the EVI and bridge-domain, and to ensure the ethernet-tag value is configured

```
Router1#show l2vpn evpn evi detail
EVPN instance:          3 (VLAN Aware)
  Profile:              evpn_va
  RD:                   1.1.1.1:32770 (auto)
  Import-RTs:           65001:3
  Export-RTs:           65001:3
  Per-EVI Label:        none
  State:                Established
  Replication Type:     Ingress (profile)
  Encapsulation:        vxlan (profile)
  IP Local Learn:       Enabled (global)
  Adv. Def. Gateway:    Disabled (global)
  Re-originate RT5:     Disabled (profile)
  AR Flood Suppress:    Enabled (global)
  Bridge Domain:        200
    Ethernet-Tag:       50200
    State:              Established
    Flood Suppress:     Attached
    Core If:
    Access If:
    NVE If:             nve1
    RMAC:               0000.0000.0000
    Core BD:            0
    L2 VNI:             50200
    L3 VNI:             0
    VTEP IP:            1.1.1.2
    Originating Router: 1.1.1.1
    Pseudoports:
      Vlan200 service instance 200 (DF state: forwarding)
        Routes: 1 MAC, 0 MAC/IP
        ESI: 0300.1200.1200.1200.0001
    Peers:
      2.2.2.2
        Routes: 1 MAC, 0 MAC/IP, 1 IMET, 1 EAD
      3.3.3.2
        Routes: 1 MAC, 0 MAC/IP, 1 IMET, 0 EAD
      4.4.4.2
        Routes: 1 MAC, 0 MAC/IP, 1 IMET, 0 EAD
```

Use the **show ip bgp l2vpn evpn all** command to verify the configuration.

```
Router1#show ip bgp l2vpn evpn all
BGP table version is 53, local router ID is 1.1.1.1
Status codes: s suppressed, d damped, h history, * valid, > best, i - internal,
              r RIB-failure, S Stale, m multipath, b backup-path, f RT-Filter,
              x best-external, a additional-path, c RIB-compressed,
              t secondary path, L long-lived-stale,
Origin codes: i - IGP, e - EGP, ? - incomplete
RPKI validation codes: V valid, I invalid, N Not found

     Network          Next Hop            Metric LocPrf Weight Path
Route Distinguisher: 1.1.1.1:32770
```

```
 *>   [1][1.1.1.1:32770][0300120012001200000 1][50200]/23
                      0.0.0.0                                32768 ?
 *mi                   2.2.2.2                   0    100      0 ?
Route Distinguisher: 10.1.2.2:2
 *>i  [1][10.1.2.2:2][0300120012001200000 1][4294967295]/23
                      2.2.2.2                   0    100      0 ?
Route Distinguisher: 10.1.2.2:32770
 *>i  [1][10.1.2.2:32770][0300120012001200000 1][50200]/23
                      2.2.2.2                   0    100      0 ?
Route Distinguisher: 100.109.165.27:8
 *>   [1][100.109.165.27:8][0300120012001200000 1][4294967295]/23
                      0.0.0.0                                32768 ?
Route Distinguisher: 1.1.1.1:32770
 *>   [2][1.1.1.1:32770][50200][48][001101000001][0][*]/20
                      0.0.0.0                                32768 ?
 *>i  [2][1.1.1.1:32770][50200][48][001101000002][0][*]/20
                      2.2.2.2                   0    100      0 ?
 *>i  [2][1.1.1.1:32770][50200][48][001201000001][0][*]/20
                      3.3.3.2                   0    100      0 ?
 *>i  [2][1.1.1.1:32770][50200][48][001301000001][0][*]/20
                      4.4.4.2                   0    100      0 ?
Route Distinguisher: 3.3.3.1:32770
 * i  [2][3.3.3.1:32770][50200][48][001201000001][0][*]/20
                      3.3.3.2                   0    100      0 ?
 *>i                   3.3.3.2                   0    100      0 ?
Route Distinguisher: 10.1.2.2:32770
 *>i  [2][10.1.2.2:32770][50200][48][001101000002][0][*]/20
                      2.2.2.2                   0    100      0 ?
Route Distinguisher: 10.1.4.4:32770
 * i  [2][10.1.4.4:32770][50200][48][001301000001][0][*]/20
                      4.4.4.2                   0    100      0 ?
 *>i                   4.4.4.2                   0    100      0 ?
Route Distinguisher: 1.1.1.1:32770
 *>   [3][1.1.1.1:32770][50200][32][1.1.1.1]/17
                      0.0.0.0                                32768 ?
 *>i  [3][1.1.1.1:32770][50200][32][3.3.3.1]/17
                      3.3.3.2                   0    100      0 ?
 *>i  [3][1.1.1.1:32770][50200][32][10.1.2.2]/17
                      2.2.2.2                   0    100      0 ?
 *>i  [3][1.1.1.1:32770][50200][32][10.1.4.4]/17
                      4.4.4.2                   0    100      0 ?
Route Distinguisher: 3.3.3.1:32770
 * i  [3][3.3.3.1:32770][50200][32][3.3.3.1]/17
                      3.3.3.2                   0    100      0 ?
 *>i                   3.3.3.2                   0    100      0 ?
Route Distinguisher: 10.1.2.2:32770
 *>i  [3][10.1.2.2:32770][50200][32][10.1.2.2]/17
                      2.2.2.2                   0    100      0 ?
Route Distinguisher: 10.1.4.4:32770
 * i  [3][10.1.4.4:32770][50200][32][10.1.4.4]/17
                      4.4.4.2                   0    100      0 ?
 *>i                   4.4.4.2                   0    100      0 ?
Route Distinguisher: 1.1.1.1:1
 *>   [4][1.1.1.1:1][0300120012001200000 1][32][1.1.1.1]/23
                      0.0.0.0                                32768 ?
Route Distinguisher: 2.2.2.1:1
 *>i  [4][2.2.2.1:1][0300120012001200000 1][32][10.1.2.2]/23
                      2.2.2.2                   0    100      0 ?
```

Use the **show nve peers** command to verify the configuration.

```
Router1#show nve peers
'M' - MAC entry download flag  'A' - Adjacency download flag
'4' - IPv4 flag  '6' - IPv6 flag
```

```
Interface  VNI      Type Peer-IP          RMAC/Num_RTs   eVNI     state flags UP time
nve1       50200    L2CP 2.2.2.2          3              50200      UP   N/A  00:19:17
nve1       50200    L2CP 3.3.3.2          2              50200      UP   N/A  00:06:09
nve1       50200    L2CP 4.4.4.2          2              50200      UP   N/A  00:06:08
```

Use the **show l2vpn evpn mac** command to verify the configuration.

```
Router1#show l2vpn evpn mac
MAC Address    EVI   BD    ESI                       Ether Tag  Next Hop(s)
-------------- ----- ----- ------------------------- ---------- ---------------
0011.0100.0001 3     200   0300.1200.1200.1200.0001 50200      Vl200:200
0011.0100.0002 3     200   0300.1200.1200.1200.0001 50200      2.2.2.2
0012.0100.0001 3     200   0000.0000.0000.0000.0000 50200      3.3.3.2
0013.0100.0001 3     200   0000.0000.0000.0000.0000 50200      4.4.4.2
```

Verify the Configuration on Router 2 (R2)

Use the **show l2vpn evpn evi detail** command to verify the configuration of the EVI and bridge-domain, and to ensure the ethernet-tag value is configured

```
Router2#show l2vpn evpn evi detail
EVPN instance:          3 (VLAN Aware)
  Profile:              evpn_va
  RD:                   10.1.2.2:32770 (auto)
  Import-RTs:           65001:3
  Export-RTs:           65001:3
  Per-EVI Label:        none
  State:                Established
  Replication Type:     Ingress (profile)
  Encapsulation:        vxlan (profile)
  IP Local Learn:       Enabled (global)
  Adv. Def. Gateway:    Disabled (global)
  Re-originate RT5:     Disabled (profile)
  AR Flood Suppress:    Enabled (global)
  Bridge Domain:        200
    Ethernet-Tag:       50200
    State:              Established
    Flood Suppress:     Attached
    Core If:
    Access If:
    NVE If:             nve1
    RMAC:               0000.0000.0000
    Core BD:            0
    L2 VNI:             50200
    L3 VNI:             0
    VTEP IP:            2.2.2.2
    Originating Router: 10.1.2.2
    Pseudoports:
      Vlan200 service instance 200 (DF state: PE-to-CE BUM blocked)
        Routes: 1 MAC, 0 MAC/IP
        ESI: 0300.1200.1200.1200.0001
    Peers:
      1.1.1.2
        Routes: 1 MAC, 0 MAC/IP, 1 IMET, 1 EAD
      3.3.3.2
        Routes: 1 MAC, 0 MAC/IP, 1 IMET, 0 EAD
      4.4.4.2
        Routes: 1 MAC, 0 MAC/IP, 1 IMET, 0 EAD
```

Use the **show ip bgp l2vpn evpn all** command to verify the configuration.

```
Router2#show ip bgp l2vpn evpn all
BGP table version is 67, local router ID is 2.2.2.2
Status codes: s suppressed, d damped, h history, * valid, > best, i - internal,
              r RIB-failure, S Stale, m multipath, b backup-path, f RT-Filter,
```

```
             x best-external, a additional-path, c RIB-compressed,
             t secondary path, L long-lived-stale,
Origin codes: i - IGP, e - EGP, ? - incomplete
RPKI validation codes: V valid, I invalid, N Not found

     Network          Next Hop            Metric LocPrf Weight Path
Route Distinguisher: 1.1.1.1:11
 *>i  [1][1.1.1.1:11][0300120012000012000001][4294967295]/23
                      1.1.1.1                  0    100      0 ?
Route Distinguisher: 1.1.1.1:32770
 *>i  [1][1.1.1.1:32770][0300120012000012000001][50200]/23
                      1.1.1.1                  0    100      0 ?
Route Distinguisher: 2.2.2.1:4
 *>   [1][2.2.2.1:4][0300120012000012000001][4294967295]/23
                      0.0.0.0                             32768 ?
Route Distinguisher: 2.2.2.1:32770
 *>   [1][2.2.2.1:32770][0300120012000012000001][50200]/23
                      0.0.0.0                             32768 ?
 *mi                   1.1.1.1                   0    100      0 ?
Route Distinguisher: 1.1.1.1:32770
 *>i  [2][1.1.1.1:32770][50200][48][001101000001][0][*]/20
                      1.1.1.1                  0    100      0 ?
 *>i  [2][1.1.1.1:32770][50200][48][001101000002][0][*]/20
                      1.1.1.1                  0    100      0 ?
Route Distinguisher: 2.2.2.1:32770
 *>i  [2][2.2.2.1:32770][50200][48][001101000001][0][*]/20
                      1.1.1.1                  0    100      0 ?
 *>i  [2][2.2.2.1:32770][50200][48][001101000002][0][*]/20
                      1.1.1.1                  0    100      0 ?
 *>i  [2][2.2.2.1:32770][50200][48][001201000001][0][*]/20
                      3.3.3.1                  0    100      0 ?
 *>i  [2][2.2.2.1:32770][50200][48][001301000001][0][*]/20
                      4.4.4.1                  0    100      0 ?
Route Distinguisher: 3.3.3.1:32770
 *>i  [2][3.3.3.1:32770][50200][48][001201000001][0][*]/20
                      3.3.3.1                  0    100      0 ?
Route Distinguisher: 4.4.4.1:32770
 *>i  [2][4.4.4.1:32770][50200][48][001301000001][0][*]/20
                      4.4.4.1                  0    100      0 ?
Route Distinguisher: 1.1.1.1:32770
 *>i  [3][1.1.1.1:32770][50200][32][1.1.1.1]/17
                      1.1.1.1                  0    100      0 ?
Route Distinguisher: 2.2.2.1:32770
 *>i  [3][2.2.2.1:32770][50200][32][1.1.1.1]/17
                      1.1.1.1                  0    100      0 ?
 *>   [3][2.2.2.1:32770][50200][32][2.2.2.1]/17
                      0.0.0.0                             32768 ?
 *>i  [3][2.2.2.1:32770][50200][32][3.3.3.1]/17
                      3.3.3.1                  0    100      0 ?
 *>i  [3][2.2.2.1:32770][50200][32][4.4.4.1]/17
                      4.4.4.1                  0    100      0 ?
Route Distinguisher: 3.3.3.1:32770
 *>i  [3][3.3.3.1:32770][50200][32][3.3.3.1]/17
                      3.3.3.1                  0    100      0 ?
Route Distinguisher: 4.4.4.1:32770
 *>i  [3][4.4.4.1:32770][50200][32][4.4.4.1]/17
                      4.4.4.1                  0    100      0 ?
Route Distinguisher: 1.1.1.1:1
 *>i  [4][1.1.1.1:1][0300120012000012000001][32][1.1.1.1]/23
                      1.1.1.1                  0    100      0 ?
Route Distinguisher: 2.2.2.2:1
 *>   [4][2.2.2.2:1][0300120012000012000001][32][2.2.2.1]/23
                      0.0.0.0                             32768 ?
```

Use the **show nve peers** command to verify the configuration.

```
Router2#show nve peers
'M' - MAC entry download flag  'A' - Adjacency download flag
'4' - IPv4 flag  '6' - IPv6 flag

Interface  VNI      Type Peer-IP          RMAC/Num_RTs   eVNI     state flags UP time
nve1       50200    L2CP 1.1.1.1          4              50200      UP   N/A  00:02:42
nve1       50200    L2CP 3.3.3.1          2              50200      UP   N/A  00:02:28
nve1       50200    L2CP 4.4.4.1          2              50200      UP   N/A  00:02:37
```

Use the **show l2vpn evpn mac** command to verify the configuration.

```
Router2#show l2vpn evpn mac
MAC Address    EVI   BD    ESI                       Ether Tag  Next Hop(s)
-------------- ----- ----- ------------------------- ---------- ---------------
0011.0100.0001 3     200   0300.1200.1200.1200.0001 50200      1.1.1.1
0011.0100.0002 3     200   0300.1200.1200.1200.0001 50200      1.1.1.1
0012.0100.0001 3     200   0000.0000.0000.0000.0000 50200      3.3.3.1
0013.0100.0001 3     200   0000.0000.0000.0000.0000 50200      4.4.4.1
```

Verify the Configuration on Router 3 (R3)

Use the **show l2vpn evpn evi detail** command to verify the configuration of the EVI and bridge-domain, and to ensure the ethernet-tag value is configured

```
Router3#show l2vpn evpn evi detail
EVPN instance:          3 (VLAN Aware)
  Profile:              evpn_va
  RD:                   3.3.3.1:32770 (auto)
  Import-RTs:           1:3
  Export-RTs:           1:3
  Per-EVI Label:        none
  State:                Established
  Replication Type:     Ingress (profile)
  Encapsulation:        vxlan (profile)
  IP Local Learn:       Enabled (global)
  Adv. Def. Gateway:    Disabled (global)
  Re-originate RT5:     Disabled (profile)
  AR Flood Suppress:    Enabled (global)
  Bridge Domain:        200
    Ethernet-Tag:       50200
    State:              Established
    Flood Suppress:     Attached
    Core If:
    Access If:
    NVE If:             nve1
    RMAC:               0000.0000.0000
    Core BD:            0
    L2 VNI:             50200
    L3 VNI:             0
    VTEP IP:            3.3.3.1
    Originating Router: 3.3.3.1
    Pseudoports:
      Vlan200 service instance 200
        Routes: 1 MAC, 0 MAC/IP
    Peers:
      1.1.1.1
        Routes: 2 MAC, 0 MAC/IP, 1 IMET, 1 EAD
      2.2.2.1
        Routes: 0 MAC, 0 MAC/IP, 1 IMET, 1 EAD
```

Use the **show ip bgp l2vpn evpn all** command to verify the configuration.

```
Router3#show ip bgp l2vpn evpn all
BGP table version is 62, local router ID is 3.3.3.3
Status codes: s suppressed, d damped, h history, * valid, > best, i - internal,
              r RIB-failure, S Stale, m multipath, b backup-path, f RT-Filter,
              x best-external, a additional-path, c RIB-compressed,
              t secondary path, L long-lived-stale,
Origin codes: i - IGP, e - EGP, ? - incomplete
RPKI validation codes: V valid, I invalid, N Not found

     Network          Next Hop            Metric LocPrf Weight Path
Route Distinguisher: 1.1.1.1:11
 * i  [1][1.1.1.1:11][0300120012000001][4294967295]/23
                      1.1.1.1                  0    100      0 ?
 *>i                   1.1.1.1                  0    100      0 ?
Route Distinguisher: 1.1.1.1:32770
 * i  [1][1.1.1.1:32770][0300120012000001][50200]/23
                      1.1.1.1                  0    100      0 ?
 *>i                   1.1.1.1                  0    100      0 ?
Route Distinguisher: 2.2.2.1:4
 * i  [1][2.2.2.1:4][0300120012000001][4294967295]/23
                      2.2.2.1                  0    100      0 ?
 *>i                   2.2.2.1                  0    100      0 ?
Route Distinguisher: 2.2.2.1:32770
 * i  [1][2.2.2.1:32770][0300120012000001][50200]/23
                      2.2.2.1                  0    100      0 ?
 *>i                   2.2.2.1                  0    100      0 ?
Route Distinguisher: 3.3.3.1:32770
 *mi  [1][3.3.3.1:32770][0300120012000001][50200]/23
                      2.2.2.1                  0    100      0 ?
 *>i                   1.1.1.1                  0    100      0 ?
Route Distinguisher: 1.1.1.1:32770
 * i  [2][1.1.1.1:32770][50200][48][001101000001][0][*]/20
                      1.1.1.1                  0    100      0 ?
 *>i                   1.1.1.1                  0    100      0 ?
 * i  [2][1.1.1.1:32770][50200][48][001101000002][0][*]/20
                      1.1.1.1                  0    100      0 ?
 *>i                   1.1.1.1                  0    100      0 ?
Route Distinguisher: 3.3.3.1:32770
 *>i  [2][3.3.3.1:32770][50200][48][001101000001][0][*]/20
                      1.1.1.1                  0    100      0 ?
 *>i  [2][3.3.3.1:32770][50200][48][001101000002][0][*]/20
                      1.1.1.1                  0    100      0 ?
 *>   [2][3.3.3.1:32770][50200][48][001201000001][0][*]/20
                      0.0.0.0                           32768 ?
Route Distinguisher: 1.1.1.1:32770
 * i  [3][1.1.1.1:32770][50200][32][1.1.1.1]/17
                      1.1.1.1                  0    100      0 ?
 *>i                   1.1.1.1                  0    100      0 ?
Route Distinguisher: 2.2.2.1:32770
 * i  [3][2.2.2.1:32770][50200][32][2.2.2.1]/17
                      2.2.2.1                  0    100      0 ?
 *>i                   2.2.2.1                  0    100      0 ?
Route Distinguisher: 3.3.3.1:32770
 *>i  [3][3.3.3.1:32770][50200][32][1.1.1.1]/17
                      1.1.1.1                  0    100      0 ?
 *>i  [3][3.3.3.1:32770][50200][32][2.2.2.1]/17
                      2.2.2.1                  0    100      0 ?
 *>   [3][3.3.3.1:32770][50200][32][3.3.3.1]/17
                      0.0.0.0                           32768 ?
```

Use the **show nve peers** command to verify the configuration.

```
Router3#show nve peers
'M' - MAC entry download flag  'A' - Adjacency download flag
'4' - IPv4 flag  '6' - IPv6 flag
```

```
Interface  VNI      Type Peer-IP          RMAC/Num_RTs   eVNI     state flags UP time
nve1       50200    L2CP 1.1.1.1          4              50200      UP   N/A  00:03:10
nve1       50200    L2CP 2.2.2.1          2              50200      UP   N/A  00:02:24
```

Use the **show l2vpn evpn mac** command to verify the configuration.

```
Router3#show l2vpn evpn mac
MAC Address    EVI   BD    ESI                       Ether Tag  Next Hop(s)
-------------- ----- ----- ------------------------- ---------- ---------------
0011.0100.0001 3     200   0300.1200.1200.1200.0001 50200      1.1.1.1
0011.0100.0002 3     200   0300.1200.1200.1200.0001 50200      1.1.1.1
0012.0100.0001 3     200   0000.0000.0000.0000.0000 50200      Vl200:200
```

Verify the Configuration on Router 4 (R4)

Use the **show l2vpn evpn evi detail** command to verify the configuration of the EVI and bridge-domain, and to ensure the ethernet-tag value is configured

```
Router4#show l2vpn evpn evi detail
EVPN instance:          3 (VLAN Aware)
  Profile:              evpn_va
  RD:                   4.4.4.1:32770 (auto)
  Import-RTs:           1:3
  Export-RTs:           1:3
  Per-EVI Label:        none
  State:                Established
  Replication Type:     Ingress (profile)
  Encapsulation:        vxlan (profile)
  IP Local Learn:       Enabled (global)
  Adv. Def. Gateway:    Disabled (global)
  Re-originate RT5:     Disabled (profile)
  AR Flood Suppress:    Enabled (global)
  Bridge Domain:        200
    Ethernet-Tag:       50200
    State:              Established
    Flood Suppress:     Attached
    Core If:
    Access If:
    NVE If:             nve1
    RMAC:               0000.0000.0000
    Core BD:            0
    L2 VNI:             50200
    L3 VNI:             0
    VTEP IP:            4.4.4.1
    Originating Router: 4.4.4.1
    Pseudoports:
      Vlan200 service instance 200
        Routes: 1 MAC, 0 MAC/IP
    Peers:
      1.1.1.1
        Routes: 2 MAC, 0 MAC/IP, 1 IMET, 1 EAD
      2.2.2.1
        Routes: 0 MAC, 0 MAC/IP, 1 IMET, 1 EAD
```

Use the **show ip bgp l2vpn evpn all** command to verify the configuration.

```
Router4#show ip bgp l2vpn evpn all
BGP table version is 25, local router ID is 4.4.4.4
Status codes: s suppressed, d damped, h history, * valid, > best, i - internal,
              r RIB-failure, S Stale, m multipath, b backup-path, f RT-Filter,
              x best-external, a additional-path, c RIB-compressed,
              t secondary path, L long-lived-stale,
Origin codes: i - IGP, e - EGP, ? - incomplete
RPKI validation codes: V valid, I invalid, N Not found
```

```
     Network          Next Hop            Metric LocPrf Weight Path
Route Distinguisher: 1.1.1.1:11
 * i  [1][1.1.1.1:11][0300120012000012000001][4294967295]/23
                      1.1.1.1                  0    100      0 ?
 *>i                   1.1.1.1                  0    100      0 ?
Route Distinguisher: 1.1.1.1:32770
 * i  [1][1.1.1.1:32770][0300120012000012000001][50200]/23
                      1.1.1.1                  0    100      0 ?
 *>i                   1.1.1.1                  0    100      0 ?
Route Distinguisher: 2.2.2.1:4
 * i  [1][2.2.2.1:4][0300120012000012000001][4294967295]/23
                      2.2.2.1                  0    100      0 ?
 *>i                   2.2.2.1                  0    100      0 ?
Route Distinguisher: 2.2.2.1:32770
 * i  [1][2.2.2.1:32770][0300120012000012000001][50200]/23
                      2.2.2.1                  0    100      0 ?
 *>i                   2.2.2.1                  0    100      0 ?
Route Distinguisher: 4.4.4.1:32770
 *mi  [1][4.4.4.1:32770][0300120012000012000001][50200]/23
                      2.2.2.1                  0    100      0 ?
 *>i                   1.1.1.1                  0    100      0 ?
Route Distinguisher: 1.1.1.1:32770
 * i  [2][1.1.1.1:32770][50200][48][001101000001][0][*]/20
                      1.1.1.1                  0    100      0 ?
 *>i                   1.1.1.1                  0    100      0 ?
 * i  [2][1.1.1.1:32770][50200][48][001101000002][0][*]/20
                      1.1.1.1                  0    100      0 ?
 *>i                   1.1.1.1                  0    100      0 ?
Route Distinguisher: 4.4.4.1:32770
 *>i  [2][4.4.4.1:32770][50200][48][001101000001][0][*]/20
                      1.1.1.1                  0    100      0 ?
 *>i  [2][4.4.4.1:32770][50200][48][001101000002][0][*]/20
                      1.1.1.1                  0    100      0 ?
 *>   [2][4.4.4.1:32770][50200][48][001301000001][0][*]/20
                      0.0.0.0                               32768 ?
Route Distinguisher: 1.1.1.1:32770
 * i  [3][1.1.1.1:32770][50200][32][1.1.1.1]/17
                      1.1.1.1                  0    100      0 ?
 *>i                   1.1.1.1                  0    100      0 ?
Route Distinguisher: 2.2.2.1:32770
 * i  [3][2.2.2.1:32770][50200][32][2.2.2.1]/17
                      2.2.2.1                  0    100      0 ?
 *>i                   2.2.2.1                  0    100      0 ?
Route Distinguisher: 4.4.4.1:32770
 *>i  [3][4.4.4.1:32770][50200][32][1.1.1.1]/17
                      1.1.1.1                  0    100      0 ?
 *>i  [3][4.4.4.1:32770][50200][32][2.2.2.1]/17
                      2.2.2.1                  0    100      0 ?
 *>   [3][4.4.4.1:32770][50200][32][4.4.4.1]/17
                      0.0.0.0                               32768 ?
```

Use the **show nve peers** command to verify the configuration.

```
Router4#show nve peers
'M' - MAC entry download flag  'A' - Adjacency download flag
'4' - IPv4 flag  '6' - IPv6 flag

Interface  VNI      Type Peer-IP          RMAC/Num_RTs   eVNI     state flags UP time
nve1       50200    L2CP 1.1.1.1          4              50200      UP   N/A  00:04:52
nve1       50200    L2CP 2.2.2.1          2              50200      UP   N/A  00:03:33
```

Use the **show l2vpn evpn mac** command to verify the configuration.

```
Router4#show l2vpn evpn mac
MAC Address    EVI   BD    ESI                       Ether Tag  Next Hop(s)
-------------- ----- ----- ------------------------- ---------- ---------------
0011.0100.0001 3     200   0300.1200.1200.1200.0001 50200      1.1.1.1
0011.0100.0002 3     200   0300.1200.1200.1200.0001 50200      1.1.1.1
0013.0100.0001 3     200   0000.0000.0000.0000.0000 50200      Vl200:200
```

Misconfiguration of Multihoming VLANs

If a second switchport is added to a multihoming VLAN, traffic flow can be disrupted, especially for Broadcast, Unknown Unicast, and Multicast (BUM) traffic. This can lead to unpredictable internal states within the EVPN Ethernet Segment, ultimately requiring correction for proper functionality.

Misconfigurations can occur by:

- adding a single-homed switchport to a multihoming VLAN.
- adding a multi-homed switchport to a multihoming VLAN.

In such cases, an error message is logged, indicating the violation of the multihoming VLAN prerequisites.

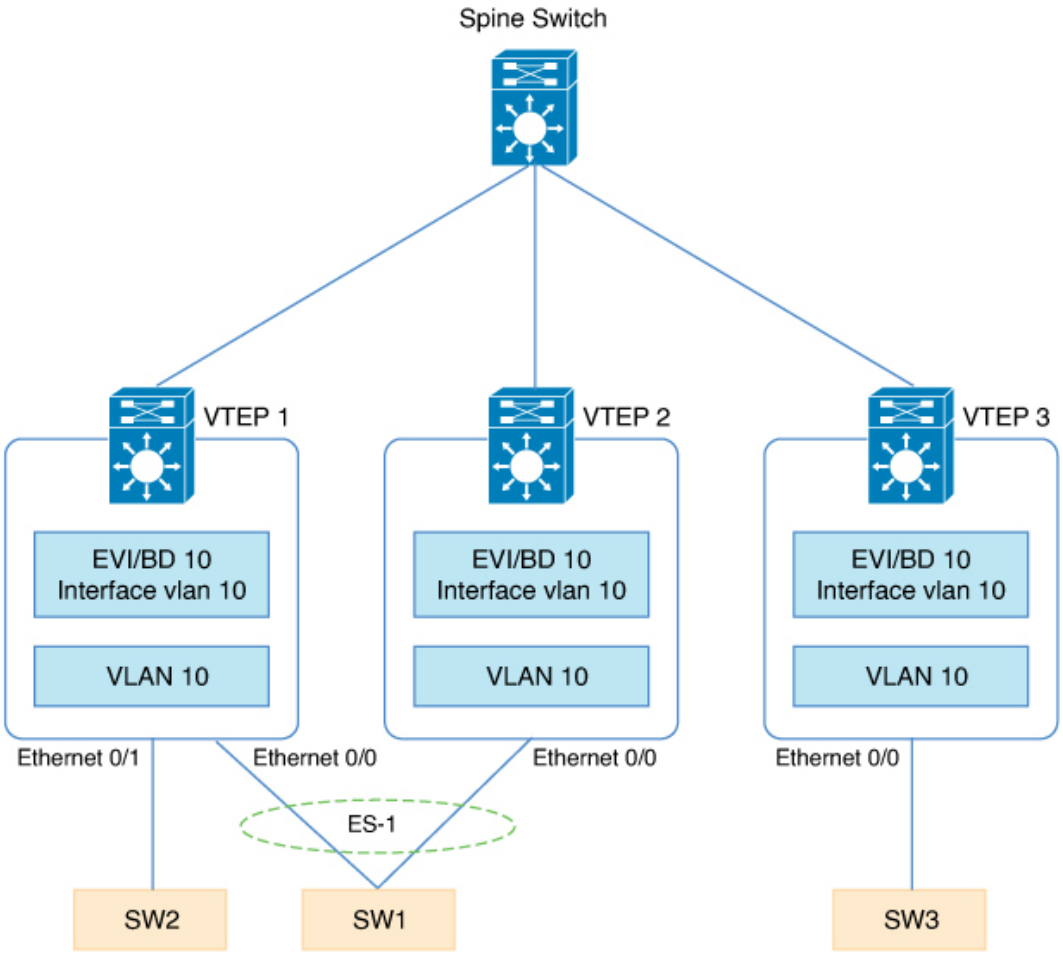
Note Ensure that only one switch port is configured per multihoming VLAN to prevent traffic loss.

Adding Another Switchport to Multihoming VLAN

Configuring a MH VLAN 10, then set up EVPN ES-1 on Ethernet0/0, and then add a new switch port, Ethernet0/1, to VLAN 10.

The following image illustrates the first secenario for misconfiguration.

Figure 26: Add Single-Homed Switchport to MH VLAN



The traffic disruption is illustrated below:

Unicast traffic

- SW1 to SW2:

  SW1 to VTEP1 to SW2: Local switching on VTEP1 VLAN 10

  SW1 to VTEP2 to VTEP1 to SW2: Forwarded by EVPN EVI/BD 10

- SW2 to SW1:

  SW1 to VTEP1 to SW2: Local switching on VTEP1 VLAN 10

- SW1 to SW3:

  Forwarded through EVPN BD 10

- SW3 to SW1:

  Forwarded through EVPN BD 10

- SW3 to SW2:

Forwarded through EVPN BD 10

BUM traffic

- SW1 BUM: Traffic Loss

  Split horizon is performed on SVI EFP member, SW2 may not receive the BUM traffic, if SW1's traffic is hashed to VTEP2.

- SW2 BUM: No Impact

  Split horizon is performed on SVI EFP member, but SW1 will receive the local switched traffic

- SW3 BUM: **Traffic Loss**

  DF election is performed on SVI EFP member, SW2 may not receive the BUM traffic, if VTEP1 is act as non-DF.

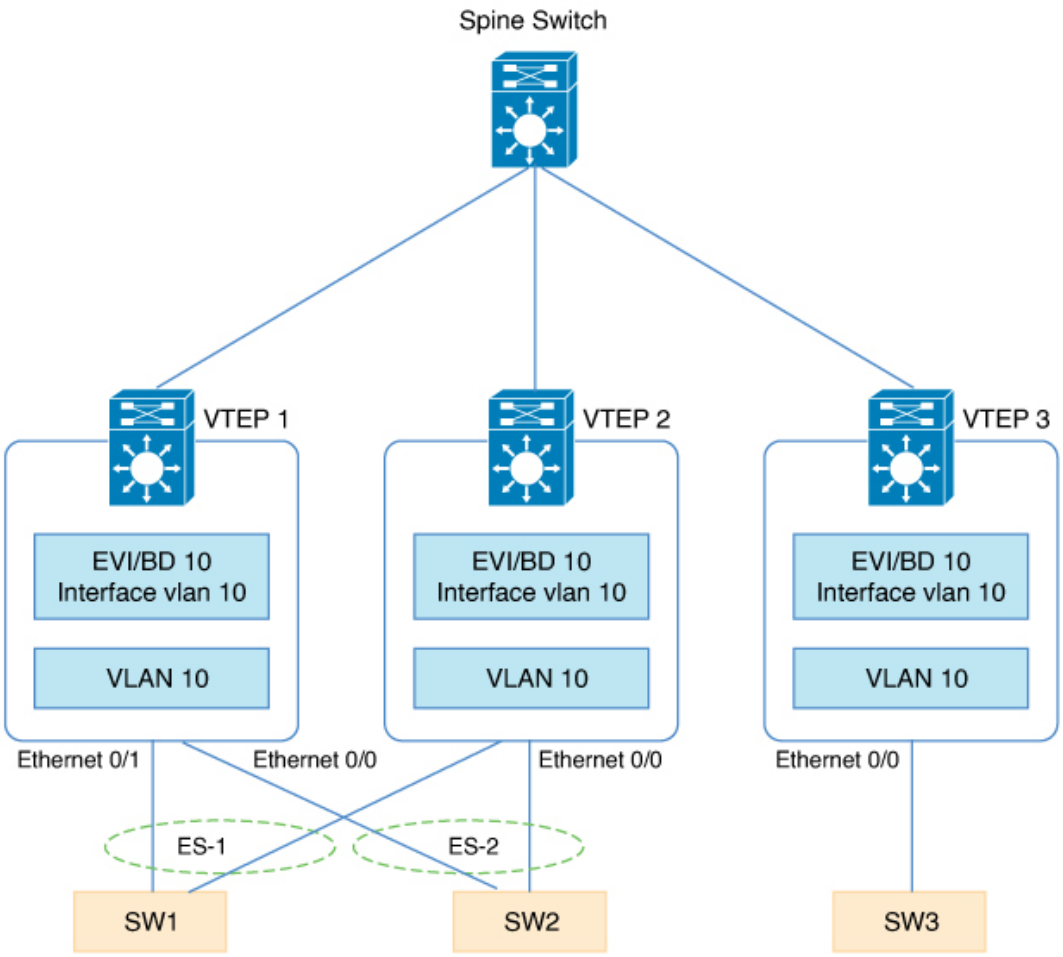
If a Port-channel is used for All-Active multihoming and if it is connected to a multihoming VLAN on a non-designated forwarder (DF) VTEP, the BUM) traffic to a single-homed client switch may be lost.

Adding Another ES Switchport to Multihoming VLAN

Configure a MH VLAN 10, then set up EVPN ES-1 on Ethernet0/0 and EVPN ES-2 on Ethernet0/1. Additionally, add the new allowed VLAN on Ethernet0/1.

The following image illustrates the second secenario for misconfiguration.

Figure 27: Add Another ES Switchport to MH VLAN



The traffic disruption is illustrated below:

Unicast traffic

- SW1 to SW2:

  Local switching on VTEP1/VTEP2 VLAN 10

- SW2 to SW1:

  Local switching on VTEP1/VTEP2 VLAN 10

- SW1 to SW3:

  Forwarded through EVPN BD 10

- SW3 to SW1:

  Forwarded through EVPN BD 10

BUM traffic

- SW1 BUM:

  SW2 receives local switched traffic from VTEP1/VTEP2.

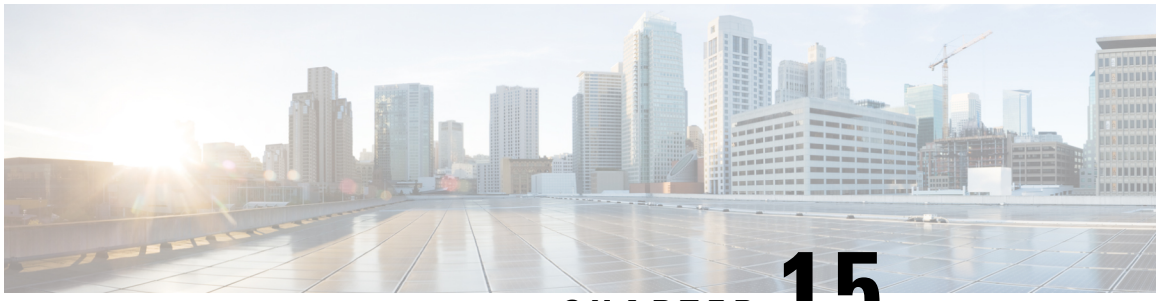
  SW3 receives the BUM traffic forward from EVPN BD 10

- SW3 BUM:

  SW1 receives BUM traffic forward from EVPN BD 10 on DF VTEP1 only.

  SW2 receives the BUM traffic forward from EVPN BD 10 on DF VTEP1 only.

Although most traffic may continue to operate if the multihoming VLAN is misconfigured, its performance will not meet expectations. The internal state of the EVPN Ethernet segment becomes unpredictable, rendering it unsupported. This issue must be resolved to ensure proper functionality.



CHAPTER 15

IP Device Tracking

- IP Device Tracking on routers, on page 195
- Guidelines and limitations, on page 196
- Options to enable SISF-based device tracking, on page 196
- Configuration and verification examples, on page 199

IP Device Tracking on routers

IP Device Tracking (IPDT) is a feature used in Cisco routers and switches to monitor IPv4 devices connected on LAN ports by associating their MAC and IP addresses.

Key functionalities:

- Sends unicast Address Resolution Protocol (ARP) probes at a default interval of 30 seconds to maintain the tracking table
- Supports integration with security features like IP ARP Inspection and DHCP Snooping.Extracts device identity (MAC and IP address) from network traffic.
- Tracks the presence, location, and movement of end-nodes in the network.
- Stores this information in a binding table for use by other security features.

Table 13: Feature History Table

Feature name	Release information	Feature description
IP Device Tracking on routers	Release 17.18.1a	Enables IP Device Tracking with SISF support on LAN ports for IPv4 device monitoring and enhanced security integration.

Switch Integrated Security Features (SISF) based device tracking

The Switch Integrated Security Features (SISF)-based device tracking feature actively monitors the presence, location, and movement of end-nodes within the network as part of the First-Hop Security (FHS) suite. The feature enables the router to snoop incoming traffic, extract device identities (MAC and IP addresses), and store them in a binding table. Many features, including IEEE 802.1X, web authentication, Cisco TrustSec, and LISP, rely on the accuracy of this information to function correctly.

For more information on SISF, see the Chapter: Configuring Switch Integrated Security Features in the Security Configuration Guide

Guidelines and limitations

Many features, including Cisco TrustSec, IEEE 802.1X, LISP, and web authentication, rely on accurate IPDT database information for proper operation.

- LAN port limitation: SISF-based device tracking is configured on LAN ports only. VLAN targets are not supported
- The IPDT feature supports only IPv4 hosts; IPv6 hosts are not supported.
- The PDT feature supports up to 1,000 entries in the IPDT database.
- DHCPv4 host entries appear as ARP-based entries in the IPDT database. If DHCP snooping is enabled on the router, DHCPv4 host entries entries are tagged with a DHCP label in the database.
- A pre-configured default policy exists on the router. You only need to enable it on the interface target to activate basic IPDT functionality.
- For advanced configurations, you may create and apply custom IPDT policies on interfaces.

Guidelines and limitations for programmatically enabling SISF-based device tracking

- Different device tracking clients can exist simultaneously. Multiple programmatic policies may be created to accommodate these clients.
- The settings of each policy depend on the specific device tracking client that generates it.
- Policies are system-defined and may have different attributes.
- Policy attributes are configurable within the device tracking configuration mode (**config-device-tracking**).

Options to enable SISF-based device tracking

SISF-based device tracking is disabled by default and can be enabled either manually or programmatically.

- Manual enablement: Create a custom policy with configurable settings and attach it to a specific interface target.
- Programmatic enablement: Apply the default device tracking policy to an interface target.

Create and attach a custom device tracking policy

To enable and configure a custom device tracking policy for managing device tracking attributes and attaching the policy to a specific interface target.

Procedure

Step 1 Enter the global configuration mode.

Example:

```
Router#configure terminal
```

Step 2 Use the **device-tracking policy <policy_name>** command to create a custom device tracking policy.

Example:

```
Router(config)#device-tracking policy my_policy
```

Step 3 (Optional) Configure policy attributes.

- **data-glean**: Enables binding recovery by data traffic source address gleaning.
- **destination-glean**: Enables binding recovery by data traffic destination address gleaning.
- **prefix-glean**: Glean prefixes in RA and DHCP-PD traffic.
- **default**: Set a command to its default.
- **device-role**: Sets the role of the device attached to the port.
- **limit**: Specifies a limit.
- **medium-type-wireless**: Forces medium type to wireless.
- **no**: Negates a command or sets its default.
- **protocol** : Sets the protocol to glean (default all).
- **security-level**: Specifies the security level.
- **tracking**: Overrides default tracking behavior.
- **trusted-port**: Sets up a trusted port.

Step 4 Exit device-tracking configuration mode.

Example:

```
Router(config-device-tracking)#end
```

Step 5 Use the **interface interface-id** command to specify an interface.

Example:

```
Router(config)#interface gigabitethernet 1/1/4
```

Step 6 Use the **device-tracking attach-policy <policy_name>** command to attach the device tracking policy.

Example:

```
Router(config-if)#device-tracking attach-policy example_policy
```

SISF-based device-tracking policies can only be disabled if they are custom policies.

Step 7 Exit configuration mode.

Example:

```
Router(config-if)#end
```

Apply the default device tracking policy to an interface target

This task enables SISF-based device tracking programmatically by applying the default device tracking policy to an interface target.

Before you begin

Confirm the interface target to which the policy will be applied.

Procedure

Step 1 Enter the global configuration mode.

Example:

```
Rouetr#configure terminal
```

Step 2 Use the **interface type number** command to specify an interface target.

Example:

```
Device(config)#interface gigabitethernet 1/1/4
```

Step 3 Use the **device-tracking** command to enable SISF-based device tracking.

Example:

```
Router(config-if)#device-tracking
```

Step 4 Exit device-tracking configuration mode.

Example:

```
Router(config-device-tracking)#end
```

Step 5 Use the **interface interface-id** command to specify an interface.

Example:

```
Router(config)#interface gigabitethernet 1/1/4
```

Step 6 Use the **device-tracking attach-policy attach-policy** command to attach the device tracking policy.

Example:

```
Router(config-if)#device-tracking attach-policy attach-policy
```

Step 7 Exit configuration mode.

Example:

```
Router(config-if)#end
```

Configuration and verification examples

Configuration examples

Displays the device-tracking configuration:

```
Router#configure terminal
Router(config)#device-tracking policy my_policy
Router(config-device-tracking)#tracking enable
```

Displays the attaching of device-tracking policy:

```
Router#configure terminal
Router(config)#int gi0/1/2
Router(config-if)#device-tracking attach-policy my_policy
Router(config-if)#end
```

Verification examples

Displays the device-tracking policies:

```
Router#show device-tracking policies
Target               Type  Policy               Feature        Target range
Gi0/1/2              PORT  my_policy            Device-tracking vlan all
```

Displays all the configured attributes of the chosen device-tracking policy:

```
Router#show device-tracking policy my_policy

Device-tracking policy my_policy configuration:
  security-level guard
  device-role node
  gleaning from Neighbor Discovery
  gleaning from DHCP6
  gleaning from ARP
  gleaning from DHCP4
  NOT gleaning from protocol unkn
  tracking enable
Policy my_policy is applied on the following targets:
Target               Type  Policy               Feature        Target range
Gi0/1/2              PORT  my_policy            Device-tracking vlan all
```

Displays the database and counters of the device-tracking policy:

```
Router#show device-tracking database

Binding Table has 1 entries, 1 dynamic (limit 100000)
Codes: L - Local, S - Static, ND - Neighbor Discovery, ARP - Address Resolution Protocol,
DH4 - IPv4 DHCP, DH6 - IPv6 DHCP, PKT - Other Packet, API - API created
Preflevel flags (prlvl):
0001:MAC and LLA match     0002:Orig trunk           0004:Orig access
0008:Orig trusted trunk    0010:Orig trusted access  0020:DHCP assigned
0040:Cga authenticated     0080:Cert authenticated   0100:Statically assigned
    Network Layer Address                  Link Layer Address     Interface  vlan
prlvl      age        state      Time left
ARP 20.1.1.20                              bc4a.56ff.e786         Gi0/1/2    20
0005       111s       REACHABLE  196 s

Router#show device-tracking counters int gi0/1/2

Received messages on Gi0/1/2:
Protocol        Protocol message
```

```
NDP
DHCPv6
ARP             REQ[1] REP[2]
DHCPv4
ACD&DAD

Received Broadcast/Multicast messages on Gi0/1/2:
Protocol        Protocol message
NDP
DHCPv6
ARP             REQ[1] REP[2]
DHCPv4

Bridged messages from Gi0/1/2:
Protocol        Protocol message
NDP
DHCPv6
ARP             REP[1]
DHCPv4
ACD&DAD

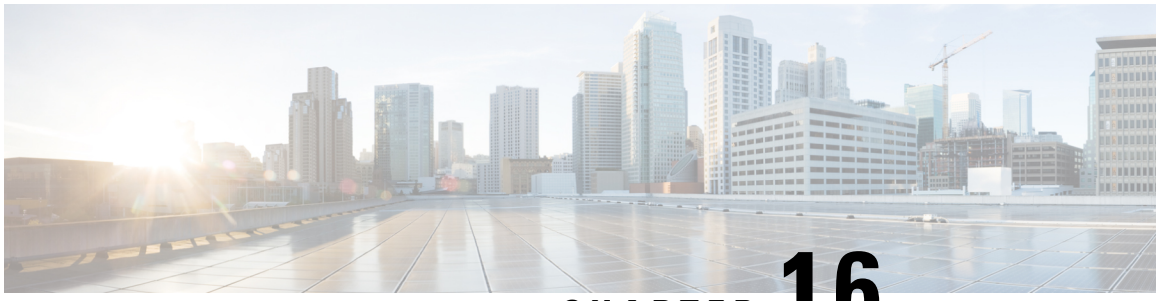
Broadcast/Multicast converted to unicast messages from Gi0/1/2:
Protocol        Protocol message
NDP
DHCPv6
ARP
DHCPv4
ACD&DAD

Probe message on Gi0/1/2:
Type            Protocol message
PROBE_SEND
PROBE_REPLY

Limited Broadcast to Local message on Gi0/1/2:
Type            Protocol message
NDP
DHCPv6
ARP
DHCPv4

Dropped messages on Gi0/1/2:
Feature             Protocol Msg [Total dropped]

Faults on Gi0/1/2:
```



CHAPTER 16

Power Over Ethernet (PoE)

This section contains the following:

- Power over Ethernet Overview, on page 201
- Device Detection and Power Allocation, on page 201
- Command Line Interface, on page 201

Power over Ethernet Overview

Power over Ethernet (PoE) is typically used to power up devices such as Access points, IP Cameras and IP Phones connected to the device's Ethernet ports. The total PoE available power is 30W to be shared by the 4 LAN ports.

The power allocation is as follows:

- 1 x POE+ (AT Type2, Class 4) port (25.5 W)
- 1 x POE (AT Type1 or AF-Class 0/3) ports (15.4 W)
- 4 x POE (AF, Class 1, Class 2) Ports (3.84W or 6.49W)

Device Detection and Power Allocation

The router will detect a Cisco Pre-standard or an IEEE-compliant PD when the PoE is enabled and the connected device is not being powered by an AC adapter.

After device detection, the router will determine the power requirements based on power classification class. Depending on the available power in the power budget, the router determines if a port can be powered. The router initially allocates this power when it detects and powers the device. Power negotiation using CDP/LLDP protocols happens thereafter. Supported protocols for power negotiation are CDP for Cisco PD, and LLDP for non-Cisco PDs. Maximum power budget for 4 LAN ports combined at any time is 30.8W. On reload the PoE ports are powered down until the unit reboots.

Command Line Interface

This section describes the CLI to use for configuring and displaying PoE.

To configure auto or off:

```
power inline auto | never
```

Configuration example:

```
Router#config terminal
Router#interface g0/1/<0,1,2,3>
Router(config-if)#power inline {auto|never}
```

To Verify your configuration:

```
Router#show power inline
Available:30.0(w)  Used:22.5(w)  Remaining:7.5(w)

Interface Admin  Oper       Power   Device              Class Max
                            (Watts)
--------- ------ ---------- ------- ------------------ ----- ----
Gi0/1/0   auto   on         15.4    Ieee PD            4     30.0
Gi0/1/1   auto   off        0.0     n/a                n/a   30.0
Gi0/1/2   auto   off        0.0     n/a                n/a   30.0
Gi0/1/3   auto   on         7.1     IP Phone 8845      2     30.0
Router#
```

To show power on a particular interface:

```
Router#show power inline {interface-id}
```

Displays PoE status for a router for the specified interface.

```
show power inline interface-id detail
```

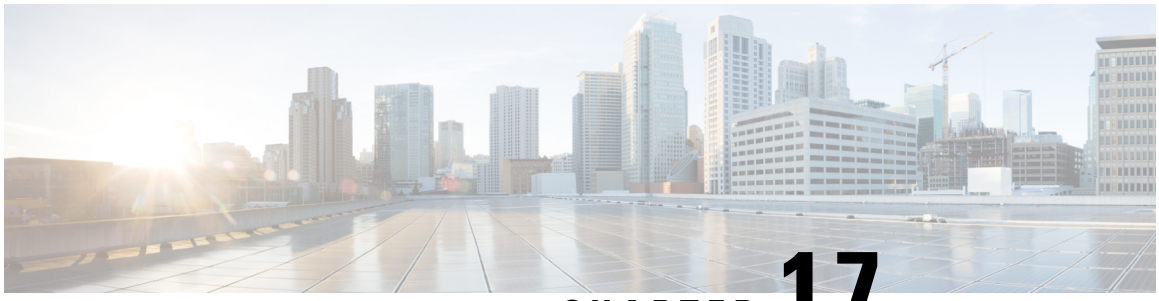
To show power consumption:

```
Router#show power
Main PSU :
    Total Power Consumed: 20.99 Watts
    Configured Mode : N/A
    Current runtime state same : N/A
    PowerSupplySource : External PS
POE Module :
    Configured Mode : N/A
    Current runtime state same : N/A
    Total power available : 30 Watts
Router#
```

The list of commands for debugging PoE follows:

Command	Description
Debug ilpower controller	Display PoE controller debug messages
Debug ilpower event	Display PoE event debug messages
Debug ilpower port	Display PoE port manager debug messages
Debug ilpower powerman	Display PoE power management debug messages
Debug ilpower cdp	Display PoE CDP debug messages
Debug ilpower registries	Display PoE registries debug messages

Command	Description
Debug ilpower scp	Display PoE scp debug messages



CHAPTER 17

vCPU and RAM Distribution

- Introduction, on page 205
- Distribution of vCPU and RAM Resources for Cisco IOx Applications, on page 205
- Higher CPU and RAM Allocation for IOx Applications , on page 206
- Configure Data Plane Heavy Template, on page 206
- Verify the Active vCPU and RAM Distribution, on page 207
- Configure Service Plane Heavy Template, on page 207
- Verify the Active vCPU and RAM Distribution, on page 208

Introduction

This chapter provides information on how to distribute Virtual Central Processing Unit (vCPU) cores and RAM resources for Cisco IOx applications on Cisco Catalyst IR1835 router.

Note vCPU is also known as physical processor.

Distribution of vCPU and RAM Resources for Cisco IOx Applications

Distributing the available resources efficiently allows you to run multiple IOx applications simultaneously.

Use these templates to distribute the vCPU and RAM resources:

- **Data Plane Heavy—**Refers to a router configuration where majority of system resources are dedicated to the data plane, which is responsible for processing and forwarding network packets.

  Data Plane Heavy template maximizes throughput and ensures high-speed packet transfer, which is essential for network traffic demands. This ensures more processing power and memory to handle the increased load on the data plane, enhancing router's ability to move large volumes of data efficiently.

Note The Data Plane Heavy is the default template for vCPU and RAM distribution in the IR1835 router.

- **Service Plane Heavy—**Refers to a router configuration where majority of system resources are allocated to the service plane, which is responsible for providing network services such as Quality of Service (QoS), security functions, and load balancing.

  Service Plane Heavy template allocates additional vCPU and RAM to IOx applications. However, it reduces data throughput (bandwidth).

Note Routers with 2 GB RAM and a single core vCPU (IOx resources) cannot run multiple IOx applications such as Unified Threat Defense and Cisco Cyber Vision.

Higher CPU and RAM Allocation for IOx Applications

From Cisco IOS XE Release 17.15.1, the IR1835 router with 8 GB RAM supports Data Plane Heavy and Service Plane Heavy distribution templates. You can allocate 3 GB RAM and two vCPU cores to IR1835 router for hosting Cisco IOx applications. We recommend the Service Plane Heavy template to allocate resources for hosting IOx applications.

Configure Data Plane Heavy Template

Procedure

Step 1 Enter the configuration command to enable the data plane heavy template:

```
Router(config)#platform resource data-plane-heavy
```

Step 2 Enter the reload command to reboot the router and activate the data plane heavy template:

```
Router#reload
```

What to do next

Verify the active vCPU and RAM distribution.

Verify the Active vCPU and RAM Distribution

Use the **show** command to verify the vCPU cores allocation for IOx applications.

```
Router#show platform software cpu allocation
CPU alloc information:
Control plane cpu alloc: 0-1
Data plane cpu alloc: 2-3
Service plane cpu alloc: 0-1
Slow control plane cpu alloc:
Template used: CLI-data_plane_heavy
```

Use the **show** command to verify the RAM allocation for IOx applications.

```
Router#show app-host resource
Resource Allocation:
  CPU Quota: 33%
  Memory Quota: 2048MB
  Storage Total: 6350MB
  Storage Available: 1404MB
```

Use the **show** command to verify the CPU units resource allocation for IOx applications.

```
Router#show app-host infra
IOX version: 2.11.0.3
App signature verification: disabled
CAF Health: Stable
Internal working directory: /vol/harddisk/iox
CPU:
  Quota: 33%
  Available: 33%
  Quota: 1617(Units)
  Available: 1617(Units)
```

Configure Service Plane Heavy Template

Procedure

Step 1 Enter the configuration command to enable the service plane heavy template:

```
Router(config)#platform resource service-plane-heavy
```

Step 2 Enter the reload command to reboot the router and activate the service plane heavy template:

```
Router#reload
```

What to do next

Verify the active vCPU and RAM distribution.

Verify the Active vCPU and RAM Distribution

Use the **show** command to verify the vCPU cores allocation for IOx applications.

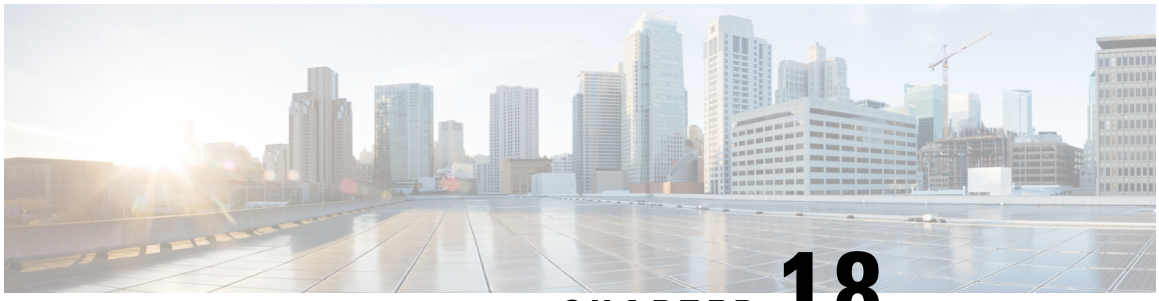
```
Router#show platform software cpu allocation
CPU Allocation Information:
  Control plane cpu alloc: 0-1
  Data plane cpu alloc: 3
  Service plane cpu alloc: 0-2
Template used: CLI-service_plane_heavy
```

Use the **show** command to verify the RAM allocation for IOx applications.

```
Router#show app-host resource
Resource Allocation:
  CPU Quota: 38%
  Memory Quota: 3072MB
  Storage Total: 6350MB
  Storage Available: 1403MB
```

Use the **show** command to verify the CPU units resource allocation for IOx applications.

```
Router#show app-host infra
IOX version: 2.11.0.3
App signature verification: disabled
CAF Health: Stable
Internal working directory: /vol/harddisk/iox
CPU:
  Quota: 38%
  Available: 38%
  Quota: 1862(Units)
  Available: 1862(Units)
```



CHAPTER 18

Cellular pluggable interface modules

The Cisco 4G LTE-Advanced Configuration chapter has been replaced by a new standalone guide called Cellular Pluggable Interface Module Configuration Guide . This guide contains updated information on all aspects of using the Cisco Cellular PIM.

Important The Pluggable Module is not hot swappable. The router must be reloaded after a new module is installed.

- Support for P-5GS6-R16SA-GL pluggable module, on page 209
- Galileo support on the LTE pluggable modules, on page 209
- Radio signal parameters and GPS coordinates of cellular telemetry monitoring using syslog messages, on page 211
- GPS NMEA unique identifier, on page 212
- Configure dead reckoning GPS NMEA unique identifier, on page 214

Support for P-5GS6-R16SA-GL pluggable module

From Cisco IOS-XE Release 17.18.2, the P-5GS6-R16SA-GL Pluggable Module is supported on ESR6300 routers.

Support for the P-5GS6-R16SA-GL Pluggable Module works the same way on the ESR6300 routers as it does on the other IoT routers. For details, see 5G Sub-6 GHz Pluggable Interface Module and Cellular Pluggable Interface Module Configuration Guide .

Galileo support on the LTE pluggable modules

The Cisco IOS XE 17.11.1a release introduces support for the Galileo constellation on LTE pluggable modules.

GPS was the only GNSS constellation supported in earlier releases. The Cisco IOS XE 17.11.1a release introduces support for Galileo constellation.

Note You can enable only one constellation at a time.

Configuration command examples:

```
config# controller cellular<slot/port>
(config-controller)# <no> lte gps constellation <gps | galileo | gnss >
```

Example:

```
(config-controller)# lte gps constellation ?
galileo  select Galileo as active constellation
gps      select GPS as active constellation
gnss      select multiple GNSS as active constellation
```

Note The default setting is GPS mode.

The galileo and gnss options in this CLI configure Galileo and simultaneous GNSS modes, such as GPS and Galileo.

If you disable the GPS configuration, make sure that no constellation is configured. This maintains consistency with GPS mode configuration.

Example:

```
config#


controller Cellular 0/4/0


(config-controller)# no lte gps constellation gps
```

Show commands example:

Note The example shows the current GNSS constellation as Galileo.

Example:

```
config#


show cellular 0/4/0 gps detail


GPS Feature =  enabled
GPS Mode Configured =  standalone
Current Constellation Configured =  galileo
GPS Port Selected =  Dedicated GPS port
GPS Status =  GPS acquiring
```

Any changes made to the configuration will require the router to be rebooted.

More information is available in the Cellular Pluggable Interface Module Configuration Guide .

Radio signal parameters and GPS coordinates of cellular telemetry monitoring using syslog messages

Cellular telemetry is a feature that enables real-time monitoring and analysis of cellular connection performance by collecting Radio Frequency (RF) parameters and Global Positioning System (GPS) coordinates from the cellular network at a fixed interval of 60 seconds.

- Monitors key radio signal parameters such as RSSI, RSRP, RSRQ, PCI, and SNR.
- Collects GPS coordinates for location tracking.
- Facilitates troubleshooting, network optimization, and ensures reliable connectivity.

Cellular Telemetry Reference Information

Cellular telemetry parameters that can be monitored include:

- Received Signal Strength Indicator (RSSI)
- Reference Signal Received Power (RSRP)
- Reference Signal Received Quality (RSRQ)
- Physical Cell Identity (PCI)
- Signal to Noise Ratio (SNR)
- Global Positioning System (GPS) coordinates

The Cisco Catalyst WAN Manager does not support cellular telemetry.

Enable cellular telemetry

To enable the cellular telemetry feature in the contoller cellular interface 0/x/0 using the CLI.

Before you begin

- Insert the cellular Pluggable Interface Module (PIM) into the IR device.
- Enable GPS in the controller for GPS coordinates to be displayed in syslogs.

Procedure

Step 1 Enter the global configuration mode.

Example:

```
Router#configure terminal
```

Step 2 Enter the cellular configuration mode.

Example:

```
Router (config)#controller cellular 0/4/0
```

Step 3 Enable the RF parameters and GPS coordinates.

Example:

```
Router (config-controller)#lte modem serviceability signal-parameters
Router (config-controller)#end
```

Disable cellular telemetry

To disable the cellular telemetry feature, use the no form of the **lte modem serviceability signal-parameters** command, as shown in the example:

```
Router#Configure terminal


Router(config)#controller cellular 0/4/0

Router(config-controller)#no lte modem serviceability signal-parameters
Router(config-controller)#end
```

Monitor cellular telemetry

You can monitor RF parameters and GPS coordinates for cellular interfaces by using the **show logging** command or check the console output.

The following example displays RF parameters and GPS coordinates at every one minute interval:

```
Router#show logging
*Sep  3 17:08:42.081: %CELLWAN-2-MODEM_SIGNAL_PARAM: Cellular0/4/0: 4G: RSSI = -54 dBm RSRP
 = -76 dBm RSRQ = -9 dB PCI = 1 SNR = 27.4 dB
Latitude =  12 Deg 56 Min 8.9260 Sec North
Longitude =  77 Deg 41 Min 44.1641 Sec East

*Sep  3 17:09:42.080: %CELLWAN-2-MODEM_SIGNAL_PARAM: Cellular0/4/0: 4G: RSSI = -54 dBm RSRP
 = -76 dBm RSRQ = -9 dB PCI = 1 SNR = 29.0 dB
Latitude =  12 Deg 56 Min 8.8989 Sec North
Longitude =  77 Deg 41 Min 44.1570 Sec East
```

The following example displays only RF parameters and not GPS coordinates since GPS is not enabled in the controller:

```
Router#show logging
*Sep  3 17:08:42.081: %CELLWAN-2-MODEM_SIGNAL_PARAM: Cellular0/4/0: 4G: RSSI = -54 dBm RSRP
 = -76 dBm RSRQ = -9 dB PCI = 1 SNR = 27.4 dB

*Sep  3 17:09:42.080: %CELLWAN-2-MODEM_SIGNAL_PARAM: Cellular0/4/0: 4G: RSSI = -54 dBm RSRP
 = -76 dBm RSRQ = -9 dB PCI = 1 SNR = 29.0 dB
```

GPS NMEA unique identifier

Starting with Cisco IOS-XE Release 26.1.1, Cisco routers can send a Unique Identifier (UID) to identify the source device when streaming NMEA GPS sentences over UDP on the Cisco IR1101 and Cisco IR1835

routers. UID streaming is supported from the Dead Reckoning (DR) module and from cellular modems that support GNSS. The UID uniquely identifies each router in GPS data processing systems.

Table 14: Feature History Table

Feature name	Release information	Feature description
Send a UID to identify the unique source when sending GPS coordinates data	Release 26.1.1	This feature allows you to configure Cisco routers to send a Unique Identifier (UID) while streaming NMEA GPS sentences over UDP. This UID streaming is supported by both the Dead Reckoning (DR) module and cellular modems with GNSS capabilities, enabling each router to be uniquely identified in GPS data processing systems.

Cisco proprietary NMEA sentence format

When you configure a UID, a Cisco proprietary NMEA sentence is sent alongside the standard NMEA sentence in the same UDP packet. The format is:

```
$PCSCU,<UID>,<Source>,<Router Time>,<NMEA Sentence Time>,<NMEA Sentence Checksum>,,,,,,*95
```

- `CSC` indicates Cisco
- `U` indicates this sentence provides a unique identifier
- `<UID>` is the serial number, hostname, or custom ID
- `<Source>` indicates the origin of the NMEA stream:
  - `0`: cellular modem0
  - `1`: cellular modem1
  - `2`: DR module

For example, for cellular modem1, if you use the serial number as the UID, the Cisco proprietary NMEA sentence appears as:

```
$PCSCU,FCW2447P0EU,1,213431.100,213431.000,75,,,,,,*95
```

In this NMEA sentence

- `CSC` indicates Cisco
- `U` indicates that this sentence provides a unique identifier
- `FCW2447P0EU` indicates serial number as the UID
- `1` indicates source as cellular modem1
- `1213431.100` indicates the router time
- `213431.000` indicates the NMEA sentence time

- `75` indicates the NMEA sentence checksum

NMEA sentence selection

By default, all NMEA sentences from the modem or DR module are sent to the configured UDP destination. To reduce UDP traffic, you can select specific NMEA sentence types to send using a hex bitmask. The table presents each NMEA sentence type and its corresponding HEX value.

Table 15: NMEA sentence type and HEX value

HEX value	NMEA sentence type
0x01	GGA
0x02	GSA
0x04	GSV
0x08	RMC
0x10	VTG
0x20	GNS

Configure dead reckoning GPS NMEA unique identifier

Follow these steps to configure dead reckoning GPS NMEA UID.

Procedure

Step 1 Use the **configure terminal** command to enter the configuration mode.

Example:

```
Router# configure terminal
```

Step 2 Use the command **controller gps-dr** command to enter the controller mode.

Example:

```
Router(config)# controller gps-dr
```

Step 3 Use the **dead-reckoning enable** command to enable Dead Reckoning.

Example:

```
Router(config)# dead-reckoning enable
```

Step 4 Use the **dead-reckoning nmea uid** {**custom** *unique-identifier* **| hostname | serial-number**} command to enable the GPS NMEA UID.

UID can be any of these:

- **custom**: user-configured string of upto 50 characters.

- **hostname**: host name of the router.
- **serial-number**: serial number of the router.

When using **serial-number** or **hostname**, the UID is automatically extracted and added to the proprietary NMEA sentence. This allows uniform CLI configuration across routers.

When using a **custom ID**, each router must be configured with a *unique-identifier* value to ensure distinct identification.

Example:

```
Router(config-controller)# dead-reckoning nmea uid custom 142
```

Example:

```
Router(config-controller)# dead-reckoning nmea uid hostname
```

Example:

```
Router(config-controller)# dead-reckoning nmea uid serial-number
```

Step 5 Use the **dead-reckoning nmea** *source-address destination-address destination-port* command to specify the UDP source and destination.

Example:

```
Router(config-controller)# dead-reckoning nmea udp 10.195.79.179 171.70.55.77 14013
```

Step 6 Use the **lte gps nmea filter** *hex-value* command to select the sentence types.

hex-value: HEX bitmask for NMEA message filtering (0x01:GGA, 0x02:GSA, 0x04:GSV, 0x08:RMC, 0x010:VTG, 0x20:GNS)

This example shows the HEX value if you want to send sentence type as only RMC (0x08) and GNS (0x20):

Example:

```
Router(config-controller)# dead-reckoning nmea filter 28
```

This example shows the HEX value if you want to send sentence type as only VTG (0x10) and GSV (0x04):

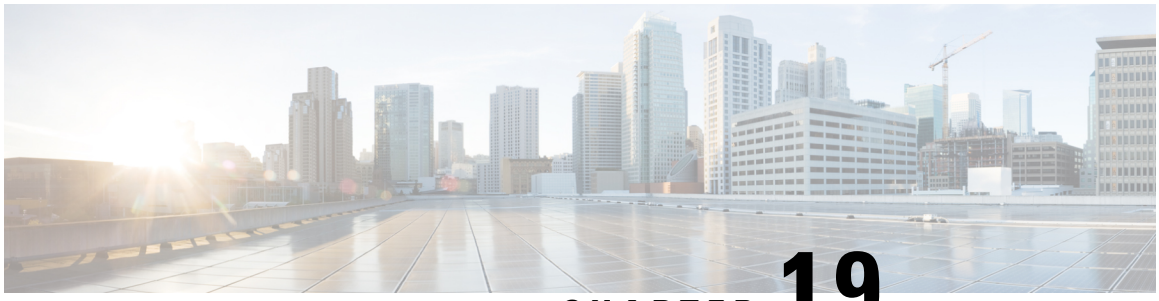
Example:

```
Router(config-controller)# dead-reckoning nmea filter 14
```

Step 7 Use the **end** command to exit the controller configuration mode and return to the privileged EXEC mode.

Example:

```
Router(config-controller)# end
```



CHAPTER 19

Cisco Wi-Fi Interface Module (WIM)

This chapter contains the following sections:

- Cisco Wi-Fi Interface Module (WIM) Overview, on page 217
- Cisco IoT Operations Dashboard (OD) Support to Configure and Manage the WP-WIFI6-x Module, on page 217
- WIM firmware upgrades, on page 217
- List of WIM firmware upgrade commands, on page 218

Cisco Wi-Fi Interface Module (WIM) Overview

This chapter of the configuration guide has been turned into a stand-alone book called Cisco Wi-Fi Interface Module (WIM) Configuration Guide to provide more details on the configuration of the WIM.

Cisco IoT Operations Dashboard (OD) Support to Configure and Manage the WP-WIFI6-x Module

Cisco IOS XE release 17.11.1a provides additional capabilities to the Cisco Wi-Fi Interface Module (WIM).

This section contains the following:

WIM firmware upgrades

A WiFi6 Interface Module (WIM) firmware upgrade is a process that:

- Automates the transfer of firmware images from a router's local flash to the WiFi6 Interface Module (WIM).
- Ensures the WiFi6 module remains consistently up to date through seamless integration with the Cisco SD-WAN controller.
- Simplifies ongoing maintenance for network operators.

You can enable seamless WIM firmware upgrades through the Cisco SD-WAN controller by automating image transfers from the router's local flash. This enhancement streamlines maintenance and ensures the WiFi6 module is always current.

Key enhancements

To facilitate the firmware upgrade process these key enhancements are added:

- VLAN 2371 is now reserved exclusively for WIM management traffic. This VLAN handles the secure transfer of firmware images from the IR1800 to the WIM.
- A new VRF named `wifi6-mgmt-vrf` has been added. The Vlan2371 interface is assigned to this VRF to isolate management traffic from standard data traffic.
- A new Switch Virtual Interface (SVI) for VLAN 2371 is created with a dedicated management IP address within the `192.168.3.1` range to communicate with the module.
- The WIM management IP address is 192.168.3.4
- The interface Wlan-GigabitEthernet0/1/4 is now configured in Trunk Mode. VLAN 2371 is automatically allowed on this trunk to ensure management connectivity.
- An internal TFTP server is automatically enabled on the IR1800, pointing to the firmware image stored in the local flash for the WIM to retrieve.
- You need to add the data traffic vlans to the allowed vlan list in Wlan-GigabitEthernet0/1/4.

List of WIM firmware upgrade commands

These commands are used for upgrading the WIM firmware on the Cisco Catalyst IR1800 Rugged Series Router.

```
vrf definition wifi6-mgmt-vrf

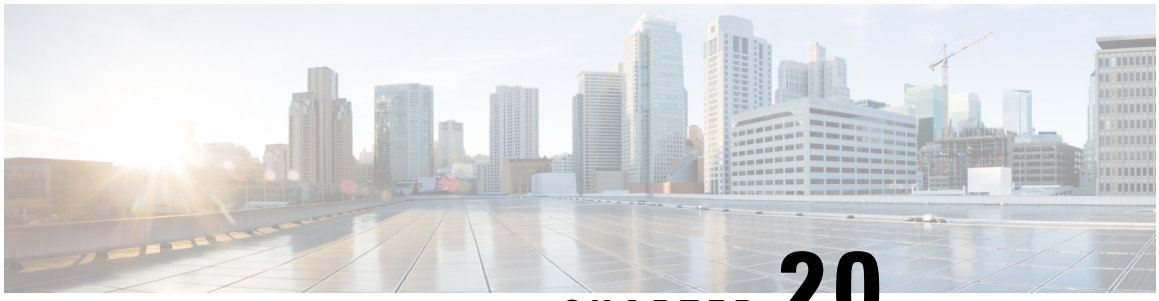
interface Wlan-GigabitEthernet0/1/4
 switchport mode trunk

interface Vlan2371
 vrf forwarding wifi6-mgmt-vrf
 ip address 192.168.3.1 255.255.255.240

tftp-server bootflash:WIM-FW-IMAGES/ap1g8t-wim-image-tar
```

Note

- **vrf definition** *vrf -name* is used for creating VRF. It creates a Virtual Routing and Forwarding (VRF) instance named `wifi6-mgmt-vrf`. This is used to segment network traffic, ensuring that the management traffic for your Wi-Fi 6 environment is isolated from other network traffic. WIM management traffic is forwarded on `Vlan 2371`.
- **interface** *interface-name* is used for assigning an interface to VRF.
- **tftp-server bootflash:** *file-name* is used for configuring the router to act as a TFTP server, pointing to a specific firmware image.



CHAPTER 20

Digital I/O, Ignition, and CAN Bus Connectivity

This section contains the following:

- Overview, on page 219
- Digital IO, on page 220
- Controller Area Network (CAN) Bus, on page 220
- IOx CAN Bus Support, on page 221
- Packet Capture Support for CANBUS, on page 221
- Configuring Digital IO, on page 222
- Ignition Power Management Overview, on page 223
- Features of Ignition Power Management, on page 223
- Ignition Sense Overview, on page 224
- IR1835 Ignition Switch, on page 225
- IR1800 Ignition and Battery Voltage, on page 227
- Command Line Interface (CLI), on page 227
- Default Values, on page 229
- Ignition Power Management Yang Model, on page 229
- Support SNMP MIB for Ignition Power Management , on page 230

Overview

This section covers the Digital I/O configuration and the CAN Bus details.

The IR1835 supports four General-Purpose Input/Output ports (GPIO), also referred to as digital I/O ports. A GPIO port can be configured as input OR output alarm. It can work as dry or wet contact, protected up to +60V .

The Controller Area Network (CAN) Bus enables the ECU (electronic control unit) in a vehicle to communicate with all other ECUs. It consists of two wires, CAN bus High and Low supporting data rate up to 1 Mbs.

The characteristics of the high speed CAN Bus 2.0B are ISO 11898-1 data link layer, ISO 11898-2 and ISO-11898-5 physical layer up to 1Mbs data rate (SW dependent).

Digital IO

A total of four Digital IO with a common return are supported. Digital IO is similar to the ALARM IN and ALARM OUT supported in the IE switches and IR routers. The differences are the ALARM IN is a dedicated input, the ALARM OUT is a dedicated output whereas the Digital IO can be input or output. ALARM OUT includes a relay to provide the Normally Open (NO) or Normally Close (NC) terminals. Digital IO implements a relay feature similar to Alarm port.

The following configuration commands are available:

```
alarm contact attach-to-iox
alarm contact <1-4> enable enable
alarm contact <1-4> application
alarm contact <1-4> description
alarm contact <1-4> severity
alarm contact <1-4> threshold <1600-2700>
alarm contact <1-4> trigger
alarm contact <1-4> output <1 | 0>
alarm contact <1-4> output relay temperature
alarm contact <1-4> output relay input-alarm <0-4>
```

All configuration commands also come with a **no** prefix to them.

Controller Area Network (CAN) Bus

Details on the CAN Bus and connectivity to the vehicle's On-Board Diagnostic (OBD-II) are covered in the Cisco Catalyst IR1800 Rugged Series Router Hardware Installation Guide.

The CAN Bus interface can be viewed using the command line interface. Some of the CLIs are:

```
IR1800#conf term
Enter configuration commands, one per line.  End with CNTL/Z.
IR1800(config)#canbus baudrate ?
  <125000-1000000>  enter baud rate ranging from 125000 to 1000000

IR1800#show platform hardware canbus ?
  interface  Display CAN Bus interface
  link       Display CAN Bus link

IR1800#show platform hardware canbus link
8: can0: <NOARP,UP,LOWER_UP,ECHO> mtu 16 qdisc pfifo_fast state UP mode DEFAULT group default
 qlen 10
    link/can  promiscuity 0
    can state ERROR-ACTIVE restart-ms 100
          bitrate 125000 sample-point 0.875
          tq 500 prop-seg 6 phase-seg1 7 phase-seg2 2 sjw 1
          mcp251x: tseg1 3..16 tseg2 2..8 sjw 1..4 brp 1..64 brp-inc 1
          clock 10000000
          re-started bus-errors arbit-lost error-warn error-pass bus-off
          0          0          0          0          0          0

numtxqueues 1 numrxqueues 1 gso_max_size 65536 gso_max_segs 65535

    RX: bytes  packets  errors  dropped overrun mcast
    0          0        0       0       0       0
    TX: bytes  packets  errors  dropped carrier collsns
    0          0        0       0       0       0
```

```
IR1800#

IR1800#show platform hardware canbus interface
can0      Link encap:UNSPEC  HWaddr 00-00-00-00-00-00-00-00-00-00-00-00-00-00-00-00
          UP RUNNING NOARP  MTU:16  Metric:1
          RX packets:0 errors:0 dropped:0 overruns:0 frame:0
          TX packets:0 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:10
          RX bytes:0 (0.0 B)  TX bytes:0 (0.0 B)
```

IOx CAN Bus Support

A vxcan tunnel is created between Linux and IOx, and then the CAN_GW inside the linux is configured to forward CANBus traffic from/to the real can bus interface (i.e. can0) to/from the end point of the vxcan tunnel.

A vxcan tunnel is established by default, the CANBus traffic will be sent over the tunnel as follows:

CANBus: [Linux] vxcan-ap and vxcan0 [IOx]

Sample output from vxcan0 [IOx]

```
vxcan0    Link encap:UNSPEC  HWaddr 00-00-00-00-00-00-00-00-00-00-00-00-00-00-00-00
          UP RUNNING NOARP  MTU:16  Metric:1
          RX packets:715 errors:0 dropped:0 overruns:0 frame:0
          TX packets:530 errors:0 dropped:0 overruns:0 carrier:0
```

Important Notes

- The feature can support multiple containers, one pair of vxcan will be setup for one container, therefore, two pairs of vxcan will be setup for two containers.
- Within one container, if multiple applications need to access vxcan, it is customer's application responsibility to include CAN_GW in their software.

Packet Capture Support for CANBUS

When enabled, this feature will capture packets sent and received on the IR1800 series CANBUS. Once captured, the data will be exported as a packet capture (PCAP) file to allow for further examination. The feature is configured in exec mode and is only temporary, meaning it is not permanent across a reboot/reload.

A file name is required for the capture. The default location for the capture file is at bootflash:/canbus_dumplogs. If the capture is started without specifying the file initially, or after the router is reloaded, you will get the following message when you check the status:

```
canbus packetdump file pcapfile path bootflash:/canbus_dumplogs/pcapfile didn't start
```

After stopping the capture, if you want to start the capture again without specifying the file name, the old specified name will be overwritten.

Use the following command to specify the name of the capture file:

```
Router#monitor canbus packetdump file <filename>
```

Note You do not need to specify the path, the only supported path is the default path **bootflash:/canbus_dumplogs**

Use the following command to start the capture using the specified <filename> from the command above:

```
Router#monitor canbus packetdump start
```

Use the following command to stop the capture:

```
Router#monitor canbus packetdump stop
```

Use the following command to check the status of the monitoring:

```
Router#show canbus packetdump
```

Command Examples

```
Router#monitor canbus packetdump ?
file CAN Bus interface packet capture destination file
start CAN Bus interface packet capture start
stop CAN Bus interface packet capture stop

Router#monitor canbus packetdump file canbusfile
Router#show canbus packetdump
canbus packetdump file canbusfile path bootflash:/canbus_dumplogs/canbusfile didn't start

Router#monitor canbus packetdump start
Router#show canbus packetdump
canbus packetdump file canbusfile path bootflash:/canbus_dumplogs/canbusfile started

Router#monitor canbus packetdump stop
Router#show canbus packetdump
canbus packetdump file canbusfile path bootflash:/canbus_dumplogs/canbusfile didn't start
```

Configuring Digital IO

To configure the feature, perform the following:

```
Router(config)# alarm contact 2 enable
Router(config)#default alarm contact 2 output
Router(config)#default alarm contact 2 severity
Router(config)#default alarm contact 2 threshold
Router(config)#default alarm contact 2 trigger
Router(config)#default alarm contact 2 application
Router(config)#default alarm contact 2 description
Router(config)#end
```

To view alarm output, perform the following:

```
Router# show alarm | section Digital I/O 2
Digital I/O 2:
Description: External digital I/O port 2
Status: Not Asserted
Application: Dry
Severity: minor
Trigger: Closed
Voltage: 3300mV
Threshold: 1600mV
```

```
Mode: Input
Router#
```

Ignition Power Management Overview

This section provides a description and instructions for configuration of the Ignition Power Management feature of the IR1800 router. Ignition Power Management prevents the router from draining the charge of the battery on automotive applications. It also keeps the router up and running while the vehicle is stopped. Therefore, users do not have to wait for routers to reload each time the vehicle is stopped.

When the engine is running, it generates energy and recharges the battery. When the ignition is turned off, the router can remain operational for a pre-determined period of time.

On the IR1800 series, there are two ways to perform Ignition Power Management. All of the routers in the series can use the software based voltage sense controlled by the MCU. The IR1835 can also use the Signal/Ignition signal, available from the Ignition pin in the GPIO 6 pins connector.

Ignition Wiring information is found in the Hardware Installation Guide.

Features of Ignition Power Management

Ignition Power Management is controlled through the MCU built into the router. The MCU provides auto detection of power input to detect if the ignition is on or off. Ignition on is detected by if the ignition signal is on or off, or by sensing the power input level.

Ignition Power Management Cabling and Connector is covered here: https://www.cisco.com/c/en/us/td/docs/routers/access/IR1800/hig/b-ir1800-hig/m-GPIO.html

The system software tries to prevent the discharge of the battery with the following:

- Turning the router off if the vehicle has the ignition off for a period of time (programmable).
- Turning the router off if the battery voltage drops to a certain level (programmable).
- Attempting to protect the router by turning the router off if the battery voltage rises above a certain level (fixed amount of time).

The system software logs the following events to the system log:

- When the user turns on or off the ignition management feature with CLI
- When the ignition is turned on or off
- When the ignition-off timer expires and the system goes off
- When the user enables or disables the feature through the CLI
- Tentatively logs the under-voltage and over-voltage events

All Ignition On, Off, Low and High input thresholds can be stored in non-volatile memory. When the device boots up, the thresholds will be retrieved from the memory to the register. When the CPU detects the front panel push button, the non-volatile memory will reset to default value.

Ignition Sense Overview

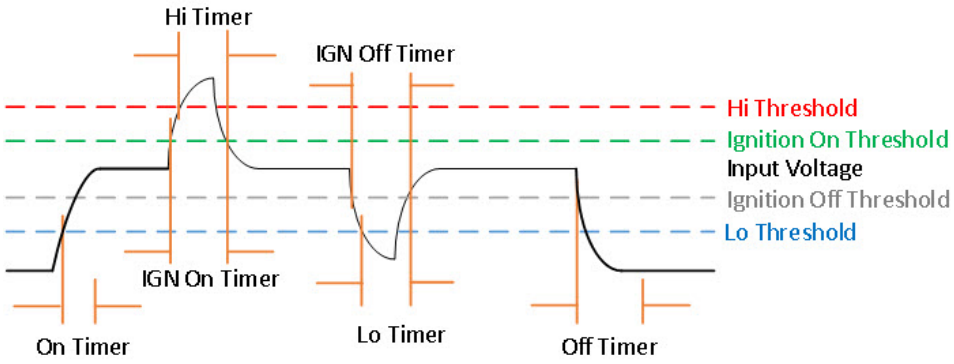
On the IR1800, Ignition Power Management adds ignition sense through software. Ignition sense can be determined by either monitoring the ignition signal pin, or the battery voltage level. The ignition pin and voltage level are continuously monitored. This input will be the main signal to start the state machine. If the ignition signal is not active, then voltage level sense will be used. There is an option for user to disable the voltage level sense by clearing the Ignition Voltage Sense Enable register.

See the following table:

Ignition Bypass	Ignition Voltage Sense Enable	Ignition Sense
0	0	Ignition Signal
0	1	Voltage Level
1	x	Ignition Disabled

The following graphic illustrates Ignition Sense:

Figure 28: Ignition Based on Voltage (Analog Input)



The following tables provide voltage details:

Table 16: Input Voltage (DC)

Minimum	9.6V
Maximum	36V
Nominal	12V or 24V

Table 17: Ignition Sense Voltage

	12V Battery	24V Battery
On	13V + 2%	26V + 2%
Off	13V -2%	26V - 2%

Table 18: Battery Voltage

	12V Battery	24V Battery
Under-Voltage	11.5V	23V
Over-Voltage	36V	36V

See the following command output example:

```
IR1800#show run | s ignition
ignition off-timer 300
ignition undervoltage threshold 9 000
ignition battery-type 12v
ignition sense-voltage threshold 13 000
ignition sense
ignition enable

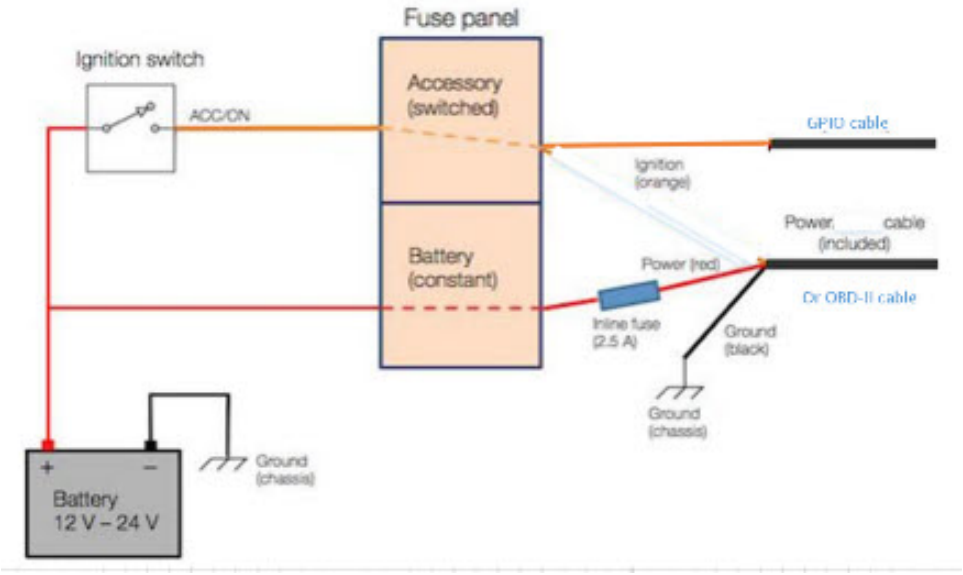
IR1800#show ignition status
Status:
  Ignition management: Enabled
  Input voltage:       11.953 V
  Ignition status:     Timing ignition off shut down
  Ignition Sense:      Enabled
  Shutdown timer:      242.0 s to off [will begin power down at ~100 sec]
  Config-ed battery:   12v
Thresholds:
  Undervoltage:        9.000 V
  Overvoltage:         37.000 V
  Sense on:            13.000 V
  Sense off:           12.800 V
  Undervoltage timer:  20.0 s
  Overvoltage timer:   1.0 s
  Ignition-Off timer:  300.0 s
```

IR1835 Ignition Switch

The IR1835 is the only model in the IR1800 series that offers ignition monitoring via the Signal/Ignition signal, available from the Ignition pin in the GPIO 6 pins connector.

Refer to the following figure:

Figure 29: Ignition based on Signal/Ignition switch(Digital Input)



There are two ways to power the router:

- CANBUS power
- Through the CAB-PWR-15-MF4 cable

There are two thresholds:

- 12V — Defaults ignition sense-voltage threshold 13
- 24V — Defaults ignition sense-voltage threshold 26

See the following command output example:

```
IR1835#show run | s ignition ! Configuring
ignition off-timer 120
ignition undervoltage threshold 9 600
ignition battery-type 12v
ignition sense-voltage threshold 13 000
no ignition sense
ignition enable

IR1835#show ignition Status ! Monitoring
 Ignition management: Enabled
Input voltage:       13.999 V
Ignition status:     Power on
Ignition Sense:      Disabled
 Shutdown timer:      0.0 s to off [will begin power down at ~100 sec]
Config-ed battery:   12v
Thresholds:
Undervoltage:        9.600 V
Overvoltage:         37.000 V
Sense on:            13.200 V
Sense off:           12.800 V
Undervoltage timer:  20.0 s
Overvoltage timer:   1.0 s
Ignition-Off timer:  120.0 s
```

IR1800 Ignition and Battery Voltage

The IR1800 can be set-up with a 12V or 24V battery. The ignition sense voltage level will change accordingly.

- 12Volt Battery
  - If input voltage > 13.200V for more than 1 second, Ignition is **ON**
  - If input voltage < 12.800V for more than 20 seconds, Ignition is **OFF**
- 24Volt Battery
  - If input voltage > 26.200V for more than 1 second, Ignition is **ON**
  - If input voltage < 25.800V for more than 20 seconds, Ignition is **OFF**

Use the folowing command to determine your battery type:

```
IR1800#conf term
Enter configuration commands, one per line.  End with CNTL/Z.
IR1800(config)#ignition battery-type ?
  12v  The battery is 12v
  24v  The battery is 24v
```

Command Line Interface (CLI)

The Ignition Power Management feature of the IR1800 series uses a command line interface.

Enabling Ignition Power Management

The feature is disabled by default, and enabled using the following command:

```
Router(config)#ignition enable
Router(config)#
*Sep 15 16:08:27.697: %IGNITION-5-IGN_ENABLE_CMD: The Ignition Power Management is enabled
Router(config)#end
```

Options to the ignition enable command are:

```
Router(config)#ignition enable ?
  enable        Enable ignition power management feature
  off-timer     Off timer delay
  sense         Enable ignition power sense feature
  undervoltage  Set undervoltage parameters for shutting the system off
```

- Ignition off timer value. After the ignition is turned off the router will stay operational for this amount of time, then it shuts down if the ignition is still off:

```
Router#ignition off-timer <value>
```

- Ignition sense value. Turning this on allows the router to detect if the the ignition is on or off.

```
Router(config)#ignition sense
*Sep 15 16:08:14.391: %IGNITION-5-IGN_SENSE_CMD: The Ignition Voltage Sense is enabled
Router(config)#end
```

- Ignition undervoltage. This command allows you to set parameters for shutting down the router.

```
Router(config)#ignition undervoltage threshold ?
<0-999>  Enter millivolts (mV), if any
```

- Over-voltage threshold. This command allows you to set parameters for shutting down the router.

```
Router#ignition overvoltage threshold <value>
```

Command Examples

Default configuration with no Ignition Management settings:

```
Router#show ignition
Status:
Ignition management: Disabled
Input voltage: 17.672 V
Ignition status: Power on
Ignition Sense: Disabled
Shutdown timer: 0.0 s to off [will begin power down at ~100 sec]
Config-ed battery: 12v
Thresholds:
Undervoltage: 9.000 V
Overvoltage: 37.000 V
Sense on: 13.200 V
Sense off: 12.800 V
Undervoltage timer: 20.0 s
Overvoltage timer: 1.0 s
Ignition-Off timer: 300.0 s
```

Configure Ignition Management:

```
Router(config)#ignition enable
Router(config)#
*Sep 15 16:08:27.697: %IGNITION-5-IGN_ENABLE_CMD: The Ignition Power Management is enabled

Router(config)#ignition sense
*Sep 15 16:08:14.391: %IGNITION-5-IGN_SENSE_CMD: The Ignition Voltage Sense is enabled
Router(config)#end
```

Verify the changes:

```
Router#show ignition
Status:
Ignition management: Enabled
Input voltage: 17.656 V
Ignition status: Power on
Ignition Sense: Enabled
Shutdown timer: 0.0 s to off [will begin power down at ~100 sec]
Config-ed battery: 12v
Thresholds:
Undervoltage: 9.000 V
Overvoltage: 37.000 V
Sense on: 13.200 V
Sense off: 12.800 V
Undervoltage timer: 20.0 s
Overvoltage timer: 1.0 s
Ignition-Off timer: 300.0 s
Router#
```

Show Ignition Status

The following commands are used to show the status of the feature:

```
Router#show ignition
Status:
```

```
Ignition management: Disabled
Input voltage: 17.672 V
Ignition status: Power on
Ignition Sense: Disabled
Shutdown timer: 0.0 s to off [will begin power down at ~100 sec]
Config-ed battery: 12v
Thresholds:
Undervoltage: 9.000 V
Overvoltage: 37.000 V
Sense on: 13.200 V
Sense off: 12.800 V
Undervoltage timer: 20.0 s
Overvoltage timer: 1.0 s
Ignition-Off timer: 300.0 s

Router#show running-config | sec ignition
ignition off-timer 300
ignition undervoltage threshold 9 000
ignition battery-type 12v
ignition sense-voltage threshold 13 000
no ignition sense
no ignition enable
```

Default Values

The following default settings apply to Ignition Power Management:

Setting	Default Value
Ignition Power Management Feature	Disabled
Ignition Sense	Disabled
Off Timer	300 seconds
Under Voltage Threshold	9.000 Volts
Under Voltage Off Timer	20 seconds
Over Voltage Off Timer	1.0 seconds
Ignition Sense On	13.200 volts (26.200 volts)
Ignition Sense Off	12.800 volts (25.800 volts)
Configured Battery	12 volts (24 volts)

Ignition Power Management Yang Model

A Yang Model is available for the Ignition Power Management Configuration Model (config-model) and Ignition Power Management Show Command (oper-model).

The ignition configuration CLI's for the config-model are as follows:

- **[no] ignition enable** – Enable/disable ignition power management.
- **ignition off-timer** <*value*> – After the ignition is turned off, the router will stay operational for this amount of time, then it turns off if the ignition is still off.
- **[no] ignition sense**– Enable/disable voltage sense
- **ignition undervoltage threshold** <*value*> – If the input voltage drops to levels below this threshold, it will cause the router to shut down.

The leaf nodes for this model are as follows:

- enable
- off-timer
- sense
- threshold-value-volt
- threshold-value-milli-volt

A Yang model file, Cisco-IOS-XE-ignition.yang is available for configuration model.

The ignition show CLI's for oper model are as follows:

show ignition – Shows all the ignition-related parameters

A Yang model file, Cisco-IOS-XE-ignition-oper-transform.yang is available for this purpose.

Support SNMP MIB for Ignition Power Management

The following is an example output from the **show ignition** CLI:

```
Status:
Ignition management: Disabled
Input voltage: 17.672 V
Ignition status: Power on
Ignition Sense: Disabled
Shutdown timer: 0.0 s to off [will begin power down at ~100 sec]
Config-ed battery: 12v
Thresholds:
Undervoltage: 9.000 V
Overvoltage: 37.000 V
Sense on: 13.200 V
Sense off: 12.800 V
Undervoltage timer: 20.0 s
Overvoltage timer: 1.0 s
Ignition-Off timer: 300.0 s
```

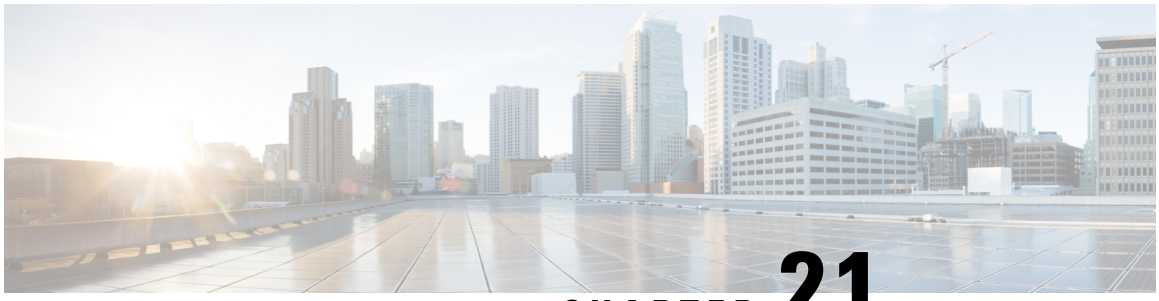
A MIB file, CISCO-IGNITION-MIB.my, is available to support the **show ignition** CLI.

The MIB file has the following fields:

- IgnitionManagement (1=True; 2=False, Boolean)
- InputVoltage (millivolt, Unsigned Integer)
- IgnitionStatus (Bootloader/Power on/Timing low voltage shut down…, State Index)

  - Bootloader (0)
  - Power On (1)
  - Low Delay (2)
  - Off Delay (3)
  - High Delay (4)
  - On Delay (5)
  - Monitor (6)
  - Sleep (7)
  - Unknown (8)

- IgnitionSense (1=True; 2=False, Boolean)
- ShutdownTimer (milliseconds, Unsigned Integer)
- ConfigBattery (volts, Interger)
- Undervoltage (millivolt, Unsigned Integer)
- Overvoltage (millivolt, Unsigned Integer)
- SenseOn (millivolts, Unsigned Integer)
- SenseOff (millivolts, Unsigned Integer)
- UndervoltageTimer (milliseconds, Unsigned Integer)
- OvervoltageTimer (milliseconds, Unsigned Integer)
- IgnitionOffTimer (milliseconds, Unsigned Integer)



CHAPTER 21

Configuring GPS

This chapter contains the following:

- GPS Overview, on page 233
- Cellular Modem-Based GPS, on page 235
- GPS/Dead Reckoning module (IRM-GNSS-ADR), on page 235
- GPS and Dead Reckoning Support for the J1939 Connector, on page 244
- National Marine Electronics Association (NMEA) IOx Support, on page 246
- NMEA UDP Socket Support, on page 246
- NMEA UDP Configuration with Yang, on page 250
- Yang Data Model Support, on page 252
- Example: Connecting to a Server Hosting a GPS Application, on page 253
- GNSS Support on the GPS/Dead Reckoning Module (IRM-GNSS-ADR), on page 254
- Galileo support on the LTE pluggable modules, on page 255
- Access Accelerometer and Gyro Sensor Data from IRM-GNSS, on page 256
- Change in Vendor for GNSS Module, on page 256
- IOX Access to IR1800 On-board Accelerometer and Gyroscope, on page 257
- Configure DR Module using Cisco Catalyst SD-WAN Manager, on page 258
- GPS NMEA unique identifier , on page 258

GPS Overview

There are two ways to receive GPS information. There is Cellular modem GPS available in the LTE modules that support GPS, and there is a dedicated GPS/Dead Reckoning module (IRM-GNSS-ADR) which provides more robust capabilities.

The IR1833 and IR1835 have a slot for a dedicated GPS field-replaceable unit (FRU) module, which will be used in addition to the one integrated in the LTE module, for more accurate dead reckoning performance. The part number is IRM-GNSS-ADR.

IRM-GNSS-ADR hardware is capable of supporting various GNSS constellations. Please inquire with your sales representative for a roadmap of support of additional constellations on the IRM-GNSS-ADR module.

Modem based GPS cannot provide the coordinates when there are no satellites in line of sight. The GPS module with DR capabilities provide the coordinates even when there are no satellites in line of sight.

With the addition of the dedicated GPS/Dead Reckoning module, along with Cellular module, there will be two sources of GPS location information. These two are independent and they can be retrieved using different CLIs:

- Cellular modem GPS information can be seen using the **show cellular** <*slot number*> **gps** command.
- GPS/DR module GPS info uses the **show platform hardware gps detail** command.

The following table provides a comparison of Modem based GPS and GPS/Dead Reckoning module based GPS.

Parameters	Modem Based GPS	GPS/Dead Reckoning Module
Type	Cellular Modem based GPS	FRU Based GPS
PIDs Supported	All of the IR1800 series	Only on IR1835 and IR1833
Configuration Modes	Standalone mode	No configuration is available to select modes. The device automatically selects either standalone | dead-reckoning mode based on satellites reception.
Number of satellites needed for co-ordinates	Standalone mode – 4	If a signal is received from 4 or more satellites, standalone GPS co-ordinates will provide co-ordinates else dead-reckoning will provide the gps co-ordinates.
Satellites Supported in show command	Co-ordinates seen in show commands output is based only on GPS.	Co-ordinates seen in the output of show commands is based only on GPS satellites. However, nmea traffic will show GPS, Gallileo, and Glonass.
Initial Calibration Required	No	Yes
Co-ordinates in Absence of Satellite	No Co-ordinates will be acquired and it stays in acquiring status.	The device seamlessly shifts to Dead-reckoning mode and provide co-ordinates based on calculation done by the FRU. The FRU takes into account vehicle speed, direction, last acquired coordinates from the GPS satellite, accelerometer and gyroscope. For dead reckoning to work, the device should have acquired the co-ordinates at least once before the loss of signal from satellite after the router boot up.
Device Name of Controller to use for Configuration	**controller cellular** <*slot*>	**controller gps-dr**

Parameters	Modem Based GPS	GPS/Dead Reckoning Module
CLI to enable feature	**lte gps enable** **lte gps mode standalone** **Note** A modem power cycle is required after enabling the configuration.	**dead-reckoning enable**
CLI to configure nmea	**lte gps nmea**	nmea is automatically enabled once "dead-reckoning enable" is configured
CLI to configure nmea udp socket	**lte gps nmea ip udp** <*source_ip*> <*destination_ip*> <*destination_port*>	**dead-reckoning nmea udp** <*source_ip*> <*destination_ip*> <*destination_port*>
CLI to verify configuration under show running-config	**show run | sec controller cellular**<*slot*>	**show run | sec controller gps-dr**
Show commands to verify gps output	**show cellular** <*slot*> **gps** **show controller cellular** <*slot*> **| inc gps**	**show platform hardware gps detail** **show platform hardware gps mode** **show platform hardware gps status** **show platform hardware gps dead-reckoning**
Access to GPS nmea traffic on IOx side	Supported	Supported
Debug Command	**debug cellular**<*slot*> **messages gps** **debug cellular**<*slot*> **messages nmea**	**debug platform hardware gps_dr** *all* | *dr* | *gps* | *nmea*
Yang Model Support	Yes	Yes

Cellular Modem-Based GPS

Cellular modem based GPS is covered in the Cellular Pluggable Interface Module Configuration Guide.

GPS/Dead Reckoning module (IRM-GNSS-ADR)

This section describes the feature when using the GPS/Dead Reckoning-Based GPS Module.

Note GPS dead-reckoning is only available on the GPS Pluggable Module.

GPS Dead Reckoning

The GPS dead-reckoning feature is supported on the 1835-K9 and 1833-K9 SKUs.

Dead Reckoning Overview

Dead Reckoning is a GPS fallback feature that provides users with location information during satellite signal interruption by calculating the current position by using a previously determined position, and advancing that position based upon known or estimated speeds over elapsed time and course.

IR18xx 3D Automotive Dead Reckoning (3D ADR) provides automotive-grade GPS services by using intelligent algorithms which combines satellite navigation data with wheel speed, gyroscope, and accelerometer data to deliver accurate positioning, even when satellite signals are partially or completely blocked.

The transition from satellite-based location service to internal-data based location service is transparent and automatic, based on the quality and presence of satellite signals.

The feature is disabled by default and CLIs are provided to enable and configure this feature. Enabling the feature automatically enables GPS, DR and the CAN bus. The feature shows the status, configuration and location data. The location data is streamed from the GPS module and will be forwarded to application via socket.

Command Line Interface

This section provides a description of the different CLIs used with GPS Dead Reckoning.

Command	Description
controller gps-dr	GPS-Dead Reckoning can be configured under controller gps-dr command.
dead-reckoning enable	Enables the GPS-DR feature.
no dead-reckoning enable	Disables the GPS-Dead Reckoning feature.
show platform hardware gps detail	Displays the following output: • Feature is enabled/disabled • GPS coordinates • Timestamps • Satellite information with SNR
show platform hardware gps status	Displays whether the feature is enabled or disabled, and the status of coordinates whether acquired or acquiring.
show platform hardware gps mode	Displays whether the feature is enabled or disabled, and whether Dead Reckoning is in use for location fix or not.

Command	Description
show platform hardware gps dead-reckoning	Displays the following output: • Firmware running on the GPS module • CAN transmit/receive count • Odometer reading • Accelerometer • Gyroscope readings • Whether Dead Reckoning is in use for location fix or not
debug platform hardware gps_dr gps | nmea | dr	Enables the debug logs for GPS | NMEA | DR **Note** The console gets flooded if logs are enabled. Configure **no logging console** and then enable this command to avoid flooding of console. Then perform **show log** to see the output. Perform **undebug all** to disable the debug. Make sure to enable logging on the console once debugging is disabled if needed.

Configuration Commands

To enable the GPS Dead Reckoning feature, perform the following:

```
Router#conf t
Enter configuration commands, one per line.  End with CNTL/Z.
Router(config)#controller gps-dr

Router(config-controller)#dead-reckoning enable
 Info: []: DR process enabled successfully.
```

To disable the GPS Dead Reckoning feature, perform the following:

```
Router#conf t
Enter configuration commands, one per line.  End with CNTL/Z.
Router(config)#controller gps-dr
 Router(config-controller)#no dead-reckoning enable
 Info: []: GPS/DR feature disabled successfully
```

Debug Commands

The following debugging commands are available:

```
Router#debug platform hardware gps_dr ?
  all   GPS DR all debug
  dr    GPS DR dr debug
  gps   GPS DR gps debug
  nmea  GPS NMEA messages debug
```

Show Commands

Use the following commands to view the status of the module and GPS details:

```
Router#show inventory
++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
INFO: Please use "show license UDI" to get serial number for licensing.
++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
NAME: "Chassis", DESCR: "Cisco Catalyst IR1835 Rugged Series Router"
PID: IR1835-K9          , VID: V00  , SN: FHH2416P00W
NAME: "Power Supply Module 0", DESCR: "Cisco IR1800 DC Power Supply"
PID: PWR-12V            , VID:      , SN:
NAME: "GE-POE Module", DESCR: "POE Module for On Board GE for Cisco IR183X"
PID: IR-183X-POE        , VID:      , SN:
NAME: "module 0", DESCR: "Cisco IR-1835-K9 Built-In NIM controller"
PID: IR-1835-K9         , VID:      , SN:
NAME: "NIM subslot 0/0", DESCR: "Front Panel 1 port Gigabitethernet Module"
PID: IR1835-1x1GE       , VID: V01  , SN:
NAME: "NIM subslot 0/1", DESCR: "IR1835-ES-4"
PID: IR1835-ES-4        , VID: V01  , SN:
NAME: "module F0", DESCR: "Cisco IR1835-K9 Forwarding Processor"
PID: IR1835-K9          , VID:      , SN:
NAME: "Gps-Dr", DESCR: "Dedicated GNSS/GPS/DR module"
PID: IRM-GNSS           , VID:V03 , SN:FOC243645DJ
```

When GPS co-ordinates are acquired from the Satellite, the following is the output from the show commands:

```
Router#show platform hardware gps detail
GPS Feature =  enabled
GPS Status =  GPS coordinates acquired
Latitude =  37 Deg 25 Min 4.7460 Sec North
Longitude =  121 Deg 55 Min 11.1840 Sec West
Timestamp (GMT) =  Tue Nov 24 03:03:55 2020
Fix type index =  0, Height =  40 m
HDOP =  4.1, GPS Mode Used =  GPS standalone
Satellite Info
---------------
Satellite #30, elevation 72, azimuth 43, SNR 0
Satellite #28, elevation 68, azimuth 277, SNR 0
Satellite #7, elevation 49, azimuth 89, SNR 0
Satellite #13, elevation 37, azimuth 312, SNR 0
Satellite #17, elevation 26, azimuth 185, SNR 25
Satellite #8, elevation 21, azimuth 43, SNR 0
Satellite #9, elevation 15, azimuth 160, SNR 17
Satellite #5, elevation 11, azimuth 260, SNR 26
Satellite #21, elevation 10, azimuth 77, SNR 0
Satellite #19, elevation 7, azimuth 194, SNR 24
Satellite #1, elevation 7, azimuth 103, SNR 0
Satellite #15, elevation 6, azimuth 322, SNR 0
Router#show platform hardware gps dead-reckoning
============================
GPS/DR Vendor Info: TELIT
GPS/DR module FW Version: V33-1.0.5-CLDR-4.7.10-N115R115-003291-3
CAN Bus Status:
  CAN Bus Tx Count: 0
  CAN Bus Rx Count: 0
  CAN NULL packet Bus RX Count: 0
  CAN Bus TX to DR Count: 0
  CAN Bus TX to DR error Count: 0
DR Sample TimeStamp in usec: 0
DR odometer count: 0
DR reverse status: 0
DR in use for location fix: No
time duration for loss of line of sight:
travel distance for loss of line of sight:
travel heading error at exit:
travel yaw error at exit:
```

```
travel gyro gain error at exit:
position error at exit:
position error ratio at exit:
position noise error at exit:
Raw Accel Data in X:  -2360
Raw Accel Data in Y:  16130
Raw Accel Data in Z:  0
Raw Gyro Data in X:  38
Raw Gyro Data in Y:  0
Raw Gyro Data in Z:  0
Router#

Router#show platform hardware gps status
GPS Feature =  enabled
GPS Status =  GPS coordinates acquired
Router#

Router#show platform hardware gps mode
GPS Feature =  enabled
DR in use for location fix: No
Router#
```

When the Antenna is not able to receive a satellite signal, it will switch to Dead Reckoning mode. During this mode only the output from the following show commands will change. The rest of the show commands output remains the same.

```
Router#show platform hardware gps mode
GPS Feature =  enabled
DR in use for location fix: Yes
Router#

Router#show platform hardware gps detail
GPS Feature =  enabled
GPS Status =  GPS coordinates acquired
Latitude =  37 Deg 25 Min 4.7460 Sec North
Longitude =  121 Deg 55 Min 11.1840 Sec West
Timestamp (GMT) =  Tue Nov 24 03:03:55 2020
Fix type index =  0, Height =  40 m
HDOP =  4.1, GPS Mode Used =  DR based GPS

Satellite Info
---------------
Satellite #30, elevation 72, azimuth 43, SNR 0
Satellite #28, elevation 68, azimuth 277, SNR 0
Satellite #7, elevation 49, azimuth 89, SNR 0
Satellite #13, elevation 37, azimuth 312, SNR 0
Satellite #17, elevation 26, azimuth 185, SNR 12
Satellite #8, elevation 21, azimuth 43, SNR 0
Satellite #9, elevation 15, azimuth 160, SNR 14
Satellite #5, elevation 11, azimuth 260, SNR 10
Satellite #21, elevation 10, azimuth 77, SNR 0
Satellite #19, elevation 7, azimuth 194, SNR 8
Satellite #1, elevation 7, azimuth 103, SNR 0
Satellite #15, elevation 6, azimuth 322, SNR 0
Router#
```

When GPS-Dead Reckoning is disabled, the output of the show commands appears as follows:

```
Router#show platform hardware gps detail
GPS Feature =  disabled
GPS Status =  GPS mode not enabled

Router#show platform hardware gps mode
GPS Feature =  disabled
```

```
Router#show platform hardware gps status
GPS Feature =  disabled
GPS Status =  GPS mode not enabled

Router#show platform hardware gps dead-reckoning
============================
GPS/DR Vendor Info:
GPS/DR module FW Version:
CAN Bus Status:
  CAN Bus Tx Count: 0
  CAN Bus Rx Count: 0
  CAN NULL packet Bus RX Count: 0
  CAN Bus TX to DR Count: 0
  CAN Bus TX to DR error Count: 0
DR Sample TimeStamp in usec: 0
DR odometer count: 0
DR reverse status: 0
DR in use for location fix: No
time duration for loss of line of sight:
travel distance for loss of line of sight:
travel heading error at exit:
travel yaw error at exit:
travel gyro gain error at exit:
position error at exit:
position error ratio at exit:
position noise error at exit:
Raw Accel Data in X:  0
Raw Accel Data in Y:  0
Raw Accel Data in Z:  0
Raw Gyro Data in X:  0
Raw Gyro Data in Y:  0
Raw Gyro Data in Z:  0
```

Feature Limitations

The following are feature limitations:

- To acquire an initial timestamp, it is required for the antenna to receive a signal from the Satellite when the device is powered on. Once acquired, the timestamp will be updated every second. Additionally, for DR to display coordinates, it is required for the antenna to acquire coordinates from the satellite at least once after the device is powered on, and DR is enabled. If the device had acquired coordinates before a power down, and the device is powered on again later, the device may try to show the coordinates based on the last known location.
- Cisco recommends using this feature only if the vehicle CAN bus is connected to this IR18xx. CAN Bus connection ensures that the GPS module is properly calibrated before the DR feature is fully functional, and location fix can be obtained even without line of sight of satellites.
- As long as sufficient satellite signals are received, the coordinates will be acquired as a standalone GPS module, whether connected to the vehicle via CAN bus or not. However, if there are no sufficient signals (or no signals at all), the location fix using DR will kick in and be accurate only if its CAN interface is connected to the vehicle via the CAN bus and gets all the required vehicle data.
- In the event of a GPS module initial deployment in the field, if CAN bus is not connected and satellite signal is not received, coordinates cannot be acquired.
- In the event of a GPS module has obtained the coordinates before, if Satellite signal is not received, coordinates will be acquired with the previously obtained value through DR, whether the CAN bus is

not connected or not. The accuracy depends on whether the location has been moved or not since last location fix with satellite signals.

IR1800 GPS DR Module Calibration

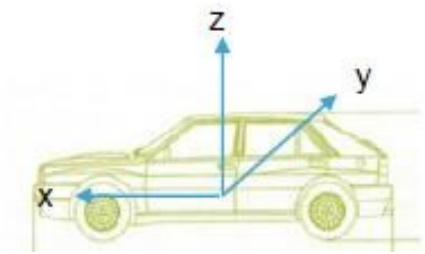
IR1800 GPS DR module provides Automotive Dead Reckoning (ADR) feature that allows the navigation on a automotive platform even when there are not a sufficient number of GPS satellite signals available. This is very common in IoT applications such as in urban canyons, underground tunnels, or any areas where line of sights of satellites are blocked.

To provide such a service in 3-D, the module needs to use the chipset's built-in three-axis gyro and accelerometer sensors to obtain the data for the change of the direction as well as the orientation and elevation of the chipset. In addition, to provide automotive grade service, it also allows the inputs of vehicle speed and direction data obtained directly from the CAN bus interface on IR1800.

In order to allow the module to properly interpret the data received from the three-axis sensors, it is necessary to provide the information describing the orientation of these built-in sensors relative to the vehicle, thus to properly calibrate the module, once the IR1800 is installed in the vehicle.

Calibration Requirements

The orientation of the vehicle is used in the module calibration as the base reference. It is arranged as illustrated in the following graphic:



- X axis: points to the vehicle forward direction
- Y axis: points to the right side of the vehicle when the viewer looks forward
- Z axis: points to upward with respect to the vehicle’s motion plane.

The DR calibration process requires obtaining the orientation information of the sensors relative to this base reference. The default orientation for this module, as used by the DR algorithm is described as the following:

- IR1800 is installed and fastened on a stable base. Its front panel faces the right side of the vehicle, for example, 90 degrees clockwise away from the vehicle forward direction. The front panel is the side with the power connector and ethernet ports.
- With the default orientation, there is no need to input any sensor orientation data to the module for the calibration. The default data will be assumed by the module for the entire calibration process.

Release 17.6.1 does not support the calibration with non-default orientation. Cisco will provide such support if any future requirements arise. For now, customers need to follow this instruction for default orientation for the module calibration.

Calibration Process

Begin with the router installed as previously described, and with the GPS module installed. Configure the module from the command line:

Procedure

Step 1 This step is only required if the DR feature is not yet enabled. Otherwise, go directly to step 2.

Example:

```
Router(config)#controller gps
Router(config-controller)#no dead-reckoning enable
Info: []: GPS/DR feature disabled successfully
Router(config-controller)#end
```

Step 2 Enable the DR feature (clear the old calibration data)

Example:

```
Router#conf t
Enter configuration commands, one per line. End with CNTL/Z.
Router(config)#controller gps
Router(config-controller)#dead-reckoning enable
Info:  DR process enabled successfully.
```

Step 3 Set the CAN bus baudrate. Please check the user manual of the vehicle for the baudrate value. Typically, 500kbps is most common, but it can vary among different vehicle manufacturers.

Example:

```
Router(config-controller)#exit
Router(config)#canbus baudrate ?
  <125000-1000000>  enter baud rate ranging from 125000 to 1000000
Router(config)#canbus baudrate <baudrate of the vehicle CAN/OBDII interface>
```

Step 4 Reload or power cycle the router and wait until the router finishes rebooting.

Calibration Execution

Typically, a vehicle driving is needed to calibrate the DR module. This process is only needed once as long as the module orientation relative to the vehicle is not changed. This is true as long as the router or the module is never repositioned inside this vehicle.

Follow these steps for calibration:

Procedure

Step 1 Choose a location where there is open sky over the entire test for a good GPS signal. Calibration will require the vehicle to be moving.

Step 2 Wait approximately 2 minutes in a stopped position with the router and vehicle on. This allows the module to initialize the yaw rate offset with reliable values.

Step 3 Log into the console and run the following command to ensure that the GPS location is acquired:

Example:

```
Router#show platform hardware gps detail
GPS Feature =  enabled
GPS Status =  GPS coordinates acquired
Latitude =  37 Deg 25 Min 5.8200 Sec North
Longitude =  121 Deg 55 Min 9.1020 Sec West
Timestamp (GMT) =  Tue Mar  9 02:36:00 2021
Fix type index =  0, Height =  1 m
HDOP =  1.0, GPS Mode Used =  GPS standalone
Satellite Info
----------------
Satellite #12, elevation 70, azimuth 147, SNR 22
Satellite #25, elevation 63, azimuth 305, SNR 25
Satellite #2, elevation 51, azimuth 43, SNR 23
Satellite #5, elevation 41, azimuth 139, SNR 20
Satellite #29, elevation 33, azimuth 294, SNR 26
Satellite #6, elevation 13, azimuth 46, SNR 18
Satellite #31, elevation 10, azimuth 317, SNR 14
Satellite #24, elevation 7, azimuth 195, SNR 0
Satellite #18, elevation 4, azimuth 230, SNR 0
Satellite #82, elevation 58, azimuth 326, SNR 22
Satellite #80, elevation 57, azimuth 289, SNR 18
Satellite #79, elevation 41, azimuth 190, SNR 22
Satellite #81, elevation 34, azimuth 37, SNR 25
Satellite #83, elevation 28, azimuth 267, SNR 0
Satellite #66, elevation 19, azimuth 79, SNR 0
Satellite #73, elevation 16, azimuth 330, SNR 18
Satellite #67, elevation 8, azimuth 129, SNR 0
Satellite #65, elevation 4, azimuth 21, SNR 16
```

Step 4 After waiting for 2 minutes, drive in a straight line direction for at least five minutes at a constant speed. The speed should be greater than 35 km/h (approx. 22mph).

Step 5 Following the straight line drive, make several left and right turns of at least 90 degrees, allowing the system to calculate the gyro yaw rate gain.

Important
Calibration will be improved with more turns completed. A minimum of 10 turns is recommended. Calibration should be performed in an open sky environment. Avoid urban canyons, tunnels, parking garages, dense foliage, etc.

Step 6 To complete calibration, the vehicle should stop and remain stationary for at least 10 seconds. For a full calibration to be successful, the above procedure must be followed.

Step 7 Use the following command to check if the calibration is done:

Example:

```
Router#show platform hardware gps dead

============================
GPS/DR Vendor Info: TELIT
GPS/DR module FW Version: V33-1.0.5-CLDR-4.7.10-N115R115-003291-3
DR Calibration Status:
  DR is calibrated
  Odometer is calibrated
  Gain is calibrated
  Offset is  calibrated
CAN Bus Status:
  CAN Bus Tx Count: 6856
```

```
  CAN Bus Tx error Count: 0
  CAN Bus Rx Count: 12724
  CAN NULL packet Bus RX Count: 0
  CAN Bus Rx unsupported packet Count: 0
  CAN Bus TX to DR Count: 12601
  CAN Bus TX to DR error Count: 123
DR data:
  DR Sample TimeStamp in usec: 0
  DR odometer count received from module: 54597690
  DR odometer count sent to module: 54597697
  DR odometer is  valid from module
  DR odometer delta count from module: 220
  DR reverse status: 0
```

Step 8 To clear calibration (for testing purposes), follow steps 1 and 2 under Calibration Process.

Dead Reckoning for GPS NMEA data streaming

The NMEA data streaming feature allows the user to forward NMEA streams over the Internet to any device running a 3rd party application for GPS location service.

The CLIs for IPv4 UDP sockets will be supported as feature parity with existing 4G modem based GPS functionality. No IPv6 UDP port support for NMEA data streaming is supported in existing 4G modem GPS at this time.

Command Line Interface

```
(config-controller)# dead-reckoning nmea ?
ip NMEA over IP interface

Router(config-controller)# dead-reckoning nmea udp ?
  A.B.C.D  Source address

(config-controller)#dead-reckoning nmea udp 10.3.4.5 ?
 A.B.C.D Destination address

(config-controller)#dead-reckoning nmea udp 10.1.1.1 10.3.4.5 ?
  Destination port

(config-controller)#dead-reckoning nmea udp 10.1.1.1 10.3.4.5 3456
```

GPS and Dead Reckoning Support for the J1939 Connector

Automotive Dead-Reckoning (DR) refers to the capability of a GNSS receiver to continue to navigate on an automotive platform when there are an insufficient number of GNSS satellite signals available. To do this, the receiver uses information provided by external sensors concerning the state of the vehicle in order to propagate the navigation solution.

Automotive DR requires information regarding the change in directional heading of the vehicle, which is provided by a three-axis digital gyroscope. Automotive DR also requires information about speed and direction of the vehicle. Speed is provided by an odometer (wheel tick) count, which is input into the IRM-GNSS-ADR pluggable module.

The automotive DR feature also accepts data from a three-axis digital accelerometer, which provides information that can be used to determine the orientation of the gyro when it is installed at a tilt angle. This information

is also used to estimate elevation. The accelerometer is integrated within the sensor included inside the pluggable module.

Prior to the 17.9.1 release, only mode obdii was available. In 17.9.1, mode j1939 is added with the existing default mode obdii.

The J1939 connector is supported on heavy duty trucks, which provides speed and reverse status data to be fed into the GPS/DR module using J1939 protocol. It is configured through the command line interface under the controller.

Configuration

The following CLIs are available.

To show what is available for dead reckoning:

```
Router(config-controller)#dead-reckoning ?
enable enable GPS feature
mode DR mode configuration
nmea NMEA Configuration
```

To configure mode j1939:

```
Router(config)#controller Gps-Dr
Router(config-controller)#dead-reckoning mode j1939
```

To view the status:

```
Router#show platform hardware gps dead-reckoning
============================
DR Vehicle interface mode: J1939
GPS/DR Vendor Info: TELIT
GPS/DR module FW Version: V33-1.0.5-CLDR-4.7.10-N115R115-003291-3
DR Calibration Status:
DR is not calibrated
Odometer is not calibrated
Gain is not calibrated
Offset is not calibrated

CAN Bus Status:
CAN Bus Tx Count: 1874
CAN Bus Tx error Count: 0

CAN Bus Rx Count: 571
CAN NULL packet Bus RX Count: 0
CAN Bus Rx unsupported packet Count: 448

CAN Bus TX to DR Count: 123
CAN Bus TX to DR error Count: 0

DR data:
DR Sample TimeStamp in usec: 0
DR odometer count received from module: 0
DR odometer count sent to module: 1353
DR odometer is not valid from module
DR odometer delta count from module: 0
DR reverse status: 0

DR in use for location fix: No
time duration for loss of line of sight:
travel distance for loss of line of sight:
travel heading error at exit:
travel yaw error at exit:
travel gyro gain error at exit:
```

```
position error at exit:
position error ratio at exit:
position noise error at exit:
Raw Accel Data in X: 0
Raw Accel Data in Y: 0
Raw Accel Data in Z: 0
Raw Gyro Data in X: 0
Raw Gyro Data in Y: 0
Raw Gyro Data in Z: 0
```

National Marine Electronics Association (NMEA) IOx Support

From linux or the IOx container, the following tty is available for NMEA traffic:

- /dev/ttyTun9
- /dev/ttyS2

NMEA UDP Socket Support

In order to configure NMEA UDP socket support, you must enable the dead reckoning feature first, and then configure the NMEA UDP socket support. In order to disable NMEA UDP socket support, you must disable NMEA UDP socket support first, and then disable the dead reckoning feature.

See the following examples.

Enable the Feature

GPS is disabled:

```
Router#show platform hardware gps detail
GPS Feature =  disabled
GPS Status =  GPS mode not enabled

Router#show platform hardware gps status
GPS Feature =  disabled
GPS Status =  GPS mode not enabled

Router#show platform hardware gps mode
GPS Feature =  disabled

Router#show platform hardware gps dead-reckoning
============================
GPS/DR Vendor Info:
GPS/DR module FW Version:
DR Calibration Status:
  DR is not calibrated
  Odometer is not calibrated
  Gain is not calibrated
  Offset is not calibrated

CAN Bus Status:
  CAN Bus Tx Count: 0
  CAN Bus Tx error Count: 0

  CAN Bus Rx Count: 0
  CAN NULL packet Bus RX Count: 0
```

```
  CAN Bus Rx unsupported packet Count: 0

  CAN Bus TX to DR Count: 0
  CAN Bus TX to DR error Count: 0

DR data:
  DR Sample TimeStamp in usec: 0
  DR odometer count received from module: 0
  DR odometer count sent to module: 0
  DR odometer is not valid from module
  DR odometer delta count from module: 0
  DR reverse status: 0
DR in use for location fix: No
time duration for loss of line of sight:
travel distance for loss of line of sight:
travel heading error at exit:
travel yaw error at exit:
travel gyro gain error at exit:
position error at exit:
position error ratio at exit:
position noise error at exit:
Raw Accel Data in X:  0
Raw Accel Data in Y:  0
Raw Accel Data in Z:  0
Raw Gyro Data in X:  0
Raw Gyro Data in Y:  0
Raw Gyro Data in Z:  0
```

Configure GPS Dead Reckoning:

```
Router#config term
Enter configuration commands, one per line.  End with CNTL/Z.

Router(config)#controller Gps-Dr

Router(config-controller)#dead-reckoning enable
Info:  DR process enabled successfully.
```

Configure Dead Reckoning NMEA UDP:

```
Router(config-controller)#dead-reckoning nmea udp 192.0.2.163 192.0.2.240 11111
NMEA UDP Socket connect successful.

Router(config-controller)#end
```

Verify the status:

```
Router#show run | sec controller Gps-Dr
controller Gps-Dr
 dead-reckoning enable
 dead-reckoning nmea udp 192.0.2.163 192.0.2.240 11111
Router#show platform hardware gps detail
GPS Feature =  enabled
NMEA UDP socket is in use
NMEA UDP socket operational status: active
GPS Status =  GPS acquiring
Latitude =  0 Deg 0 Min 0 Sec North
Longitude =  0 Deg 0 Min 0 Sec East
Timestamp (GMT) =  Sun Jan  6 00:00:00 1980

Fix type index =  0
HDOP =  , GPS Mode Used =  not configured

Satellite Info
----------------
```

```
Router#show platform hardware gps status
GPS Feature =  enabled
NMEA UDP socket is in use
NMEA UDP socket operational status: active
GPS Status =  GPS acquiring

Router#show platform hardware gps mode
GPS Feature =  enabled
DR in use for location fix: No

Router#show platform hardware gps dead-reckoning
============================
GPS/DR Vendor Info:
GPS/DR module FW Version:
DR Calibration Status:
  DR is not calibrated
  Odometer is not calibrated
  Gain is not calibrated
  Offset is not calibrated

CAN Bus Status:
  CAN Bus Tx Count: 135
  CAN Bus Tx error Count: 0

  CAN Bus Rx Count: 0
  CAN NULL packet Bus RX Count: 0
  CAN Bus Rx unsupported packet Count: 0

  CAN Bus TX to DR Count: 0
  CAN Bus TX to DR error Count: 0

DR data:
  DR Sample TimeStamp in usec: 0
  DR odometer count received from module: 0
  DR odometer count sent to module: 0
  DR odometer is not valid from module
  DR odometer delta count from module: 0
  DR reverse status: 0

DR in use for location fix: No
time duration for loss of line of sight:
travel distance for loss of line of sight:
travel heading error at exit:
travel yaw error at exit:
travel gyro gain error at exit:
position error at exit:
position error ratio at exit:
position noise error at exit:
Raw Accel Data in X:  0
Raw Accel Data in Y:  0
Raw Accel Data in Z:  0
Raw Gyro Data in X:  0
Raw Gyro Data in Y:  0
Raw Gyro Data in Z:  0
```

Disable the Feature

Reverse the procedure to disable NMEA UDP Support. See the following examples:

```
Router#config term
Enter configuration commands, one per line.  End with CNTL/Z.

Router(config)#controller Gps-Dr
```

```
Router(config-controller)#no dead-reckoning nmea udp 192.0.2.163 192.0.2.240 11111
NMEA UDP Socket is disabled successfully.

Router(config-controller)#no dead-reckoning enable
Info:  GPS/DR feature disabled successfully

Router(config-controller)#end
```

Verify the status:

```
Router#show platform hardware gps detail
GPS Feature =  disabled
GPS Status =  GPS mode not enabled

Router#show platform hardware gps status
GPS Feature =  disabled
GPS Status =  GPS mode not enabled

Router# show platform hardware gps mode
GPS Feature =  disabled
Router#

Router# show platform hardware gps dead-reckoning
============================
GPS/DR Vendor Info:
GPS/DR module FW Version:
DR Calibration Status:
  DR is not calibrated
  Odometer is not calibrated
  Gain is not calibrated
  Offset is not calibrated

CAN Bus Status:
  CAN Bus Tx Count: 0
  CAN Bus Tx error Count: 0

  CAN Bus Rx Count: 0
  CAN NULL packet Bus RX Count: 0
  CAN Bus Rx unsupported packet Count: 0

  CAN Bus TX to DR Count: 0
  CAN Bus TX to DR error Count: 0

DR data:
  DR Sample TimeStamp in usec: 0
  DR odometer count received from module: 0
  DR odometer count sent to module: 0
  DR odometer is not valid from module
  DR odometer delta count from module: 0
  DR reverse status: 0

DR in use for location fix: No
time duration for loss of line of sight:
travel distance for loss of line of sight:
travel heading error at exit:
travel yaw error at exit:
travel gyro gain error at exit:
position error at exit:
position error ratio at exit:
position noise error at exit:
Raw Accel Data in X:  0
Raw Accel Data in Y:  0
Raw Accel Data in Z:  0
Raw Gyro Data in X:  0
```

```
Raw Gyro Data in Y:  0
Raw Gyro Data in Z:  0
Router#
```

NMEA UDP Configuration with Yang

The Yang Model can be used to enable the feature in the same way as the command line. The same rules apply:

In order to configure NMEA UDP socket support, you must enable the dead reckoning feature first, and then configure the NMEA UDP socket support. In order to disable NMEA UDP socket support, you must disable NMEA UDP socket support first, and then disable the dead reckoning feature.

Enable Dead Reckoning:

```
<rpc xmlns="urn:ietf:params:xml:ns:netconf:base:1.0" message-id="101">
  <edit-config>
    <target>
      <running/>
    </target>
    <config>
      <native xmlns="http://cisco.com/ns/yang/Cisco-IOS-XE-native">
        <controller>
          <Gps-Dr xmlns="http://cisco.com/ns/yang/Cisco-IOS-XE-controller">
            <dead-reckoning>
              <enable/>
            </dead-reckoning>
          </Gps-Dr>
        </controller>
      </native>
    </config>
  </edit-config>
</rpc>
```

Enable UDP Socket:

```
<rpc xmlns="urn:ietf:params:xml:ns:netconf:base:1.0" message-id="101">
  <edit-config>
    <target>
      <running/>
    </target>
    <config>
      <native xmlns="http://cisco.com/ns/yang/Cisco-IOS-XE-native">
        <controller>
          <Gps-Dr xmlns="http://cisco.com/ns/yang/Cisco-IOS-XE-controller">
            <dead-reckoning>
              <nmea>
                <udp>
                  <source-address>172.27.169.162</source-address>
                  <destination-address>172.27.169.140</destination-address>
                  <destination-port>11111</destination-port>
                </udp>
              </nmea>
            </dead-reckoning>
          </Gps-Dr>
        </controller>
      </native>
    </config>
  </edit-config>
</rpc>
```

Get Status:

```
<rpc xmlns="urn:ietf:params:xml:ns:netconf:base:1.0" message-id="101">
  <get>
    <filter>
      <gnss-dr-oper-data xmlns="http://cisco.com/ns/yang/Cisco-IOS-XE-gnss-dr-oper">
        <gnss-dr-data/>
      </gnss-dr-oper-data>
    </filter>
  </get>
</rpc>
```

Delete UDP Socket:

```
<rpc xmlns="urn:ietf:params:xml:ns:netconf:base:1.0" message-id="101">
  <edit-config>
    <target>
      <running/>
    </target>
    <config>
      <native xmlns="http://cisco.com/ns/yang/Cisco-IOS-XE-native">
        <controller>
          <Gps-Dr xmlns="http://cisco.com/ns/yang/Cisco-IOS-XE-controller">
            <dead-reckoning>
              <nmea>
              <udp xmlns:nc="urn:ietf:params:xml:ns:netconf:base:1.0" nc:operation="delete">

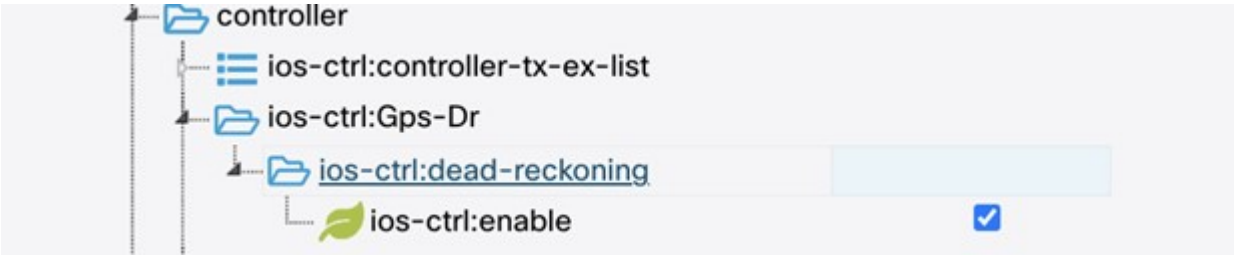
                  <source-address>172.27.169.162</source-address>
                  <destination-address>172.27.169.240</destination-address>
                  <destination-port>11111</destination-port>
                </udp>
              </nmea>
            </dead-reckoning>
          </Gps-Dr>
        </controller>
      </native>
    </config>
  </edit-config>
</rpc>
```

Delete Dead Reckoning Configuration:

```
<rpc xmlns="urn:ietf:params:xml:ns:netconf:base:1.0" message-id="101">
  <edit-config>
    <target>
      <running/>
    </target>
    <config>
      <native xmlns="http://cisco.com/ns/yang/Cisco-IOS-XE-native">
        <controller>
          <Gps-Dr xmlns="http://cisco.com/ns/yang/Cisco-IOS-XE-controller">
            <dead-reckoning>
              <enable xmlns:nc="urn:ietf:params:xml:ns:netconf:base:1.0"
nc:operation="delete"/>
            </dead-reckoning>
          </Gps-Dr>
        </controller>
      </native>
    </config>
  </edit-config>
</rpc>
```

Yang Data Model Support

Controller yang model is present under Cisco-IOS-XE-controller - Cisco-IOS-XE-native:



The following is an XML example to enable the GPS-Dead Reckoning feature:

```
<rpc xmlns="urn:ietf:params:xml:ns:netconf:base:1.0" message-id="101">
  <edit-config>
    <target>
      <running/>
    </target>
    <config>
      <native xmlns="http://cisco.com/ns/yang/Cisco-IOS-XE-native">
        <controller>
          <Gps-Dr xmlns="http://cisco.com/ns/yang/Cisco-IOS-XE-controller">
            <dead-reckoning>
              <enable/>
            </dead-reckoning>
          </Gps-Dr>
        </controller>
      </native>
    </config>
  </edit-config>
</rpc>
```

The following is an XML example to disable the GPS-Dead Reckoning feature:

```
<rpc xmlns="urn:ietf:params:xml:ns:netconf:base:1.0" message-id="101">
  <edit-config>
    <target>
      <running/>
    </target>
    <config>
      <native xmlns="http://cisco.com/ns/yang/Cisco-IOS-XE-native">
        <controller>
          <Gps-Dr xmlns="http://cisco.com/ns/yang/Cisco-IOS-XE-controller">
            <dead-reckoning>
              <enable xmlns:nc="urn:ietf:params:xml:ns:netconf:base:1.0"
nc:operation="delete"/>
            </dead-reckoning>
          </Gps-Dr>
        </controller>
      </native>
    </config>
  </edit-config>
</rpc>
```

GPS-Dead Reckoning oper commands are present under Cisco-IOS-XE-gnss-dr-oper model:



The following is an example of the XML for the oper command yang model of GPS-Dead Reckoning:

```
<rpc xmlns="urn:ietf:params:xml:ns:netconf:base:1.0" message-id="101">
  <get>
    <filter>
      <gnss-dr-oper-data xmlns="http://cisco.com/ns/yang/Cisco-IOS-XE-gnss-dr-oper">
        <gnss-dr-data/>
      </gnss-dr-oper-data>
    </filter>
  </get>
</rpc>
```

YANG operational and configuration data support will be provided for the previously mentioned CLIs as well.

The YANG model will allow the user to configure the UDP socket (ip address, UDP port etc, under XE-controller), as well as display the operational status of this feature (under XE-gnss-dr-oper), via YANG application software.

Example: Connecting to a Server Hosting a GPS Application

You can feed the NMEA data to a remote server that hosts the GPS application. The server can be connected to the router either directly using an Ethernet cable or through a LAN or WAN network. If the application supports serial port, run a serial port emulation program to create a virtual serial port over the LAN or WAN connection.

Note Microsoft Streets & Trips is a licensed software that you can download from the Microsoft website.

To connect a Cisco 4G LTE-Advanced through IP to a PC running Microsoft Streets & Trips, perform the following steps:

1. Connect the PC to the router using an Ethernet cable.
2. nsure that the PC and router can ping.
3. Launch the serial port redirector on the PC.
4. Create a virtual serial port that connects to the NMEA port on the router.

5. Launch **Microsoft Streets & Trips** on your PC.
6. Select the GPS Menu.
7. Click Start Tracking.
8. If you have acquired a location fix from the **show cellular 0/3/0 gps** command output on the router, the current location is plotted on the graph, and a reddish brown dotted cursor with a circle around it is seen on the map.

Note If you have not acquired a location fix, the Microsoft application times out and disconnects.

GNSS Support on the GPS/Dead Reckoning Module (IRM-GNSS-ADR)

Prior to the Cisco IOS XE 17.11.1a release, the only GNSS constellation supported was GPS. This release introduces support for GPS and Galileo.

Note Only ONE constellation can be enabled at a time.

There are new CLI options available to support the new constellation:

Configuration Commands:

```
(config-controller)# controller gps
<no> dead-reckoning constellation <gps | galileo |gnss >
```

Note The default setting is gps mode. The new galileo and gnss options in the above CLI example is used to configure Galileo and Multiple/Simultaneous GNSS (GPS + Galileo etc) respectively.

Show Commands

```
show platform hardware gps <mode | status | details>
....
Current Constellation Configured =  gps | galileo | gnss
....
```

Any changes made to the configuration will require the router to be rebooted.

More information is available in the Configuring GPS chapter of the IR1800 Software Configuration Guide.

Galileo support on the LTE pluggable modules

The Cisco IOS XE 17.11.1a release introduces support for the Galileo constellation on LTE pluggable modules.

GPS was the only GNSS constellation supported in earlier releases. The Cisco IOS XE 17.11.1a release introduces support for Galileo constellation.

Note You can enable only one constellation at a time.

Configuration command examples:

```
config# controller cellular<slot/port>
(config-controller)# <no> lte gps constellation <gps | galileo | gnss >
```

Example:

```
(config-controller)# lte gps constellation ?
galileo  select Galileo as active constellation
gps      select GPS as active constellation
gnss      select multiple GNSS as active constellation
```

Note The default setting is GPS mode.

The galileo and gnss options in this CLI configure Galileo and simultaneous GNSS modes, such as GPS and Galileo.

If you disable the GPS configuration, make sure that no constellation is configured. This maintains consistency with GPS mode configuration.

Example:

```
config#

controller Cellular 0/4/0

(config-controller)# no lte gps constellation gps
```

Show commands example:

Note The example shows the current GNSS constellation as Galileo.

Example:

```
config#

show cellular 0/4/0 gps detail

GPS Feature =  enabled
```

```
GPS Mode Configured =  standalone
Current Constellation Configured =  galileo
GPS Port Selected =  Dedicated GPS port
GPS Status =  GPS acquiring
```

Any changes made to the configuration will require the router to be rebooted.

More information is available in the Cellular Pluggable Interface Module Configuration Guide .

Access Accelerometer and Gyro Sensor Data from IRM-GNSS

This feature allows accelerometer and gyro sensor data from IRM-GNSS (GPS DR) module to be streamed to the IOX via a TTY in IR1800. Prior to this release, the IRM-GNSS module pushed the sensor data to the host in NMEA via port /dev/ttyS2. Previous IOS XE releases already parsed and cached the data.

The feature will forward the sensor data to IOX via the TTY whenever the data is received from the NMEA. Currently, there is no control on the frequency the data is sent from the module, which totally relies on the module itself.

There are no new commands for this feature. It is enabled by default once dead reckoning is enabled. Existing CLIs can be used to view the sensor data, for example:

```
Router# show platform hardware gps dead-reckoning

DR Vehicle interface mode: OBDDII
GPS/DR Vendor Info: TELIT
GPS/DR module FW Version: V33-1.0.5-CLDR-4.7.10-N115R115-003291-3
…
Raw Accel Data in X:  -542
Raw Accel Data in Y:  538
Raw Accel Data in Z:  16964
Raw Gyro Data in X:  153
Raw Gyro Data in Y:  -80
Raw Gyro Data in Z:  99
```

The existing **debug platform hardware gps dead-reckoning** command has been enhanced to provide additional debug messages for better serviceability. The debug messages will cover the following:

- How frequent the sensor data are pushed from the module to IOS, it must at least once per second.
- The latest sensor data received from the module.

Change in Vendor for GNSS Module

This feature applies to the IR1833 and IR1835 only. There was a change in chip manufacturers on the IRM-GNSS-ADR pluggable module. There have been no changes in functionality, however you will see a change in the display of vendor information and firmware version.

See the following example:

```
Router#show platform hardware gps dead-reckoning
DR Vehicle interface mode: OBDDII
GPS/DR Vendor Info: VIC3DA
GPS/DR module FW Version: 4.6.18.11
DR Calibration Status:
DR is not calibrated
Odometer is not calibrated
```

```
Gain is not calibrated
Offset is not calibrated

CAN Bus Status:
CAN Bus Tx Count: 11
CAN Bus Tx error Count: 150930
```

IOX Access to IR1800 On-board Accelerometer and Gyroscope

This feature allows on-board accelerometer and gyroscope sensor data to be streamed to IOX via a TTY. This feature is disabled by default, and will keep feature parity with IR829 accelerometer and gyroscope sensor data feature. The CLIs for this feature are defined below.

Configuration Commands

The following commands are available:

```
Router(config)#acc-gyro ?
  enable     Enable
  frequency  Frequency in reading

Router(config)#acc-gyro freq ?
  four/sec  Reading 4 times per second
  one/min   Reading 1 times per minute
  one/sec   Reading 1 time per second (default value)
  ten/min   Reading 10 times per minute
```

Show Commands

The following command is available:

```
Router# show platform hardware acc-gyro sensor-data

Date                 Time   G-X     G-Y     G-Z    XL-X    XL-Y   XL-Z
2022:10:26:16:58:13.855143 1137.50 -297.50 621.25 -18.056 -3.111 -966.057
2022:10:26:16:58:14.863668 1058.75 -122.50 735.00 -17.629 -2.989 -965.996
2022:10:26:16:58:15.869117 1207.50 -140.00 726.25 -18.361 -3.294 -965.813
2022:10:26:16:58:16.874036 1268.75 -192.50 717.50 -18.178 -3.050 -965.874
2022:10:26:16:58:17.884764 1163.75 -420.00 717.50 -18.056 -2.989 -965.813
2022:10:26:16:58:18.894063 1347.50 -148.75 708.75 -18.117 -3.477 -965.935
2022:10:26:16:58:19.900830 1137.50 -315.00 577.50 -18.239 -3.599 -965.935
2022:10:26:16:58:20.908765 1137.50 -131.25 726.25 -17.873 -3.538 -965.813
2022:10:26:16:58:21.916674 1137.50 -262.50 726.25 -18.361 -2.867 -965.935
2022:10:26:16:58:22.927371 1137.50 -323.75 516.25 -17.934 -3.477 -965.569
2022:10:26:16:58:23.934275 1120.00 -647.50 516.25 -18.361 -3.416 -965.752
2022:10:26:16:58:24.940819 1111.25 -262.50 743.75 -18.422 -2.989 -965.386
2022:10:26:16:58:25.947471 1190.00 -201.25 673.75 -17.995 -3.416 -966.057
2022:10:26:16:58:26.953120 1093.75 -288.75 577.50 -17.995 -3.233 -965.874
2022:10:26:16:58:27.961469 1137.50 -428.75 551.25 -18.117 -2.745 -965.996
2022:10:26:16:58:28.971354 1050.00 -271.25 717.50 -18.361 -3.233 -965.508
2022:10:26:16:58:29.981967 1172.50 78.75 840.00 -18.117 -3.538 -965.386
```

Other

The existing **debug hardware acc-gyro sensor-data** command has been enhanced to provide additional debug messages for better serviceability. The debug messages will cover the following:

- How frequent the sensor data are pushed from the module to IOS, it must at least once per second.
- The latest sensor data received from the module.

Configure DR Module using Cisco Catalyst SD-WAN Manager

The IR1833 and IR1835 routers incorporate a Dead Reckoning (DR) module that calibrates data to estimate a vehicle's location when the GPS signal is weak or unavailable. The Cisco Catalyst SD-WAN Manager facilitates the use of DR CLI commands through the CLI template to estimate GPS coordinates.

Cisco Catalyst SD-WAN Manager leverages YANG data models to configure the DR module. It translates these DR CLI commands into YANG models and subsequently pushes them to the device.

Configuration Example

You can configure the device that supports the DR module through the CLI template.

This is the sample configuration as seen in the Cisco Catalyst SD-WAN Manager CLI template.

```
controller Gps-Dr
dead-reckoning enable
dead-reckoning nmea udp 192.168.1.1 192.168.2.1 1111
dead-reckoning mode j1939
dead-reckoning can-filter extended
dead-reckoning constellation gnss
dead-reckoning iox-acc-frequency ten/min
dead-reckoning iox-gyro-frequency four/sec
dead-reckoning tty-baudrate 115200
End
canbus baudrate 500000
```

Verification

Verify the status of DR module

```
Router#show platform hardware gps dead-reckoning
DR Vehicle interface mode: OBDDII
GPS/DR Vendor :
GPS/DR Vendor Info:
GPS/DR module FW Version: 4.6.18.11
DR Calibration Status:
DR is calibrated in DRUM-DRAW mode
 Odometer is calibrated
 Gain is calibrated
 Offset is calibrated

CAN Bus Status:
  CAN Bus Tx Count: 8621
  CAN Bus Tx error Count: 0
  CAN Bus Rx Count: 38519
  CAN Bus Rx Missed Count: 0
  CAN NULL packet Bus RX Count: 0
  CAN Bus Rx unsupported packet Count: 2841
  CAN Bus TX to DR Count: 8634
  CAN Bus TX to DR error Count: 0
```

GPS NMEA unique identifier

Starting with Cisco IOS-XE Release 26.1.1, Cisco routers can send a Unique Identifier (UID) to identify the source device when streaming NMEA GPS sentences over UDP on the Cisco IR1101 and Cisco IR1835

routers. UID streaming is supported from the Dead Reckoning (DR) module and from cellular modems that support GNSS. The UID uniquely identifies each router in GPS data processing systems.

Table 19: Feature History Table

Feature name	Release information	Feature description
Send a UID to identify the unique source when sending GPS coordinates data	Release 26.1.1	This feature allows you to configure Cisco routers to send a Unique Identifier (UID) while streaming NMEA GPS sentences over UDP. This UID streaming is supported by both the Dead Reckoning (DR) module and cellular modems with GNSS capabilities, enabling each router to be uniquely identified in GPS data processing systems.

Cisco proprietary NMEA sentence format

When you configure a UID, a Cisco proprietary NMEA sentence is sent alongside the standard NMEA sentence in the same UDP packet. The format is:

```
$PCSCU,<UID>,<Source>,<Router Time>,<NMEA Sentence Time>,<NMEA Sentence Checksum>,,,,,,*95
```

- `CSC` indicates Cisco
- `U` indicates this sentence provides a unique identifier
- `<UID>` is the serial number, hostname, or custom ID
- `<Source>` indicates the origin of the NMEA stream:
  - `0`: cellular modem0
  - `1`: cellular modem1
  - `2`: DR module

For example, for cellular modem1, if you use the serial number as the UID, the Cisco proprietary NMEA sentence appears as:

```
$PCSCU,FCW2447P0EU,1,213431.100,213431.000,75,,,,,,*95
```

In this NMEA sentence

- `CSC` indicates Cisco
- `U` indicates that this sentence provides a unique identifier
- `FCW2447P0EU` indicates serial number as the UID
- `1` indicates source as cellular modem1
- `1213431.100` indicates the router time
- `213431.000` indicates the NMEA sentence time

- `75` indicates the NMEA sentence checksum

NMEA sentence selection

By default, all NMEA sentences from the modem or DR module are sent to the configured UDP destination. To reduce UDP traffic, you can select specific NMEA sentence types to send using a hex bitmask. The table presents each NMEA sentence type and its corresponding HEX value.

Table 20: NMEA sentence type and HEX value

HEX value	NMEA sentence type
0x01	GGA
0x02	GSA
0x04	GSV
0x08	RMC
0x10	VTG
0x20	GNS

Configure dead reckoning GPS NMEA unique identifier

Follow these steps to configure dead reckoning GPS NMEA UID.

Procedure

Step 1 Use the **controller gps-dr** command to enter the controller mode.

Example:

```
Router> controller gps-dr
```

Step 2 Use the **dead-reckoning enable** command to enable Dead Reckoning.

Example:

```
Router(config-controller)# dead-reckoning enable
```

Step 3 Use the **dead-reckoning nmea uid** {**custom** *unique- identifier* **| hostname | serial-number**} command to enable GPS NMEA UID.

UID can be any of these:

- **custom**: user-configured string of upto 50 characters.
- **hostname**: host name of the router.
- **serial-number**: serial number of the router.

When using **serial-number** or **hostname**, the UID is automatically extracted and added to the proprietary NMEA sentence. This allows uniform CLI configuration across routers.

When using a **custom ID**, each router must be configured with a *unique-identifier* value to ensure distinct identification.

Example:

```
Router(config-controller)# dead-reckoning nmea uid custom 142
```

Example:

```
Router(config-controller)# dead-reckoning nmea uid hostname
```

Example:

```
Router(config-controller)# dead-reckoning nmea uid serial-number
```

Step 4 Use the **dead-reckoning nmea udp** *source-address destination-address destination-port* command to specify the UDP source and destination.

Example:

```
Router(config-controller)# dead-reckoning nmea udp 10.195.79.179 171.70.55.77 14013
```

Step 5 Use the **lte gps nmea filter** *hex-value* command to select the sentence types.

hex-value: HEX bitmask for NMEA message filtering (0x01:GGA, 0x02:GSA, 0x04:GSV, 0x08:RMC, 0x010:VTG, 0x20:GNS)

This example shows the HEX value if you want to send sentence type as only RMC (0x08) and GNS (0x20):

Example:

```
Router(config-controller)# dead-reckoning nmea filter 28
```

This example shows the HEX value if you want to send sentence type as only VTG (0x10) and GSV (0x04):

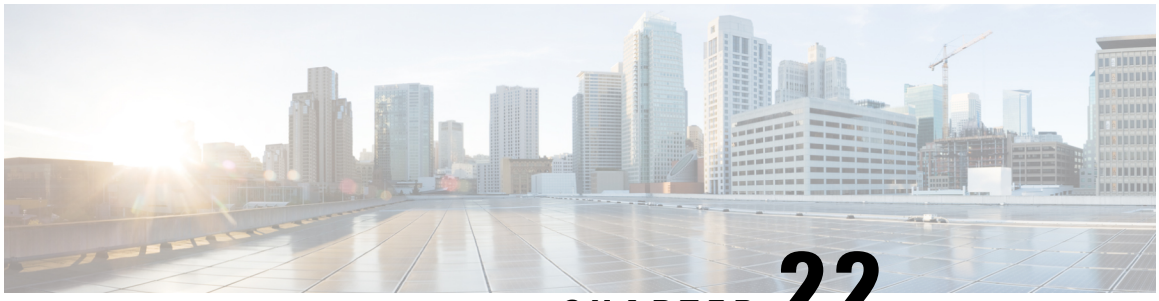
Example:

```
Router(config-controller)# dead-reckoning nmea filter 14
```

Step 6 Use the **end** command to exit the controller configuration mode and return to the privileged EXEC mode.

Example:

```
Router(config-controller)# end
```



CHAPTER 22

Information About SCADA

This section contains the following topics:.

- Supervisory Control And Data Acquisition (SCADA) Overview, on page 263
- Configuring Protocol Translation, on page 266
- Configuring the T101 Protocol Stack, on page 267
- Configuring the T104 Protocol Stack, on page 271
- Configuring the DNP3 Protocol Stacks, on page 275
- SCADA enhancement for TNB, on page 279
- Verifying Configuration, on page 280
- SCADA Debug Commands, on page 280

Supervisory Control And Data Acquisition (SCADA) Overview

SCADA refers to a control and management system employed in industries such as water management, electric power, and manufacturing. A SCADA system collects data from various types of equipment within the system and forwards that information back to a Control Center for analysis. Generally, individuals located at the Control Center monitor the activity on the SCADA system and intervene when necessary.

The Remote Terminal Unit (RTU) acts as the primary control system within a SCADA system. RTUs are configured to control specific functions within the SCADA system, which can be modified as necessary through a user interface.

On the IR1800, line is 0/2/0 or 0/2/1, same as the Async interface.

Role of the IR1800

In the network, the Control Center always serves as the master in the network when communicating with the IR1800. The IR1800 serves as a proxy master station for the Control Center when it communicates with the RTU.

The IR1800 provides protocol translation to serve as a SCADA gateway to do the following:

- Receive data from RTUs and relay configuration commands from the Control Center to RTUs.
- Receive configuration commands from the Control Center and relay RTU data to the Control Center
- Terminate incoming requests from the Control Center, when an RTU is offline

The IR1800 performs Protocol Translation for the following protocols:

- IEC 60870 T101 to/from IEC 60870 T104.
- DNP3 serial to DNP3 IP

Key Terms

The following terms are relevant when you configure the T101 and T104 protocol stacks on the IR1800:

- Channel–A channel is configured on each IR1800 serial port interface to provide a connection to a single RTU for each IP connection to a remote Control Center. Each connection transports a single T101 (RTU) or T104 (Control Center) protocol stack.
- Link Address–Refers to the device or station address.
- Link Mode (Balanced and Unbalanced)–Refers to the modes of data transfer.
  - An Unbalanced setting refers to a data transfer initiated from the master.
  - A Balanced setting can refer to either a master or slave initiated data transfer.
- Sector–Refers to a single RTU within a remote site.
- Sessions–Represents a single connection to a remote site.

The following terms are relevant when you configure the DNP3 protocol stacks on the on the IR1800:

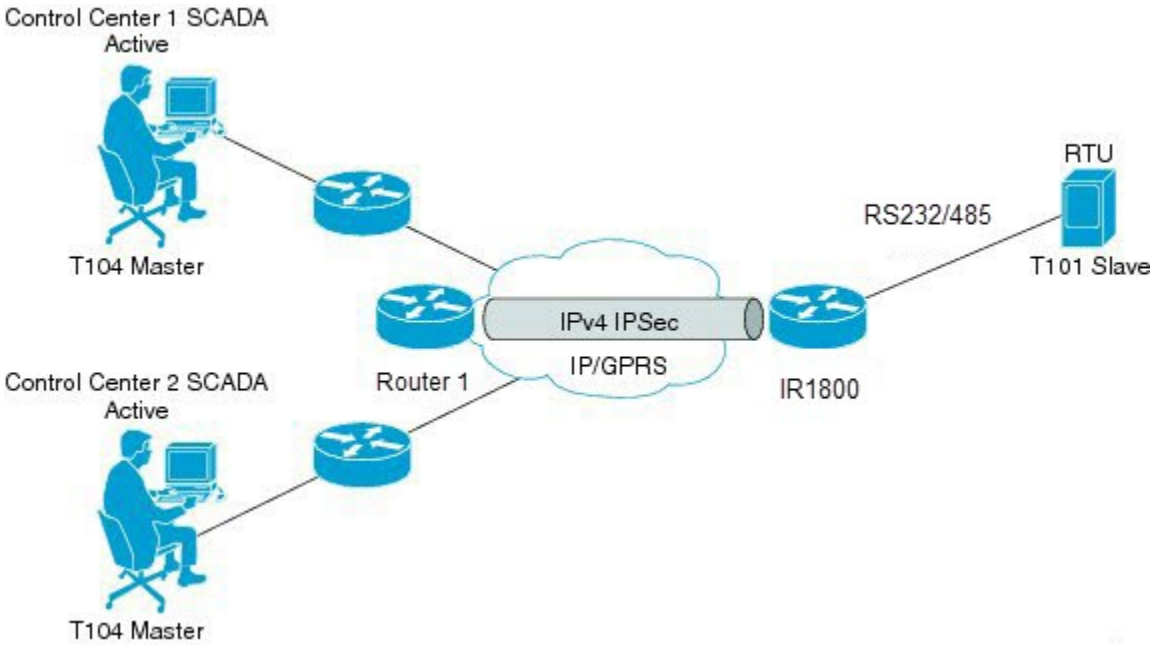
- Channel–A channel is configured on the IR1800 serial port interface to provide a connection to a single RTU for each IP connection to a remote Control Center. Each connection transports a single DNP3 serial (RTU) or DNP3 IP (Control Center) protocol stack.
- Link Address–Refers to the device or station address.
- Sessions–Represents a single connection to a remote site.

Protocol Translation Application

In Figure 30: Routers Within a SCADA System, on page 265 the IR1800 (installed within a secondary substation of the Utility Network) employs Protocol Translation to provide secure, end-to-end connectivity between Control Centers and RTUs within a SCADA System.

The IR1800 connects to the RTU (subordinate) through a RS232/RS485 connection. To protect the traffic when forwarded over public infrastructures (for example, cellular), the IR1800 forwards SCADA data from the RTU to the Control Center in the SCADA system through an IPSec tunnel (FlexVPN site-to-site or hub and spoke). The IPSec tunnel protects all traffic between the IR1800 and the Head-end aggregation router. SCADA traffic can be inspected through an IPS device positioned in the path of the SCADA traffic before it is forwarded to the proper Control Center.

Figure 30: Routers Within a SCADA System



Prerequisites

RTUs must be configured and operating in the network.

For each RTU that connects to the IR1800, you will need the following information for T101/T104:

- Channel information
  - Channel name
  - Connection type: serial
  - Link transmission procedure setting: unbalanced or balanced
  - Address field of the link (number expressed in octets)
- Session information
  - Session name
  - Size of common address of Application Service Data Unit (ASDU) (number expressed in octets)
  - Cause of transmission (COT) size (number expressed in octets)
  - Information object address (IOA) size (number expressed in octets)
- Sector information
  - Sector name
  - ASDU address, (number expressed in octets)

For each RTU that connects to the IR1800, you will need the following information for DNP3:

- Channel information
  - Channel name
  - Connection type: serial
  - Link address
- Session information
  - Session name

Guidelines and Limitations

- Each channel supports only one session.
- Each session supports only one sector.
- The object types 8, 17, 18, 19, 20, 38, 39, and 40 are not supported for IEC protocol translation.

Default Settings

DNP3 Parameters	Default
Unsolicited Response (DNP3-serial)	Not Enabled
Send Unsolicited Message (DNP3-IP)	Enabled

Configuring Protocol Translation

This section includes the following topics:

Note Before making any configuration changes to a IR1800 operating with Protocol Translation, please review the section on Starting and Stopping the Protocol Translation Engine, on page 279.

Enabling the IR1800 Serial Port and SCADA Encapsulation

Before you can enable and configure Protocol Translation on the IR1800, you must first enable the serial port on the IR1800 and enable SCADA encapsulation on that port.

Before you begin

Determine the availability of a serial port on the IR1800.

Procedure

	Command or Action	Purpose
Step 1	**configure terminal**	Enters the global configuration mode.
Step 2	**interface async** *slot/port/interface*	Enters the interface command mode for the async slot/port/interface. *slot* –value of 0 *port* –value of 2 *interface* –value of 0 or 1
Step 3	**no shutdown**	Brings up the port, administratively.
Step 4	**encapsulation scada**	Enables encapsulation on the serial port for protocol translation and other SCADA protocols.

Enable Serial Port Example

This example shows how to enable serial port 0/2/0 and how to enable encapsulation on that interface to support SCADA protocols.

```
router# configure terminal
router(config)# interface async 0/2/0
router (config-if)# no shutdown
router (config-if)# encapsulation scada
```

Configuring T101 and T104 Protocol Stacks

You can configure T101 and T104 protocol stacks, which allow end-to-end communication between Control Centers (T104) and RTUs (T101) within a SCADA system.

- Configuring the T101 Protocol Stack, on page 267
- Configuring the T104 Protocol Stack, on page 271
- Starting and Stopping the Protocol Translation Engine, on page 279

Protocol Stack Prerequisites

Ensure that you have gathered all the required configuration information.

Enable the serial port and SCADA encapsulation.

Configuring the T101 Protocol Stack

Configure the channel, session, and sector parameters for the T101 protocol stack.

Procedure

	Command or Action	Purpose
Step 1	**configure terminal**	Enters global configuration mode.
Step 2	**scada-gw protocol t101**	Enters the configuration mode for the T101 protocol.
Step 3	**channel** *channel_name*	Enters the channel configuration mode for the T101 protocol. *channel_name* –Identifies the channel on which the serial port of the IR1800 communicates to the RTU. **Note** When the entered channel name does not already exist, the router creates a new channel. Entering the **no** form of this command deletes an existing channel. However, all sessions must be deleted before you can delete a channel.
Step 4	**link-mode** *{balanced | unbalanced}*	Configures the link-mode as either balanced or unbalanced. unbalanced–Refers to a data transfer initiated from the master. balanced–Refers to either a master or slave data transfer.
Step 5	**link-addr-size {none | one | two}**	Defines the link address size in octets.
Step 6	**bind-to-interface async** *slot/port/interface*	Defines the IR1800 serial interface on which the system sends its T101 protocol traffic. *slot* –Value of 0 *port* –Value of 2 *interface* –Value of 0 or 1
Step 7	**exit**	Ends configuration of the channel and exits the channel configuration mode. Saves all settings.
Step 8	**session** *session_name*	Enters the session configuration mode and assigns a name to the session.
Step 9	**attach-to-channel** *channel_name*	Attaches the session to the channel. Enter the same channel name that you entered in Step 3 .

	Command or Action	Purpose
		channel_name –Identifies the channel.
Step 10	**common-addr-size** *{one | two | three}*	Defines the common address size in octets.
Step 11	**cot size** *{one | two | three}*	Defines the cause of transmission such as spontaneous or cyclic data schemes in octets.
Step 12	**info-obj-addr-size** *{one | two | three}*	Defines the information object element address size in octets.
Step 13	**link-addr-size** *{one | two | three}*	Defines the link address size in octets.
Step 14	**link-addr** *link_address*	Refers to the link address of the RTU. **Note** The link address entered here must match the value set on the RTU to which the serial port connects. *link_address* –Range of 0-65535.
Step 15	**exit**	Exits the session configuration mode.
Step 16	**sector** *sector_name*	Enters the sector configuration mode and assigns a name to the sector for the RTU. *sector_name* –Identifies the sector.
Step 17	**attach-to-session** *session_name*	Attaches the RTU sector to the session. Enter the same session name that you entered in Step 9 . *session_name*- Identifies the session.
Step 18	**asdu-addr** *asdu_address*	Refers to the ASDU structure address of the RTU.
Step 19	**exit**	Exits the sector configuration mode.
Step 20	**exit**	Exits the protocol configuration mode.

T101 Protocol Stack Example

This example shows how to configure the parameters for the T101 protocol stack for *RTU_10* .

```
router# configure terminal
router(config)# scada-gw protocol t101
router(config-t101)# channel rtu_channel

router(config-t101-channel)# link-mode unbalanced
router(config-t101-channel)# link-addr-size
 one
router(config-t101-channel)# bind-to-interface async 0/2/0
router(config-t101-channel)# exit
router(config-t101)# session rtu_session
```

```
router(config-t101-session)# attach-to-channel rtu_channel
router(config-t101-session)# common-addr-size two
router(config-t101-session)# cot-size one
router(config-t101-session)# info-obj-addr-size two
router(config-t101-session)# link-addr 3
router(config-t101-session)# exit
router(config-t101)# sector rtu_sector
router(config-t101-sector)# attach-to-session rtu_session
router(config-t101-sector)# asdu-addr 3
router(config-t101-sector)# exit
router(config-t101)# exit
router(config)#
```

T101 Configuration Example

The following example shows how to configure the serial port interface for T101 connection, configure T101 and T104 protocol stacks, and starts the Protocol Translation Engine on the IR1800.

```
router# configure terminal
router(config)# interface async 0/2/0
router (config-if)# no shutdown
router (config-if)# encapsulation scada
router (config-if)# exit
router(config)# scada-gw protocol t101
router(config-t101)# channel rtu_channel

router(config-t101-channel)# link-mode unbalanced
router(config-t101-channel)# link-addr-size one
router(config-t101-channel)# bind-to-interface async 0/2/0
router(config-t101-channel)# exit
router(config-t101)# session rtu_session
router(config-t101-session)# attach-to-channel rtu_channel
router(config-t101-session)# common-addr-size two
router(config-t101-session)# cot-size one
router(config-t101-session)# info-obj-addr-size two
router(config-t101-session)# link-addr 3
router(config-t101-session)# exit
router(config-t101)# sector rtu_sector
router(config-t101-sector)# attach-to-session rtu_session
router(config-t101-sector)# asdu-addr 3
router(config-t101-sector)# exit
router(config-t101)# exit
router(config)# scada-gw protocol t104
router(config-t104)# channel cc_master1
router(config-t104-channel)# k-value 12
router(config-t104-channel)# w-value 8
router(config-t104-channel)# t0-timeout 30
router(config-t104-channel)# t1-timeout 15
router(config-t104-channel)# t2-timeout 10
router(config-t104-channel)# t3-timeout 30
router(config-t104-channel)# tcp-connection 0 local-port 2050 remote-ip any
router(config-t104-channel)# tcp-connection 1 local-port 2051 remote-ip any
router(config-t104-channel)# exit
router(config-t104)# session cc_master1
router(config-t104-session)# attach-to-channel cc_master1

router(config-t104-session)# exit
router(config-t104)# sector cc_master1-sector
router(config-t104-sector)# attach-to-session cc_master1
router(config-t104-sector)# asdu-adr 3
router(config-t104-sector)# map-to-sector rtu_sector
```

```
router(config-t104)# exit
router(config-t104)# session cc_master2
router(config-t104-session)# attach-to-channel cc_master2

router(config-t104-session)# exit
router(config-t104)# sector cc_master2-sector
router(config-t104-sector)# attach-to-session cc_master2
router(config-t104-sector)# asdu-adr 3
router(config-t104-sector)# map-to-sector rtu_sector
router(config-t104-sector)# exit
router(config-t104)# exit
router(config)# scada-gw enable
```

This example configures end-to-end communication between Control Centers and RTUs within a SCADA system using the DNP3 protocol stacks and starts the Protocol Translation Engine on the IR1800:

```
router# configure terminal
router(config)# interface async 0/2/0
router (config-if)# no shutdown
router (config-if)# encapsulation scada
router (config-if)# exit
router(config)# scada-gw protocol dnp3-serial
router(config-dnp3s)# channel rtu_channel
router(config-dnp3s-channel)# bind-to-interface async 0/2/0
router(config-dnp3s-channel)# link-addr source 3
router(config-dnp3s-channel)# unsolicited-response enable
router(config-dnp3s-channel)# exit
router(config-dnp3s)# session rtu_session
router(config-dnp3s-session)# attach-to-channel rtu_channel
router(config-dnp3s-session)# link-addr dest 3
router(config-dnp3s-session)# exit
router(config-dnp3s)# exit
router(config)# scada-gw protocol dnp3-ip
router(config-dnp3n)# channel cc_channel
router(config-dnp3n-channel)# link-addr dest 3
router(config-dnp3n-channel)# tcp-connection local-port default remote-ip any
router(config-dnp3n-channel)# exit
router(config-dnp3n)# session cc_session
router(config-dnp3n-session)# attach-to-channel cc_channel
router(config-dnp3n-session)# link-addr source 3
router(config-dnp3n-session)# map-to-session rtu_session
router(config-dnp3n)# exit
router(config)# exit
router(config)# scada-gw enable
```

Configuring the T104 Protocol Stack

Follow the steps below for each Control Center that you want to connect to over a T104 protocol.

Before you begin

Ensure that you have gathered all the required configuration information. (See Prerequisites, on page 265)

Enable the serial port and SCADA encapsulation. (See Enabling the IR1800 Serial Port and SCADA Encapsulation, on page 266)

Procedure

	Command or Action	Purpose
Step 1	**configure terminal**	Enters configuration mode.
Step 2	**scada-gw protocol t104**	Enters the configuration mode for the T104 protocol.
Step 3	**channel** *channel_name*	Enters the channel configuration mode for the T104 protocol. *channel_name* –Identifies the channel on which the router communicates with the Control Center. **Note** When the entered channel name does not already exist, the router creates a new channel. Entering the **no** form of this command deletes an existing channel. However, all sessions must be deleted before you can delete a channel.
Step 4	**k-value** *value*	Sets the maximum number of outstanding Application Protocol Data Units (APDUs) for the channel. **Note** An APDU incorporates the ASDU and a control header. *value* –Range of values from 1 to 32767. Default value is 12 APDUs.
Step 5	**w-value** *value*	Sets the maximum number of APDUs for the channel. *value* –Range of values from 1 to 32767. Default value is 8 APDUs.
Step 6	**t0-timeout** *value*	Defines the t0-timeout value for connection establishment of the T104 channel.
Step 7	**t1-timeout** *value*	Defines the t1-timeout value for send or test APDUs on the T104 channel.
Step 8	**t2-timeout** *value*	Defines the t2-timeout value for acknowledgements when the router receives no data message. **Note** The t2 value must always be set to a lower value than the t1 value on the T104 channel.

	Command or Action	Purpose
Step 9	**t3-timeout** *value*	Defines the t3-timeout value for sending s-frames in case of a long idle state on the T104 channel. **Note** The t3 value must always be set to a higher value than the t1 value on the T104 channel.
Step 10	**tcp-connection {0|1} local-port** {*port_number* | **default}** **remote-ip** {*A.B.C.D* | *A.B.C.D/LEN* | **any**} [**vrf** *WORD*]	In a configuration where there are redundant Control Centers, sets the connection value for the secondary Control Center as defined on the primary Control Center. *port-number* –value between 2000 and 65535. default–value of 2404. *A.B.C.D* –single host. *A.B.C.D/nn* –subnet A.B.C.D/LEN. any–any remote hosts 0.0.0.0/0. WORD–VRF name.
Step 11	**exit**	Exits the channel configuration mode.
Step 12	**session** *session_name*	Enters the session configuration mode and assigns a name to the session. *session_name* –Use the same name that you assigned to the channel in Step 3 .
Step 13	**attach-to-channel** *channel_name*	Defines the name of the channel that transports the session traffic.
Step 14	**exit**	Exits the session configuration mode.
Step 15	**sector** *sector_name*	Enters the sector configuration mode and assigns a name to the sector for the Control Center.
Step 16	**attach-to-session** *session_name*	Attaches the Control Center sector to the channel. *session_name* –Use the same name that you assigned to the channel in Step 3 .
Step 17	**asdu-addr** *asdu_address*	Refers to the ASDU structure address. Value entered here must match the ASDU value on the RTU. *asdu_address* –*asdu_address* –Value of 1 or 2.

	Command or Action	Purpose
Step 18	**map-to-sector** *sector_name*	Maps the Control Center (T104) sector to the RTU (T101) sector.
Step 19	Return to Step 1 .	Repeat all steps in this section for each Control Center active in the network.

Configure T104 Protocol Stack Example

This example shows how to configure the parameters for the T104 protocol stack on *Control Center 1* and *Control Center 2*, both of which are configured as *masters* , and how to map the T104 sector to the T101 sector.

To configure Control Center 1 (*cc_master1*), enter the following commands.

```
router# configure terminal
router(config)# scada-gw protocol t104
router(config-t104)# channel cc_master1
router(config-t104-channel)# k-value 12
router(config-t104-channel)# w-value 8
router(config-t104-channel)# t0-timeout 30
router(config-t104-channel)# t1-timeout 15
router(config-t104-channel)# t2-timeout 10
router(config-t104-channel)# t3-timeout 30
router(config-t104-channel)# tcp-connection 0 local-port 2050 remote-ip 209.165.200.225
router(config-t104-channel)# tcp-connection 1 local-port 2051 remote-ip 209.165.201.25
router(config-t104-channel)# exit
router(config-t104)# session
 cc_master1
router(config-t104-session)# attach-to-channel cc_master1

router(config-t104-session)# exit
router(config-t104)# sector cc_master1-sector
router(config-t104-sector)# attach-to-session cc_master1
router(config-t104-sector)# asdu-adr 3
router(config-t104-sector)# map-to-sector rtu_sector
router(config-t104)# exit
router(config)#
```

To configure Control Center 2 (*cc_master2*), enter the following commands.

```
router(config)# scada-gw protocol t104
router(config-t104)# channel cc_master2
router(config-t104-channel)# k-value 12
router(config-t104-channel)# w-value 8
router(config-t104-channel)# t0-timeout 30
router(config-t104-channel)# t1-timeout 15
router(config-t104-channel)# t2-timeout 10
router(config-t104-channel)# t3-timeout 30
router(config-t104-channel)# tcp-connection 0 local-port 2060 remote-ip 209.165.201.237
router(config-t104-channel)# tcp-connection 1 local-port 2061 remote-ip 209.165.200.27
router(config-t104-channel)# exit
router(config-t104)# session cc_master2
router(config-t104-session)# attach-to-channel cc_master2

router(config-t104-session)# exit
router(config-t104)# sector cc_master2-sector
router(config-t104-sector)# attach-to-session cc_master2
router(config-t104-sector)# asdu-adr 3
```

```
router(config-t104-sector)# map-to-sector rtu_sector
router(config-t104-sector)# exit
router(config-t104)# exit
router(config)#
```

Configuring the DNP3 Protocol Stacks

You can configure the DNP3 serial and DNP3 IP protocol stacks, which allow end-to-end communication between Control Centers and RTUs within a SCADA system.

Configuring DNP3 Serial

Configure the channel and session parameters for the DNP serial communication with an RTU.

Procedure

	Command or Action	Purpose
Step 1	**configure terminal**	Enters global configuration mode.
Step 2	**scada-gw protocol dnp3-serial**	Enters configuration mode for the DNP3 serial protocol.
Step 3	**channel** *channel_name*	Enters channel configuration mode for the DNP3 serial protocol. *channel_name* –Identifies the channel on which the router serial port communicates to the RTU. **Note** When the entered channel name does not already exist, the router creates a new channel Entering the **no** form of this command deletes an existing channel. However, all sessions must be deleted before you can delete a channel.
Step 4	**bind-to-interface async0/2/0**	Defines the router async interface on which the system sends its DNP3 protocol traffic.
Step 5	**link-addr source** *source_address*	Refers to the link address of the master. *source_address* –Range of values from 1 to 65535.
Step 6	**unsolicited-response enable**	(Optional) Allows unsolicited responses. Entering the **no** form of this command disables unsolicited responses. The default is disabled.

	Command or Action	Purpose
Step 7	**exit**	Ends configuration of the channel and exits channel configuration mode. Saves all settings.
Step 8	**session** *session_name*	Enters session configuration mode and assigns a name to the session. **Note** When the entered session name does not already exist, the router creates a new session. Entering the **no** form of this command deletes an existing session.
Step 9	**attach-to-channel** *channel_name*	Attaches the session to the channel. **Note** Enter the same channel name that you entered in Step 3 above. *channel_name* –Identifies the channel.
Step 10	**link-addr dest** *destination_address*	Refers to the link address of the slave. *destination_address* –Range of values from 1 to 65535.
Step 11	**exit**	Exits session configuration mode.
Step 12	**exit**	Exits protocol configuration mode.

DPN3-Serial Protocol Stack Example

This example shows how to configure the parameters for the DPN3-serial protocol stack:

```
router# configure terminal
router(config)# scada-gw protocol dnp3-serial
router(config-dnp3s)# channel rtu_channel
router(config-dnp3s-channel)# bind-to-interface async 0/2/0
router(config-dnp3s-channel)# link-addr source 3
router(config-dnp3s-channel)# unsolicited-response enable
router(config-dnp3s-channel)# exit
router(config-dnp3s)# session rtu_session
router(config-dnp3s-session)# attach-to-channel rtu_channel
router(config-dnp3s-session)# link-addr dest 3
router(config-dnp3s-session)# exit
router(config-dnp3s)# exit
router(config)#
```

Configuring DNP3 IP

Follow the steps below for the Control Center that you want to connect to over DNP3 IP. For redundancy, you can create multiple connections that share the same session configuration under the same session.

Procedure

	Command or Action	Purpose
Step 1	**configure terminal**	Enters configuration mode.
Step 2	**scada-gw protocol dnp3-ip**	Enters configuration mode for the DNP-IP protocol.
Step 3	**channel** *channel_name*	Enters channel configuration mode for the DNP-IP protocol. *channel_name* –Identifies the channel on which the router communicates with the Control Center. **Note** When the entered channel name does not already exist, the router creates a new channel. Entering the **no** form of this command deletes an existing channel. However, all sessions must be deleted before you can delete a channel.
Step 4	**link-addr dest** *destination_address*	Refers to the link address of the master. *destination_address* –Range of values from 1 to 65535.
Step 5	**send-unsolicited-msg enable**	(Optional) Allow unsolicited messages. The default is enabled.
Step 6	**tcp-connection local-port** [**default** | *local_port*] **remote-ip** [**any** | *remote_ip* | *remote_subnet*]	Configures the local port number and remote IP address for the TCP connection: • **default** –20000. • *local_port* –Range of values from 2000 to 65535. • any–Any remote hosts 0.0.0.0/0 • *remote_ip* –Single host: A.B.C.D • *remote_subnet* –Subnet: A.B.C.D/LEN If remote_subnet is specified, when two channels have the same local ports, the remote subnets cannot overlap each other. **Note** Every <local-port, remote-ip> must be unique per channel. If remote_subnet is specified,

	Command or Action	Purpose
		when two channels have the same local ports, the remote subnets cannot overlap each other.
Step 7	**exit**	Exits channel configuration mode.
Step 8	**session** *session_name*	Enters session configuration mode and assigns a name to the session. **Note** When the entered session name does not already exist, the router creates a new session. Entering the **no** form of this command deletes an existing session.
Step 9	**attach-to-channel** *channel_name*	Attaches the session to the channel. Enter the same channel name that you entered in Step 3. *channel_name* –Identifies the channel.
Step 10	**link-addr source** *source_address*	Refers to the link address of the slave. *source_address* –Value of 1-65535.
Step 11	**map-to-session** *session_name*	Maps the dnp3-ip session to an existing dnp3-serial session. **Note** One dnp3-ip session can be mapped to only one dnp3-serial session.
Step 12	**exit**	Exits session configuration mode.
Step 13	**exit**	Exits protocol configuration mode.

DNP3 IP Parameters Example

This example shows how to configure the DNP3 IP parameters:

```
router# configure terminal
router(config)# scada-gw protocol dnp3-ip
router(config-dnp3n)# channel cc_channel
router(config-dnp3n-channel)# link-addr dest 3
router(config-dnp3n-channel)# tcp-connection local-port default remote-ip any
router(config-dnp3n-channel)# exit
router(config-dnp3n)# session cc_session
router(config-dnp3n-session)# attach-to-channel cc_channel
router(config-dnp3n-session)# link-addr source 4
router(config-dnp3n-session)# map-to-session rtu_session
router(config-dnp3n)# exit
router(config)# exit
```

Starting and Stopping the Protocol Translation Engine

Before starting the Protocol Translation Engine on the router for the first time, make sure you complete the following items:

Enabling the IR1800 Serial Port and SCADA Encapsulation, on page 266

Configuring T101 and T104 Protocol Stacks, on page 267

You must start the Protocol Translation Engine to use Protocol Translation on the IR1800.

Starting– After enabling SCADA encapsulation on the IR1800 serial port and configuring the T101 and T104 protocols on the IR1800, you can start the Protocol Translation Engine.

Stopping– Before you can make any configuration changes to Protocol Translation on the IR1800 with an active Protocol Translation Engine, you must stop the engine.

Procedure

	Command or Action	Purpose
Step 1	**configure terminal**	Enters global configuration mode.
Step 2	**scada-gw enable**	Starts (**scada-gw enable**) or stops (**no scada-gw enable**) the Protocol Translation Engine on the IR1800.

Start Protocol Translation Engine Example

To start the protocol translation engine on the router, enter the following commands:

```
router# configure terminal
router(config)# scada-gw enable
```

To stop the protocol translation engine on the router, enter the following commands:

```
router# configure terminal
router(config)# no
 scada-gw enable
```

SCADA enhancement for TNB

This enhancement provides compatibility with TNB's WG RTUs, including:

- TNB RTUs require Reset-Link message to be sent out along with Link-Status message to ensure correct initialization of the serial. The feature can be selectively turned on using the new configuration CLI **scada-gw protocol force reset-link.**
- When clock passthru is enabled and if the router hasn't received the timestamp from the DNP3-IP master, the router's hardware time will be sent downstream to RTU. Upon receiving a new timestamp from DNP3-IP master, the router will start sending the new timestamp sourced from DNP3-IP master to RTU.
- The number of bufferable DNP3 events in memory will be increased from 600 to 10000.

- The **scada-gw protocol interlock** command will be supported for DNP3. Previously, the support only existed for T101/T104. With this new enhancement, the router will disconnect Serial link if the DNP3-IP master is down or unreachable. Similarly, when the Serial link to RTU is down, the TCP connection to DNP3-IP master will be untethered.
- Custom "requests" will be automatically ordered based on priority so that the user can specify them in any order that they would like to.

Verifying Configuration

Command	Purpose
show running-config	Shows the configuration of the router including active features and their settings.
show scada database	Displays details on the SCADA database.
show scada statistics	Shows statistics for the SCADA gateway, including the number of messages sent and received, timeouts, and errors.
show scada tcp	Displays TCP connections associated with the SCADA gateway.

This example shows the output from the show scada tcp and show scada statistics commands:

```
router# show scada tcp
DNP3 network channel [test]: 4 max simultaneous connections
conn: local-ip: 3.3.3.21        local-port 20000        remote-ip 3.3.3.15      data-socket
 1
Total:
  1 current client connections
  0 total closed connections
router# show scada statistics
DNP3 network Channel [test]:
  5 messages sent, 2 messages received
  0 timeouts, 0 aborts, 0 rejections
  2 protocol errors, 2 link errors, 0 address errors
DNP3 serial Channel [test]:
  152 messages sent, 152 messages received
  1 timeouts, 0 aborts, 0 rejections
  0 protocol errors, 0 link errors, 0 address errors
```

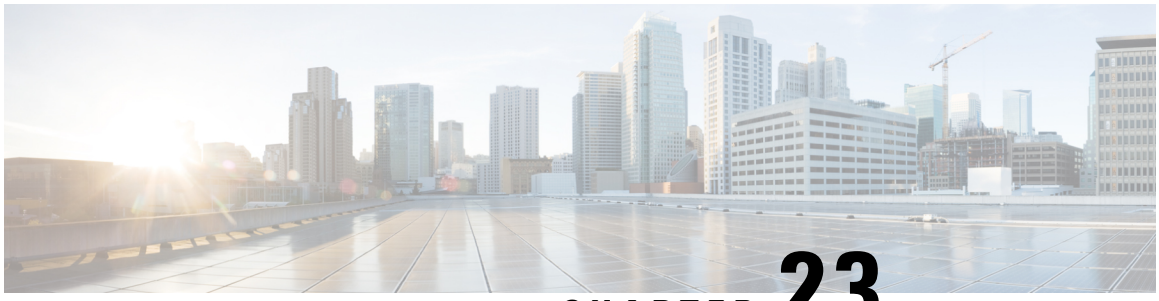
SCADA Debug Commands

This section lists some debug commands that are helpful when troubleshooting.

Table 21: SCADA Function Level Debug Commands

Command	Purpose
debug scada function config	Configuration trace

Command	Purpose
debug scada function control	Control trace
debug scada function file	File trace
debug scada function freeze	Freeze trace
debug scada function physical	Physical trace
debug scada function poll	Poll trace
debug scada function stack	Stack trace
debug scada function umode	Umode trace



CHAPTER 23

Raw Socket Transport

This section contains the following topics:

- Raw Socket Transport Overview, on page 283
- Information About Raw Socket Transport, on page 283
- Prerequisites, on page 286
- Guidelines and Limitations, on page 286
- Default Settings, on page 286
- Configuring Raw Socket Transport, on page 286
- Rawsocket Keepalive Configuration CLI, on page 292
- Verifying Configuration, on page 293
- Raw Socket Transport Configuration Examples, on page 293

Raw Socket Transport Overview

Raw Socket Transport transports streams of characters from one serial interface to another over an IP network for utility applications.

This document describes Raw Socket Transport for the IR1800 and provides a reference section describing the Raw Socket Transport commands.

Information About Raw Socket Transport

Raw Socket is a method for transporting serial data through an IP network. The feature can be used to transport Supervisory Control and Data Acquisition (SCADA) data from Remote Terminal Units (RTUs). This method is an alternative to the Block Serial Tunnel (BSTUN) protocol.

Raw Socket Transport supports TCP or UDP as the transport protocol. An interface can be configured to use either protocol but not both at the same time. TCP transport is suitable for applications such as control applications that require acknowledged and sequenced delivery of data. For latency-sensitive applications such as line SEL relays, UDP transport provides faster transport of serial data than TCP.

Raw Socket Transport supports the following for the asynchronous serial interface:

- TCP as the transport protocol, with built-in auto TCP connection retry mechanism.
- Up to 32 TCP sessions.

- Interface configuration as a server, client, or a combination of both.
- One server interface, but multiple clients.
- VRF-awareness, which enables the router to send Raw Socket Transport traffic to a server host connected through a Virtual Private Network (VPN) Virtual Routing and Forwarding (VRF) interface.

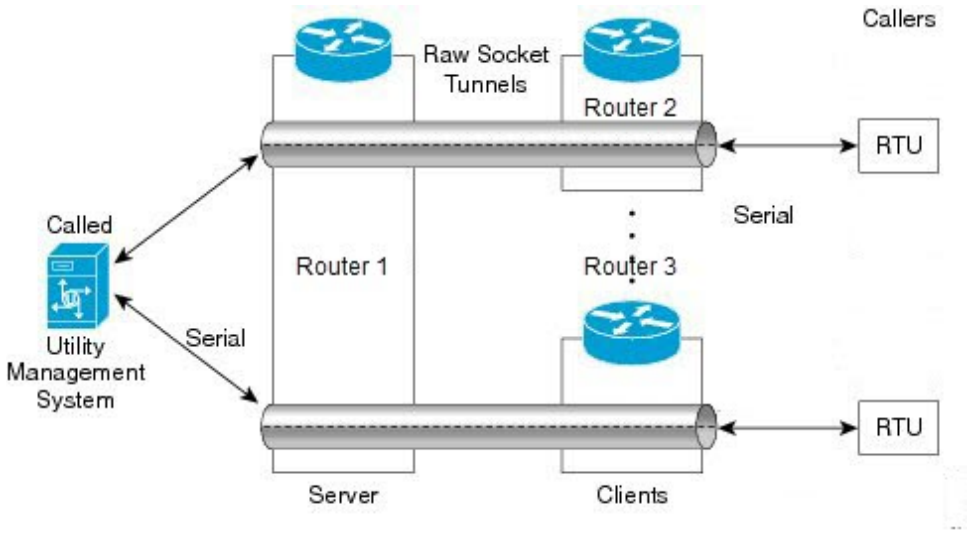
This section includes the following topics:

TCP Transport

TCP Raw Socket transport uses a client-server model. At most one server and multiple clients can be configured on a single asynchronous serial line. In client mode, the IR1800 can initiate up to 32 TCP sessions to Raw Socket servers, which can be other IR1800 routers or third-party devices.

The following figure shows a sample Raw Socket TCP configuration. In this example, serial data is transferred between RTUs and a utility management system across an IP network that includes several IR1800 routers. One IR1800 router (Router 1) acts as a Raw Socket server, listening for TCP connection requests from the other IR1800 routers (Router 2 and Router 3), which are configured as Raw Socket clients.

A Raw Socket client receives streams of serial data from the RTUs and accumulates this data in its buffer, then places the data into packets, based on user-specified packetization criteria. The Raw Socket client initiates a TCP connection with the Raw Socket server and sends the packetized data across the IP network to the Raw Socket server, which retrieves the serial data from the packets and sends it to the serial interface, and on to the utility management system.



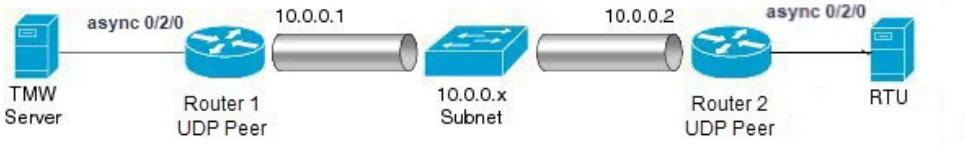
Note When you configure the serial link interface on the router as a server, the interface's peer is the serial link interface on the client router and vice versa.

UDP Transport

UDP transport uses a peer-to-peer model. Multiple UDP connections can be configured on an asynchronous serial line.

The following figure shows a sample Raw Socket UDP configuration. In this example, serial data is transferred between RTUs and a utility management system across an IP network that includes two routers (Router 1 which is an IR1800 and Router 2 which is an IR807) that are configured as Raw Socket UDP peers.

In this example, the Raw Socket UDP peer receives streams of serial data from the RTUs and accumulates this data in its buffer, then places the data into packets, based on user-specified packetization criteria. The Raw Socket UDP peer sends the packetized data across the IP network to the Raw Socket peer at the other end, which retrieves the serial data from the packets and sends it to the serial interface, and on to the utility management system.



Serial Data Processing

When the default serial protocol, Asynchronous Communication Protocol, is used, the streams of serial data received by a Raw Socket peer can be packetized based on the following criteria:

- **Packet length** – You can specify a packet length that triggers the IR1800 to transmit the serial data to the peer. Once the IR1800 collects this much data in its buffer, it packetizes the accumulated data and forwards it to the Raw Socket peer.
- **Packet-timer value** – The packet timer specifies the amount of time the IR1800 waits to receive the next character in a stream. If a character is not received by the time the packet timer expires, the data the IR1800 has accumulated in its buffer is packetized and forwarded to the Raw Socket peer.
- **Special character** – You can specify a character that will trigger the IR1800 to packetize the data accumulated in its buffer and send it to the Raw Socket peer. When the special character (for example, a CR/LF) is received, the IR1800 packetizes the accumulated data and sends it to the Raw Socket peer.

See the "Configuring Common Raw Socket Line Options" procedure on page 6 for information about configuring the processing options.

VRF-Aware Raw Socket

The VRF-aware Raw Socket Transport feature enables you to isolate Raw Socket traffic using a VRF for efficient management and control of serial data. After configuring a VRF, you can associate the serial interface configured for Raw Socket Transport with the VRF. See the Raw Socket VRF, on page 295 for a configuration example.

Prerequisites

Determine how you want Raw Socket traffic transported in your network, including the network devices and interfaces to use, how the router packetizes the serial data, and whether to use VRF.

Guidelines and Limitations

The guidelines and limitations are as given:

- Typically, UDP traffic is blocked by firewalls in the network. If the network has such firewalls, make sure to configure pinholes to allow the raw socket UDP traffic.
- The `access-class <acl> in` command is unsupported for serial raw-socket line configurations on IR platforms using IOS-XE.

Default Settings

Feature	Default Setting
Raw Socket Transport	Disabled.
Packet length	No packet length is configured.
Serial Protocol	Asynchronous Communication Protocol
Packet timeout	15 ms.
Special character	No special character is configured.
Raw Socket mode	Best-effort mode is off, not supported on the IR1801.
TCP idle timeout	5 minutes.

Configuring Raw Socket Transport

This section includes the following topics:

Enabling Raw Socket Transport on the Serial Interface

To enable Raw Socket Transport on the IR1800 router, you must first enable an asynchronous serial port and enable Raw Socket TCP or UDP encapsulation for that port.

Procedure

	Command or Action	Purpose
Step 1	**configure terminal**	Enters global configuration mode.
Step 2	**interface async0** */slot /port*	Enters the interface command mode for the async slot/port.
Step 3	**no ip address**	Disables IP processing on the interface.
Step 4	Do one of the following: • **encapsulation raw-tcp** • **encapsulation raw-udp**	Enables Raw Socket TCP encapsulation or UDP encapsulation for the serial port.

Enable Serial Port Example

This example shows how to enable serial port 0/2/0 and how to enable Raw Socket TCP encapsulation on that port.

```
router# configure terminal
router(config)# interface async0/2/0
router(config-if)# no ip address
router(config-if)# encapsulation raw-tcp
router(config-if)# exit
```

Configuring Common Raw Socket Line Options

You can configure options common to all connections on a line. The common options apply to both TCP and UDP.

Before you begin

Enable Raw Socket Transport as described in Enabling Raw Socket Transport on the Serial Interface, on page 286.

Procedure

	Command or Action	Purpose
Step 1	**configure terminal**	Enters global configuration mode.
Step 2	**line 0** */slot /port*	Enters line command mode for the serial slot/port.
Step 3	**raw-socket packet-length** *length*	Specifies the packet size that triggers the IR1800 to transmit the data to the peer. When the IR1800 accumulates this much data in its buffer, it packetizes the data and forwards it to the Raw Socket peer. *length*— 2 to 1400 bytes.

	Command or Action	Purpose
		By default, the packet-length trigger is disabled.
Step 4	**raw-socket packet-timer** *timeout*	Specifies the maximum time in milliseconds the IR1800 waits to receive the next character in a stream. If a character is not received by the time the packet-timer expires, the accumulated data is packetized and forwarded to the Raw Socket peer. *timeout* —3 to 1000 ms. The default is 15 ms.
Step 5	**raw-socket spec-char** *ascii_char*	Specifies a character that will trigger the IR1800 to packetize the data accumulated in its buffer and send it to the Raw Socket peer. *ascii_char*— 0 to 255. By default, the special character trigger is disabled.

What to do next

Use the **no** form of these commands to return to the default values.

Configuring Common Raw Socket Line Options Example

```
router# configure terminal
router(config)# line 0/2/0
router(config-line)# raw-socket packet-length 32
router(config-line)# raw-socket packet-timer 500
router(config-line)# raw-socket special-char 3
```

Configuring Raw Socket TCP

After enabling Raw Socket TCP encapsulation, you configure the TCP server and/or clients.

Configuring the Raw Socket TCP Server

Before you begin

Enable a serial port and Raw Socket TCP encapsulation for that port, as described in Enabling Raw Socket Transport on the Serial Interface, on page 286.

Procedure

	Command or Action	Purpose
Step 1	**configure terminal**	Enters configuration mode.

	Command or Action	Purpose
Step 2	**line 0** */slot /port*	Enters line command mode for the serial slot/port.
Step 3	**raw-socket tcp server** *port* [*ip_address*]	Starts the Raw Socket Transport TCP server for an asynchronous line interface. In Raw Socket server mode, the IR1800 listens for incoming connection requests from Raw Socket clients. *port* – Port number the server listens on. *ip_address* – (Optional) Local IP address on which the server listens for connection requests.
Step 4	**raw-socket tcp idle-timeout** *session_timeout*	Sets the Raw Socket Transport TCP session timeout for the asynchronous line interface. If no data is transferred between the client and server over this interval, then the TCP session closes. The client then automatically attempts to reestablish the TCP session with the server. This timeout setting applies to all Raw Socket Transport TCP sessions under this particular line. *session_timeout* –Currently configured session idle timeout in minutes. The default is 5 minutes.

What to do next

To remove a Raw Socket TCP server, use the **no raw-socket tcp server** command.

Configuring Common Raw Socket TCP Server Example

This example shows how to configure a Raw Socket TCP server for an asynchronous serial line. The TCP server listens for TCP client connection requests on local port 4000 and local IP address 10.0.0.1. If no data is exchanged between the Raw Socket TCP server and one of the TCP clients for 10 minutes, then the TCP session closes, and the Raw Socket client attempts to reestablish the session with the Raw Socket server.

```
router# configure terminal

router(config)# line 0/2/0
router(config-line)# raw-socket tcp server 4000 10.0.0.1
router(config-line)# raw-socket tcp idle-timeout 10
router(config-line)# exit
router(config)#
```

Configuring the Raw Socket TCP Client

Before you begin

Enable a serial port and Raw Socket TCP encapsulation for that port, as described in Enabling Raw Socket Transport on the Serial Interface, on page 286.

Procedure

	Command or Action	Purpose
Step 1	**configure terminal**	Enters configuration mode.
Step 2	**line 0** */slot /port*	Enters line command mode for the serial slot/port.
Step 3	**raw-socket tcp client** *dest_ip_address dest_port [local_ip_address] [local_port]*	Specifies settings for Raw Socket Transport TCP client sessions. *dest_ip_address* – Destination IP address of the remote Raw Socket server. *dest_port* – Destination port number to use for the TCP connection to the remote server. *local_ip_address* – (Optional) Local IP address that the client can also bind to. *local_port* – (Optional) Local port number that the client can also bind to.
Step 4	**raw-socket tcp idle-timeout** *session_timeout*	Sets the Raw Socket Transport TCP session timeout for the asynchronous line interface. If no data is transferred between the client and server over this interval, then the TCP session is closed. The client then automatically attempts to reestablish the TCP session with the server. This timeout setting applies to all Raw Socket Transport TCP sessions under this particular line. *session_timeout* –Currently configured session idle timeout in minutes. The default is 5 minutes.
Step 5	**tcp keepalive** *interval*	Sets the Raw Socket Transport TCP session keepalive interval for the asynchronous line interface. The router sends keepalive messages based on the configured interval. You may need to configure this interval, for example, when sending raw TCP traffic over a cellular interface. *interval* – Currently configured keepalive interval in seconds. Range is 1-864000 seconds. The default is 1 second.

What to do next

To remove a Raw Socket TCP client, use the **no raw-socket tcp client** command.

Raw Socket TCP Client Example

This example shows how to configure a Raw Socket TCP client for an asynchronous serial line. The IR1800 (router), serving as a Raw Socket client, initiates TCP sessions with a Raw Socket server and forwards packetized serial data to it. The router collects streams of serial data in its buffer; when it accumulates 827 bytes in its buffer, the router packetizes the data and forwards it to the Raw Socket server. If the router and the Raw Socket server do not exchange any data for 10 minutes, then the TCP session with the Raw Socket server closes, and the router attempts to reestablish the session with the Raw Socket server.

```
router# configure terminal
router(config)# line 0/2/0
router(config-line)# raw-socket tcp client 10.0.0.1 4000
router(config-line)# raw-socket packet-length 827
router(config-line)# raw-socket tcp idle-timeout 10
router(config-line)# exit
router(config)#
```

Raw socket feature enhancement

A raw socket feature enhancement is a network configuration capability that:

- allows the user to input the maximum number of retries available to the write socket,
- provides a range from 1 to 1000 retries with a default of 10 retries, and
- includes a new CLI command **raw-socket tcp max-retries <1-1000>** where <1-1000> is the maximum number of retries.

Configuring a Raw Socket UDP Peer-to-Peer Connection

After enabling Raw Socket UDP encapsulation and the common line options, you configure the Raw Socket UDP peer-to-peer connection. The local port on one end of the connection should be the destination port on the other end.

Before you begin

Enable a serial port and Raw Socket UDP encapsulation for that port, as described in Enabling Raw Socket Transport on the Serial Interface, on page 286.

Procedure

	Command or Action	Purpose
Step 1	**configure terminal**	Enters configuration mode.
Step 2	**line 0** */slot /port*	Enters line command mode for the serial slot/port.
Step 3	**raw-socket udp connection** *dest_ip_address dest_port local_port [local_ip_address]*	Specifies settings for Raw Socket Transport UDP connections. *dest_ip_address* – Destination IP address to use for the UDP connection.

	Command or Action	Purpose
		dest_port – Destination port number to use for the UDP connection. *local_port* – Local port number for the UDP connection. *local_ip_address* – (Optional) Local IP address for the UDP connection.

What to do next

To remove a Raw Socket UDP connection, use the **no raw-socket udp connection** command.

Raw Socket UDP Connection Example

This example shows how to configure a Raw Socket UDP connection between router A (local IP address 192.168.0.8) and router B (local IP address 192.168.0.2).

Router A

```
router# configure terminal
router(config)# line 0/2/0
router(config-line)# raw-socket udp connection 192.168.0.2 5000 7000
router(config-line)# exit
router(config)#
```

Router B

```
router# configure terminal
router(config)# line 0/2/0
router(config-line)# raw-socket udp connection 192.168.0.8 7000 5000
router(config-line)# exit
router(config)#
```

Rawsocket Keepalive Configuration CLI

Rawsocket keepalive for async interfaces is a feature that existed in classic IOS platforms. As part of 17.10.1a, the feature will be extended to IOS-XE based platforms. A new CLI with the following syntax will be added under rawsocket.

```
Router(config-line)#raw-socket tcp keepalive interval
```

CLI Changes

On IOS-XE platforms starting from 17.10.1a, there is a CLI correction and an additional CLI was added as part of raw-socket.

The correction is for the **raw-socket idle timeout** command. There is now an option to configure the timeout based on minutes and seconds, whereas the previous configuration used only minutes.

```
Router(config-line)# raw-socket tcp idle-timeout [0-1440] [<0-59> | cr]
```

The additional CLI is for clearing the raw-socket TCP clients. The command syntax is **clear raw-socket line** *[1-145|tty|x/y/z]* for example:

```
Router# clear raw-socket line 0/2/0
```

Note When initiating clear raw-socket line, raw-socket sessions will be cleared for raw-socket clients from the **show raw-socket tcp sessions** command. Connections will be re-established after a TCP hand-shake, which can be done by doing shut/no shut on TCP connection interface.

Verifying Configuration

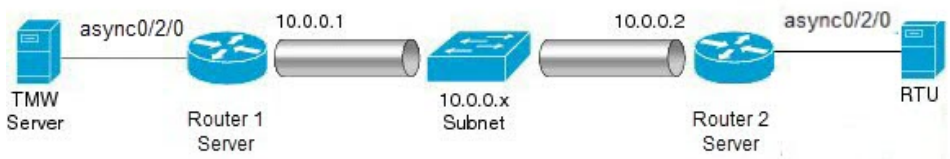
Command	Purpose
show running-config	Shows the configuration of the IR1800, including those features that are active and their settings.
show raw-socket tcp detail	Displays information about Raw Socket Transport TCP activity.
show raw-socket tcp sessions	Displays information about Raw Socket Transport TCP sessions.
show raw-socket tcp statistics	Displays Raw Socket Transport TCP statistics for each asynchronous serial line.
show raw-socket udp detail	Displays information about Raw Socket Transport UDP activity.
show raw-socket udp sessions	Displays information about Raw Socket Transport UDP sessions.
show raw-socket udp statistics	Displays Raw Socket Transport UDP statistics for each asynchronous serial line.
clear raw-socket statistics	Clears Raw Socket Transport statistics for a specific TTY interface or for all asynchronous serial lines.

Raw Socket Transport Configuration Examples

The following sections include Raw Socket Transport configuration examples:

Raw Socket TCP

The following example shows a Raw Socket Transport configuration in which an IR1800 router (Router 1) acts as the server, and another IR809 (Router 2) acts as the client.



The following table displays the configuration of the server and client IR1800s highlighted in the above figure:

IR1800 Server Configuration	IR807 Client Configuration
... interface async0/2/0 no ip address **encapsulation raw-tcp** ! ... line 0/2/0 **raw-socket tcp server 5000 10.0.0.1** **raw-socket packet-timer 3** **raw-socket tcp idle-timeout 5** ...	... interface async0 no ip address **encapsulation raw-tcp** ! interface async1 no ip address **encapsulation raw-tcp** ! ... line 1 **raw-socket tcp client 10.0.0.1 5000 10.0.0.2 9000** **raw-socket packet-length 32** **raw-socket tcp idle-timeout 5** line 2 **raw-socket tcp client 10.0.0.1 5000 10.0.0.2 9001** **raw-socket packet-length 32** **raw-socket tcp idle-timeout 5**

Raw Socket UDP Example

This example shows the configuration for a Raw Socket UDP connection between two IR1800 routers:

From Router1

```
interface GigabitEthernet0/1
ip address 192.168.0.8 255.255.255.0
duplex auto
speed auto
interface async0/2/0
no ip address
encapsulation raw-udp
line 0/2/0
raw-socket udp connection 192.168.0.2 2 2
```

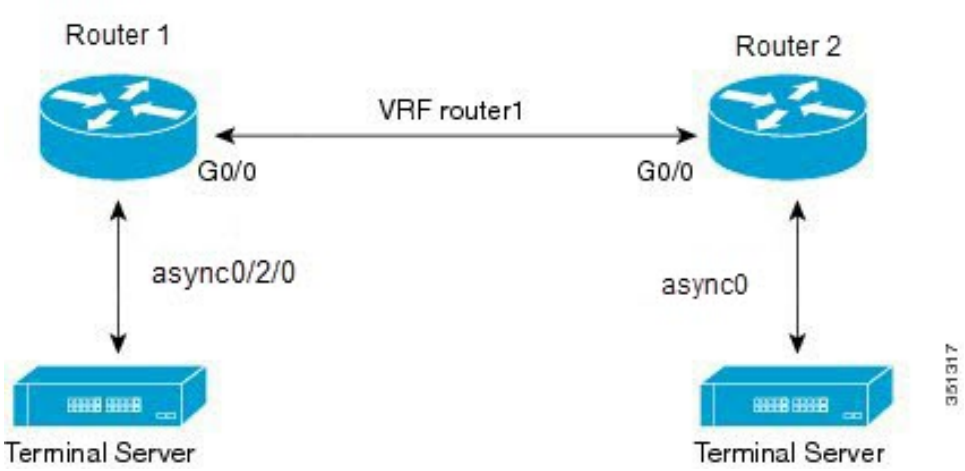
From Router2

```
interface GigabitEthernet0/1
ip address 192.168.0.2 255.255.255.0
load-interval 60
duplex auto
speed auto
no keepalive
```

```
interface async0/2/0
no ip address
encapsulation raw-udp
line 0/2/0
raw-socket udp connection 192.168.0.8 2 2
```

Raw Socket VRF

The following example shows a Raw Socket VRF configuration in which two routers, configured for Raw Socket Transport, connect through a VRF. Router1 is an IR1800, serves as the Raw Socket TCP server, and Router2 is an IR807 serves as the Raw Socket TCP client.



Following are the configurations of Router1 and Router2 as shown in the above figure:

Router1 Configuration

Defining VRF on the router:

```
vrf definition router1
 rd 100:1
 route-target export 100:3
 route-target import 100:3
 !
 address-family ipv4
 exit-address-family
```

Applying VRF configuration on the interface:

```
interface GigabitEthernet0/0
 vrf forwarding router1
 ip address 100.100.100.2 255.255.255.0
 duplex auto
 speed auto
```

Applying raw-tcp on the serial interface:

```
interface async0/2/0
 vrf forwarding router1
 no ip address
 encapsulation raw-tcp
```

Applying raw-tcp on the line:

```
line 0/2/0
 raw-socket tcp server 5000 4.4.4.4
```

Router2 Configuration

Defining VRF on the router:

```
vrf definition router1
 rd 100:1
 route-target export 100:3
 route-target import 100:3
 !
 address-family ipv4
 exit-address-family
```

Applying VRF configuration on the interface:

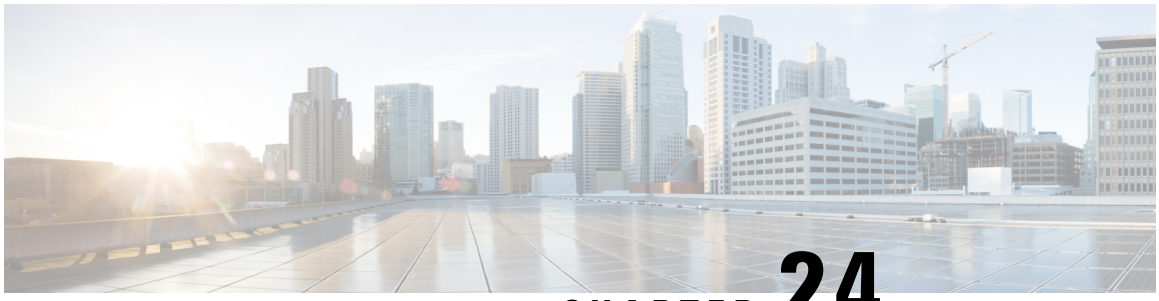
```
interface GigabitEthernet0/0
 vrf forwarding router1
 ip address 100.100.100.1 255.255.255.0
 duplex auto
 speed auto
```

Applying raw-tcp on the serial interface:

```
interface async0
 vrf forwarding router1
 no ip address
 encapsulation raw-tcp
```

Applying raw-tcp on line:

```
line 1
  raw-socket tcp client 4.4.4.4 5000
```



CHAPTER 24

IOx Application Hosting

This section contains the following topics:

- Application Hosting, on page 297
- Information About Application Hosting, on page 297
- Application Hosting on the IR1800 Industrial Integrated Services Router, on page 299
- How to Configure Application Hosting, on page 301
- Installing and Uninstalling Apps, on page 304
- Overriding the App Resource Configuration, on page 304
- Verifying the Application Hosting Configuration, on page 306
- Configuration Examples for Application Hosting, on page 307
- Native docker support, on page 308
- Digital IO for IOx container applications, on page 309
- Signed Application Support, on page 310
- Cisco ThousandEyes Enterprise Agent, on page 310

Application Hosting

A hosted application is a software as a service solution, and it can be run remotely using commands. Application hosting gives administrators a platform for leveraging their own tools and utilities.

This module describes the Application Hosting feature and how to enable it.

Information About Application Hosting

This section contains the following:

Need for Application Hosting

The move to virtual environments has given rise to the need to build applications that are reusable, portable, and scalable. Application hosting gives administrators a platform for leveraging their own tools and utilities. An application, hosted on a network device, can serve a variety of purposes. This ranges from automation, configuration management monitoring, and integration with existing tool chains.

Cisco devices support third-party off-the-shelf applications built using Linux tool chains. Users can run custom applications cross-compiled with the software development kit that Cisco provides.

IOx Overview

IOx is a Cisco-developed end-to-end application framework that provides application hosting capabilities for different application types on Cisco network platforms.

IOx architecture for the IR1800 is different compared to other Cisco platforms that use the hypervisor approach. In other platforms, IOx runs as a virtual machine. IOx is running as a process on the IR1800.

Cisco Application Hosting Overview

The IR1800 enables the user to deploy the application using the app-hosting CLIs. These app-hosting CLIs are not available on the other older platforms. There are additional ways to deploy the applications using the Local Manager and Fog Director.

Application hosting provides the following services:

- Launches designated applications in containers.
- Checks available resources (memory, CPU, and storage), and allocates and manages them.
- Provides support for console logging.
- Provides access to services via REST APIs.
- Provides a CLI endpoint.
- Provides an application hosting infrastructure referred to as Cisco Application Framework (CAF).
- Helps in the setup of platform-specific networking (packet-path) via VirtualPortGroup and management interfaces.

The container is referred to as the virtualization environment provided to run the guest application on the host operating system. The Cisco IOS-XE virtualization services provide manageability and networking models for running guest applications. The virtualization infrastructure allows the administrator to define a logical interface that specifies the connectivity between the host and the guest. IOx maps the logical interface into the Virtual Network Interface Card (vNIC) that the guest application uses.

Applications to be deployed in the containers are packaged as TAR files. The configuration that is specific to these applications is also packaged as part of the TAR file.

The management interface on the device connects the application hosting network to the IOS management interface. The Layer 3 interface of the application receives the Layer 2 bridged traffic from the IOS management interface. The management interface connects through the management bridge to the container/application interface. The IP address of the application must be on the same subnet as the management interface IP address.

IOXMAN

IOXMAN is a process that establishes a tracing infrastructure to provide logging or tracing services for guest applications, except Libvirt, that emulates serial devices. IOXMAN is based on the lifecycle of the guest application to enable and disable the tracing service, to send logging data to IOS syslog, to save tracing data to IOx tracelog, and to maintain IOx tracelog for each guest application.

Application Hosting on the IR1800 Industrial Integrated Services Router

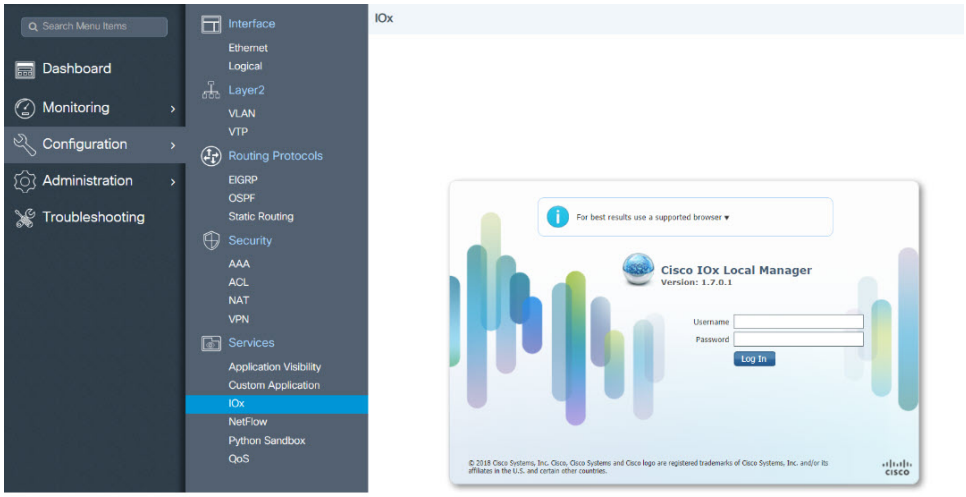
This section describes the application-hosting characteristics specific to the IR1800 Industrial Integrated Services Router.

Note

- The IR1800 CPU is not based on x86 architecture like other Routers. Therefore, this requires the application to comply with the ARM 64-bits architecture.
- Before installing or activating an IOx application from Local Manager, you need to configure the required Virtual Port Group interfaces.

Application hosting can be achieved using the app-hosting cli's as well using the Local Manager and Fog Director. Application hosting using Local Manager is done through the WebUI. In order to deply the applications using Local Manager, WebUI should be enabled and then login to the Local Manager.

Figure 31: Local Manager



1. From the WebUI, click on **Configuration > Services > IOx**
2. Login using the username and password configured.
3. Follow the steps for the application lifecycle in the **Cisco IOx Local Manager Reference Guide** using this link: https://www.cisco.com/c/en/us/td/docs/routers/access/800/software/guides/iox/lm/reference-guide/1-7/b_iox_lm_ref_guide_1_7/b_iox_lm_ref_guide_1_7_chapter_011.html

The next section explains the deployment of an application using the app-hosting cli's.

VirtualPortGroup

The VirtualPortGroup is a software construct on Cisco IOS that maps to a Linux bridge IP address. As such, the VirtualPortGroup represents the switch virtual interface (SVI) of the Linux container. Each bridge can

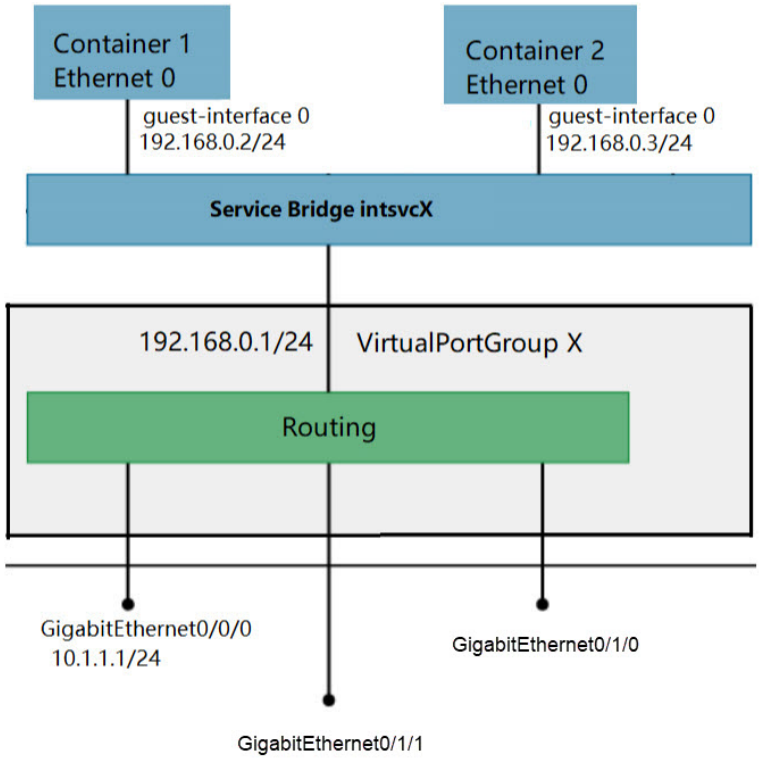
contain multiple interfaces; each mapping to a different container. Each container can also have multiple interfaces.

VirtualPortGroup interfaces are configured by using the interface virtualportgroup command. Once these interfaces are created, IP address and other resources are allocated.

The VirtualPortGroup interface connects the application hosting network to the IOS routing domain. The Layer 3 interface of the application receives routed traffic from IOS. The VirtualPortGroup interface connects through the SVC Bridge to the container/application interface.

The following graphic helps to understand the relationship between the VirtualPortGroup and other interfaces, as it is different than the IR8x9 routers.

Figure 32: Virtual Port Group Mapping



vNIC

For the container life cycle management, the Layer 3 routing model that supports one container per internal logical interface is used. This means that a virtual Ethernet pair is created for each application; and one interface of this pair, called vNIC is part of the application container. The other interface, called vpgX is part of the host system.

NIC is the standard Ethernet interface inside the container that connects to the platform dataplane for the sending and receiving of packets. IOx is responsible for the gateway (VirtualPortGroup interface), IP address, and unique MAC address assignment for each vNIC in the container.

The vNIC inside the container/application are considered as standard Ethernet interfaces.

How to Configure Application Hosting

This section contains the following:

Enabling IOx

Perform this task to enable access to the IOx Local Manager. The IOx Local Manager provides a web-based user interface that you can use to manage, administer, monitor, and troubleshoot apps on the host system, and to perform a variety of related activities

Note In the steps that follow, IP HTTP commands do not enable IOX, but allow the user to access the WebUI to connect the IOX Local Manager.

Procedure

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device>enable`	Enables privileged EXEC mode. Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device#configure terminal`	Enters global configuration mode.
Step 3	**iox** **Example:** `Device(config)#iox`	Enables IOx.
Step 4	**ip http server** **Example:** `Device(config)#ip http server`	Enables the HTTP server on your IP or IPv6 system.
Step 5	**ip http secure-server** **Example:** `Device(config)#ip http secure-server`	Enables a secure HTTP (HTTPS) server.
Step 6	**username** *name* **privilege** *level* **password** {**0** | **7** | *user-password* }*encrypted-password* **Example:** `Device(config)#username cisco privilege 15 password 0 cisco`	Establishes a username-based authentication system and privilege level for the user. The username privilege level must be configured as 15.

	Command or Action	Purpose
Step 7	**end** **Example:** `Device(config-if)#end`	Exits interface configuration mode and returns to privileged EXEC mode

Configuring a VirtualPortGroup to a Layer 3 Data Port

Multiple Layer 3 data ports can be routed to one or more VirtualPortGroups or containers. VirutalPortGroups and Layer 3 data ports must be on different subnets.

Enable the **ip routing** command to allow external routing on the Layer 3 data-port.

Procedure

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device>enable`	Enables privileged EXEC mode. Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device#configure terminal`	Enters global configuration mode.
Step 3	**ip routing** **Example:** `Device(config)#ip routing`	Enables IP routing. The **ip routing** command must be enabled to allow external routing on Layer 3 data ports.
Step 4	**interface** *type number* **Example:** `Device(config)#interface gigabitethernet 0/0/0`	Configures an interface and enters interface configuration mode.
Step 5	**ip address** *ip-address mask* **Example:** `Device(config)#ip address 10.1.1.1 255.255.255.0`	Configures an IP address for the interface.
Step 6	**exit** **Example:** `Device(config-if)#exit`	Exits interface configuration mode and returns to global configuration mode.
Step 7	**interface** *type number* **Example:** `Device(config)#interface virtualportgroup 0`	Configures an interface and enters interface configuration mode.

	Command or Action	Purpose
Step 8	**ip address** *ip-address mask* **Example:** `Device(config-if)#ip address 192.168.0.1 255.255.255.0`	Configures an IP address for the interface.
Step 9	**end** **Example:** `Device(config-if)#end`	Exits interface configuration mode and returns to privileged EXEC mode
Step 10	**configure terminal** **Example:** `Device#configure terminal`	Enters global configuration mode.
Step 11	**app-hosting appid** *app_number* **Example:** `Device(config)#app-hosting appid app1`	Configures the application and enters the application configuration mode.
Step 12	**app-vnic gateway0 virtualportgroup 0 guest-interface 0** **Example:** `Device(config-app-hosting)#app-vnic gateway0 virtualportgroup 0 guest-interface 0`	Configures the application interface and the gateway of the application.
Step 13	**guest-ipaddress** *ip_address* **netmask** *netmask* **Example:** `Device(config-app-hosting-gateway0)#guest-ipaddress 192.168.0.2 netmask 255.255.255.0`	Configures the application Ethernet interface ip address.
Step 14	**app-default-gateway** *ip_address* **guest-interface 0** **Example:** `Device(config-app-hosting-gateway0)#app-default-gateway 192.168.0.1 guest-interface 0`	Configures the default gateway for the application.
Step 15	**end** **Example:** `Device#end`	Exits interface configuration mode and returns to privileged EXEC mode

Installing and Uninstalling Apps

Procedure

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device>enable`	Enables privileged EXEC mode. Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device#configure terminal`	Enters global configuration mode.
Step 3	**app-hosting install appid** *application-name* **package** *package-path* **Example:** `Device#app-hosting install appid lxc_app package flash:my_iox_app.tar`	Installs an app from the specified location. The app can be installed from any local storage location such as, flash, bootflash, and usbflash0.
Step 4	**app-hosting start appid** *application-name* **Example:** `Device#app-hosting start appid app1`	Starts the application. Application start-up scripts are activated.
Step 5	**app-hosting stop appid** *application-name* **Example:** `Device#app-hosting stop appid app1`	Stops the application.
Step 6	**app-hosting deactivate appid** *application-name* **Example:** `Device#app-hosting deactivate appid app1`	Deactivates all resources allocated for the application.
Step 7	**app-hosting uninstall appid** *application-name* **Example:** `Device#app-hosting uninstall appid app1`	Uninstalls the application. Uninstalls all packaging and images stored. All changes and updates to the application are also removed.

Overriding the App Resource Configuration

Resource changes will take effect only after the app-hosting activate command is configured.

Procedure

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device>enable`	Enables privileged EXEC mode. Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device#configure terminal`	Enters global configuration mode.
Step 3	**app-hosting appid** *name* **Example:** `Device(config-app-hosting)#app-resource profile custom`	Enables application hosting and enters application hosting configuration mode.
Step 4	**app-resource profile** *name* **Example:** `Device#app-hosting start appid app1`	Configures the custom application resource profile, and enters custom application resource profile configuration mode. Only the custom profile name is supported.
Step 5	**cpu** *unit* **Example:** `Device(config-app-resource-profile-custom)# cpu 800`	Changes the default CPU allocation for the application. Resource values are application-specific, and any adjustment to these values must ensure that the application can run reliably with the changes.
Step 6	**memory** *memory* **Example:** `Device(config-app-resource-profile-custom)# memory 512`	Changes the default memory allocation.
Step 7	**vcpu** *number* **Example:** `Device(config-app-resource-profile-custom)# vcpu 2`	Changes the virtual CPU (vCPU) allocation for the application.
Step 8	**end** **Example:** `Device(config-app-resource-profile-custom)# end`	Exits custom application resource profile configuration mode and returns to privileged EXEC mode.

Verifying the Application Hosting Configuration

Procedure

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

Step 3	**show iox-service**
	Example:
	`Device(config)#show iox-service`
	`IOx Infrastructure Summary:`
	`---------------------------------`
	`IOx service (CAF) 1.8.0.2 : Running`
	`IOx service (HA) : Not Supported`
	`IOx service (IOxman) : Running`
	`Libvirtd 1.3.4 : Running`
	`Device#`
		Displays the status of all IOx services.

Step 4	**show app-hosting detail**
	Example:
	`Device#show app-hosting detail`
	`App id : app1`
	`Owner : iox`
	`State : RUNNING`
	`Application`
	`Type : lxc`
	`Name : nt08-stress`
	`Version : 0.1`
	`Description : Stress Testing`
	`Application`
	`Path : usbflash0:my_iox_app.tar`
	`Activated profile name : custom`
	`Resource reservation`
	`Memory : 64 MB`
	`Disk : 2 MB`
	`CPU : 500 units`
	`Attached devices`
	`Type Name Alias`
	`---------- ---------- -----`
	`serial/shell iox_console_shell serial0`
	`serial/aux iox_console_aux serial1`
		Displays detailed information about the application.

	Command or Action	Purpose
	serial/syslog iox_syslog serial2 serial/trace iox_trace serial3 Network interfaces ---------- eth0: MAC address : 52:54:dd:fa:25:ee	
Step 5	**show app-hosting list** **Example:** Device#**show app-hosting list** App id State ---------- app1 RUNNING	Displays the list of applications and their status.

Configuration Examples for Application Hosting

See the following examples:

Example: Enabling IOx

```
Device> enable
Device# configure terminal
Device(config)# iox
Device(config)# ip http server
Device(config)# ip http secure-server
Device(config)# username cisco privilege 15 password 0 cisco
Device(config)# end
```

Example: Configuring a VirtualPortGroup to a Layer 3 Data Port

```
Device> enable
Device# configure terminal
Device(config)# ip routing
Device(config)# interface gigabitethernet 0/0/0
Device(config-if)# ip address 10.1.1.1 255.255.255.0
Device(config-if)# exit
Device(config)# interface virtualportgroup 0
Device(config-if)# ip address 192.168.0.1 255.255.255.0
Device(config-if)# end
```

Example: Installing and Uninstalling Apps

```
Device> enable
Device# app-hosting install appid app1 package flash:my_iox_app.tar
```

```
Device# app-hosting activate appid app1
Device# app-hosting start appid app1
Device# app-hosting stop appid app1
Device# app-hosting deactivate appid app1
Device# app-hosting uninstall appid app1
```

Example: Overriding the App Resource Configuration

```
Device# configure terminal
Device(config)# app-hosting appid app1
Device(config-app-hosting)# app-resource profile custom
Device(config-app-resource-profile-custom)# cpu 800
Device(config-app-resource-profile-custom)# memory 512
Device(config-app-resource-profile-custom)# vcpu 2
Device(config-app-resource-profile-custom)# end
```

Native docker support

Native Docker Support enables users to deploy the docker applications on the IR1800. The application lifecycle process is similar to the procedure in the Installing and Uninstalling Apps section. For docker applications, entry point configuration is required as part of the application configuration. Please refer to the following example for the entry point configuration.

```
Router#conf t
Enter configuration commands, one per line. End with CNTL/Z.
Router(config)#app-hosting appid app3
Router(config-app-hosting)#app-vnic gateway0 virtualportgroup 0 guest-interface 0
Router(config-app-hosting-gateway0)#guest-ipaddress 192.168.0.7 netmask 255.255.255.0
Router(config-app-hosting-gateway0)#app-default-gateway 192.168.0.1 guest-interface 0
Router(config-app-hosting)#app-resource docker
Router(config-app-hosting-docker)#run-opts 1 "--entrypoint '/bin/sleep 10000'"
Router(config-app-hosting-docker)#end
Router#
```

The output for docker applications is shown in the following example:

```
Router#show app-hosting detail
App id : app1
Owner : iox
State : RUNNING
Application
Type : docker
Name : aarch64/busybox
Version : latest
Description :
Path : bootflash:busybox.tar
Activated profile name : custom
Resource reservation
Memory : 431 MB
Disk : 10 MB
CPU : 577 units
VCPU : 1
Attached devices
Type Name Alias
---------------------------------------------
serial/shell iox_console_shell serial0
serial/aux iox_console_aux serial1
serial/syslog iox_syslog serial2
```

```
serial/trace iox_trace serial3
Network interfaces
---------------------------------------
eth0:
MAC address : 52:54:dd:e9:ab:7a
IPv4 address : 192.168.0.7
Network name : VPG0
Docker
------
Run-time information
Command :
Entry-point : /bin/sleep 10000
Run options in use : --entrypoint '/bin/sleep 10000'
Application health information
Status : 0
Last probe error :
Last probe output :
Router#
```

Digital IO for IOx container applications

IOx container applications are able to access the digital IO. There is a CLI for alarm contact command.

```
Router(config)# alarm contact ?
  <1-4>          Digital I/O port number)
attach-to-iox  Allow access of all Digital IO Ports access from IOX
Router(config)# alarm contact attach-to-iox
```

Enabling the **attach-to-iox** command will provide complete control of all Digital IO ports to IOx. The ports will be exposed as four character devices /dev/dio-[1-4] to IOX applications. You can use read/write functions to get/set values of the Digital IO ports.

If you wish to update the mode, you can write the mode value to the character device file. This is accomplished by IOCTL calls to read/write the state, change mode, and read the true analog voltage of the port. Following this method, you can attach analog sensors to the IR1800. All ports are initially set to Input mode with voltage pulled up to 3.3v.

The following are examples of IOCTL calls:

Read Digital IO Port:

```
cat /dev/dio-1
```

Write to Digital IO Port:

```
echo 0 > /dev/dio-1
echo 1 > /dev/dio-1
```

Change mode:

```
echo out > /dev/dio-1
echo in > /dev/dio-1
```

List of IOCTLs supported:

```
DIO_GET_STATE = 0x1001
DIO_SET_STATE = 0x1002
DIO_GET_MODE = 0x1003
DIO_SET_MODE_OUTPUT = 0x1004
DIO_SET_MODE_INPUT = 0x1005
DIO_GET_THRESHOLD 0x1006
```

```
DIO_SET_THRESHOLD = 0x1007
DIO_GET_VOLTAGE = 0x1009
```

Read State using IOCTL:

```
import fcntl, array
file = open("/dev/dio-1","rw")
state = array.array('L',[0])
fcntl.ioctl(file, DIO_GET_STATE, state)
print(state[0])
```

Change mode using IOCTL:

```
import fcntl
file = open("/dev/dio-1","rw")
fcntl.ioctl(file, DIO_SET_MODE_OUTPUT, 0)
```

Signed Application Support

Cisco Signed applications are now supported on the IR1800. In order to install a signed application, signed verification has to be enabled on the device. Signed verification can be enabled by following the following instructions.

```
Router#configure terminal
Enter configuration commands, one per line. End with CNTL/Z.
Router(config)#
Router(config)#app-hosting signed-verification
Router(config)#
Router(config)#exit
```

After enabling the signed verification, follow the instructions in the Installing and Uninstalling Apps section under IOx Application Hosting in order to install the application.

Cisco ThousandEyes Enterprise Agent

A Cisco ThousandEyes Enterprise Agent is a network monitoring solution that

- runs a variety of tests using agents
- provides real-time monitoring of network and application performance, and
- offers multidimensional insights with routing and device data for end-to-end visibility.

Additional information

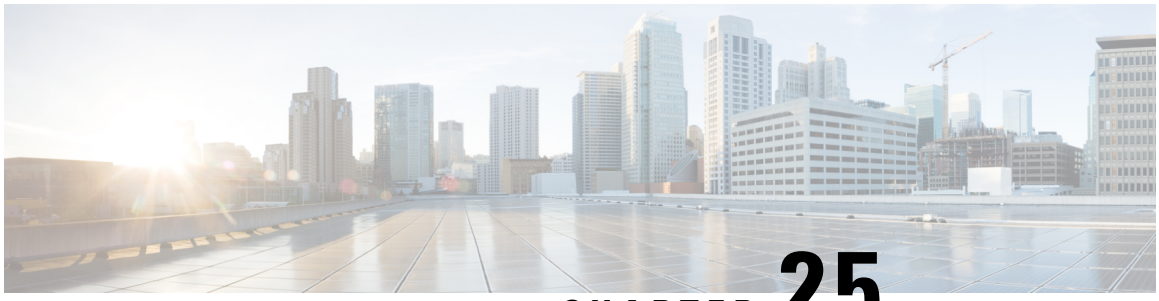
The Cisco ThousandEyes Enterprise Agent enables you to view end-to-end paths across networks and services affecting your business. It actively monitors network traffic paths across internal, external, and internet networks, helping analyze network performance and application availability.

You can use application-hosting features to deploy the Cisco ThousandEyes Enterprise Agent as a container application on Cisco Industrial IoT Routers. The agent runs as a Docker container using the Cisco IOx.

Starting with Cisco IOS XE Release 17.18.2, you can configure Cisco ThousandEyes Enterprise Agent in Controller mode.

For details about configuring Cisco ThousandEyes Enterprise Agent in controller mode, see Cisco SD-WAN Systems and Interfaces Configuration Guide.

For more information about Cisco ThousandEyes Enterprise Agent, see ThousandEyes Documentation.



CHAPTER 25

Serial Relay Service

This chapter contains the following:

- IOx Serial Relay Service, on page 313
- Data Paths, on page 313
- Configuration Commands, on page 315

IOx Serial Relay Service

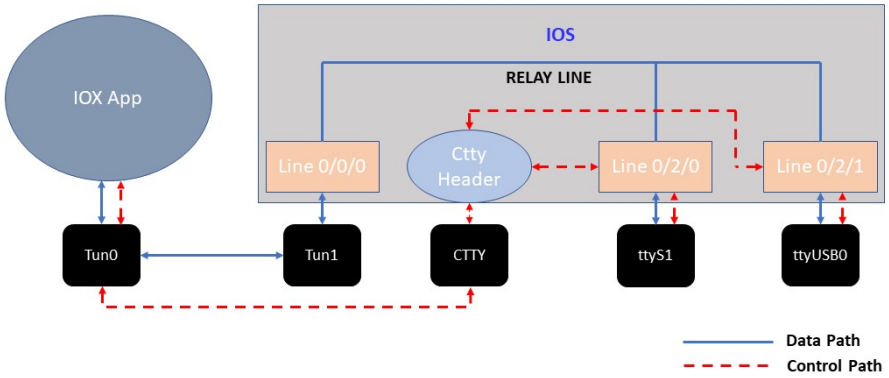
IOx Serial Relay service on the IR1800 enables IOx apps to communicate with the Async Serial port (/dev/ttyS1 or /dev/ttyUSB0 under IOS-XE). The configuration of IOx Serial Relay service is similar to that of the IR800.

Data Paths

On the IR1800, IOS-XE has complete control over the data path and control path of the Async Serial port. This aspect is essential to other encapsulations supported on the Aysnc port such as PPP, raw-socket, SCADA, etc. The IOx app is never allowed to exercise full control over the device. All data and configurations are passed through IOS-XE before going to the device.

Instead of exposing the actual Serial port to IOx apps, the Serial relay service creates a software emulated serial tty device enumerated as /dev/ttyTun0 (shown below). The pair of devices /dev/ttyTun0 and /dev/ttyTun1 represent a data tunnel whose primary function is to act as a pass-through gateway during any data transfer. /dev/ttyTun1 is open by IOS-XE and all the ingress/egress data from IOS to the app uses this device during data transfer. Line 0/0/0 is used to communicated with /dev/ttyTun1. Serial relay service should be configured beforehand to allow the connection between two lines.

Figure 33: Data Paths



Data Path:

1. When the IOx app sends a character to /dev/ttyTun0, the tunnel driver automatically pushes the data to /dev/ttyTun1.
2. IOS reads the data which it then passes to the Serial relay service.
3. The Serial relay service retrieves information about the other end of the relay service (Line 0/2/0 or Line 0/2/1 in this case) and forwards the data to the Line's buffer.
4. The line driver actively pushes the data into the actual serial device (/dev/ttyS1 or /dev/ttyUSB0) based on buffer availability.
5. The reverse path functions the same with the roles of /dev/ttyS1 or /dev/ttyUSB0 and /dev/tun0 reversed.

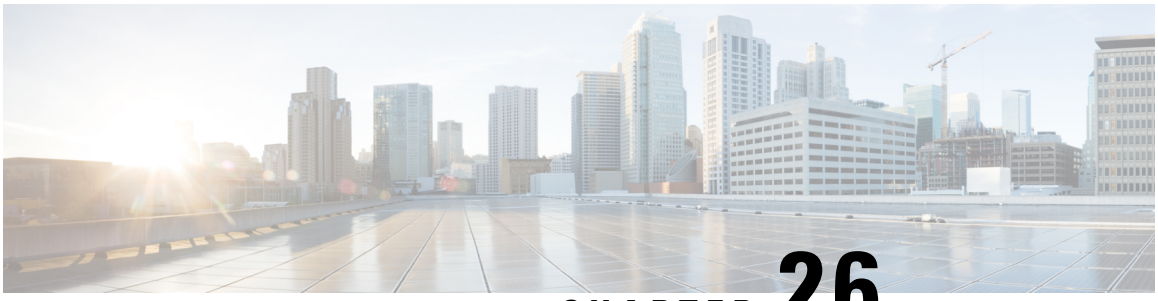
Control Path:

1. When the IOx app performs TCGETS ioctl call on /dev/ttyTun0, the tunnel driver uses /dev/cttyTun to send request to the CTTY handler service running in IOS.
2. CTTY handler service and the kernel driver use a client-server architecture to communicate configuration objects.
3. Upon receiving the request about TCGETS from /dev/cttyTun, the CTTY handler examines the request and requests Line driver to populate the required data into control data structures.
4. Upon receiving the control data structures, CTTY handler sends out a response to /dev/cttyTun which eventually goes back to /dev/ttyTun0.
5. /dev/ttyTun0 passes the control data to IOx app as requested.

6. Similar path can be extrapolated for TCSETS where the CTTY handler requests the Line driver to update the settings of the underneath /dev/ttyS1 or /dev/ttyUSB0 driver.
7. Line driver of Line 0/2/0 or Line 0/2/1 and driver config on /dev/ttyTun0 are always in sync with each other. Any configuration changes such as baud rate modification is transparently propagated to the Line driver without any additional configuration overhead. This emulates the propagation feature of Serial relay on the IR800 series where the virtual serial port can configure the parameters of the real serial port.

Configuration Commands

```
IR1800#configure terminal
IR1800(config)#interface async 0/2/0
IR1800(config-if)#encapsulation relay-line
IR1800(config-if)#exit
IR1800(config)#relay line 0/2/0 0/0/0
IR1800(config)#exit
IR1800#
```



CHAPTER 26

Support for MACsec

- Software Supported MACsec, on page 317

Software Supported MACsec

All existing Cisco IOS XE based routers use special transceivers to perform MACsec encryption and decryption. The software MACsec uses CDAL infrastructure in QFP to perform crypto operations. Compared to the hardware supported MACsec, the process of configuration, status and datapath is performed, it is different which creates certain limitations in the functionality when used.

MACsec is supported only on L2 interfaces. The MACsec port must be put into access mode. As the encryption happens on the egress switch virtual interface (SVI), the VLAN used for the port should be unique, and no other interface must use that VLAN. This is because the QFP does not have the MAC table information.

Note

- Since MACsec is being done through software, performances are not line rate on L2 interfaces.
- Cisco supports only the *shouldsecure* MACsec mode for IR1800, which allows unencrypted traffic even in a secured state.

  The IR1800 does not support the *mustsecure* mode.

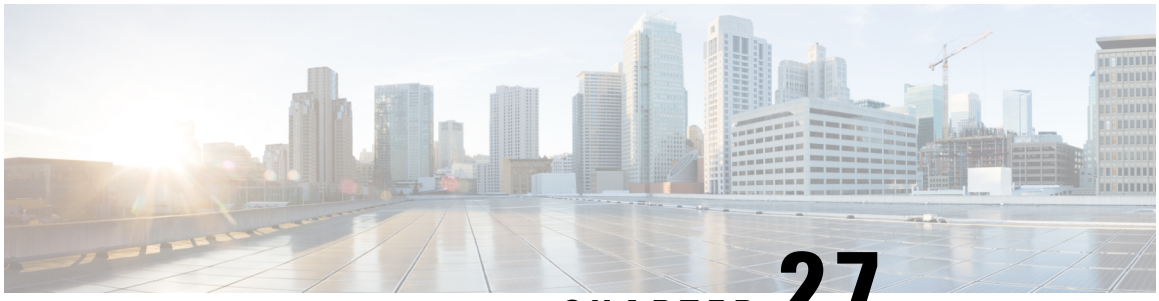
For an egress packet, the SVI is aware that the packet needs to be sent out on a VLAN without information about any specific interface. It is the switch chip that decides which port to send it to. All the packets without the MACsec tag are processed without any changes. The outgoing L2 packets will also egress without encryption or modification.

For this feature, the Network Essentials and Network Advantage license support GCM-AES-128. This feature is not available running the NPE image.

Limitations

- MACsec is not supported in controller mode.
- There must be a unique vlan id for a MACsec interface.
- Only gcm-aes-128 is supported

- Both explicit and non-explicit SCI are supported on ingress side. The IR1800 sends out only explicit SCI packets as it is not an end system.
- The IR1800 does not support confidentiality offset.
- Integrity only is not supported.
- For gcm-aes-128, up to 32 bytes are added to an encrypted packet compared to a plain packet. So the MTU setup should add 32 for it to work properly.
- The MACsec key is managed by the MKA module. For that device, it requires a static key for MKA to negotiate MACsec key.
- There is no MIB support.
- Jumbo Frame is not supported.
- MACsec is not supported on the WAN port.
- IP Device Tracking (IPDT) is not supported on Host to switch MACsec



CHAPTER 27

ROM Monitor Overview

- ROM Monitor Overview, on page 319
- Access ROM Monitor Mode, on page 320
- Displaying the Configuration Register Setting, on page 322
- Environment Variable Settings, on page 322
- Exiting ROM Monitor Mode, on page 324
- ROMMON Configuration Example, on page 324
- Upgrading the ROMmon for a Router, on page 325

ROM Monitor Overview

The *ROM Monitor* is a bootstrap program that initializes the hardware and boots the Cisco IOS XE software when you power on or reload a router. When you connect a terminal to the router that is in ROM Monitor mode, the ROM Monitor (rommon 1>) prompt is displayed.

During normal operation, users do not use ROM Monitor mode. ROM Monitor mode is used only in special circumstances, such as reinstalling the entire software set, resetting the router password, or specifying a configuration file to use at startup.

The *ROM Monitor software* is known by many names. It is sometimes called *ROMMON* because of the CLI prompt in ROM Monitor mode. The ROM Monitor software is also called the *boot software* , *boot image* , or *boot helper* . Although it is distributed with routers that use the Cisco IOS XE software, ROM Monitor is a separate program from the Cisco IOS XE software. During normal startup, the ROM Monitor initializes the router, and then control passes to the Cisco IOS XE software. After the Cisco IOS XE software takes over, the ROM Monitor is no longer in use.

Attention On the IR1800 series routers, if power is lost 20 times in a row during bootup, the router will drop into ROM Monitor.

Environmental Variables and the Configuration Register

Two primary connections exist between ROM Monitor and the Cisco IOS XE software: the ROM Monitor environment variables and the configuration register.

The ROM Monitor environment variables define the location of the Cisco IOS XE software and describe how to load it. After the ROM Monitor has initialized the router, it uses the environment variables to locate and load the Cisco IOS XE software.

The *configuration register* is a software setting that controls how a router starts up. One of the primary uses of the configuration register is to control whether the router starts in ROM Monitor mode or Administration EXEC mode. The configuration register is set in either ROM Monitor mode or Administration EXEC mode as needed. Typically, you set the configuration register using the Cisco IOS XE software prompt when you need to use ROM Monitor mode. When the maintenance in ROM Monitor mode is complete, you change the configuration register so the router reboots with the Cisco IOS XE software.

Accessing ROM Monitor Mode with a Terminal Connection

When the router is in ROM Monitor mode, you can access the ROM Monitor software only from a terminal connected directly to the console port of the card. Because the Cisco IOS XE software (EXEC mode) is not operating, non-management interfaces are not accessible. Basically, all Cisco IOS XE software resources are unavailable. The hardware is available, but no configuration exists to make use of the hardware.

Network Management Access and ROM Monitor Mode

It is important to remember that ROM Monitor mode is a router mode, not a mode within the Cisco IOS XE software. It is best to remember that ROM Monitor software and the Cisco IOS XE software are two separate programs that run on the same router. At any given time, the router runs only one of these programs.

One area that can be confusing when using ROM Monitor and the Cisco IOS XE software is the area that defines the IP configuration for the Management Ethernet interface. Most users are comfortable with configuring the Management Ethernet interface in the Cisco IOS XE software. When the router is in ROM Monitor mode, however, the router does not run the Cisco IOS XE software, so that Management Ethernet interface configuration is not available.

When you want to access other devices, such as a TFTP server, while in ROM Monitor mode on the router, you must configure the ROM Monitor variables with IP access information.

Note TFTP access variables are currently not supported on the IR1800 platform.

Access ROM Monitor Mode

The following sections describe how to enter the ROMMON mode, and contains the following sections:

Checking the Current ROMMON Version

To display the version of ROMmon running on a router, use the **show rom-monitor** command. To show all variables that are set in ROMmon, use the **show romvar** command.

```
Router#show rom-monitor r0
============================================================

System Bootstrap, Version 3.9(REL), RELEASE SOFTWARE
Copyright (c) 1994-2021 by cisco Systems, Inc.
```

```
Router# show romvar
ROMMON variables:
PS1 = rommon ! >
THRPUT =
LICENSE_BOOT_LEVEL =
RET_2_RTS =
DEVICE_MANAGED_MODE = autonomous
BOOT = bootflash:/ir1800-universalk9.17.06.01prd18.SPA.bin,12;
BOOT_STAGED = bootflash:ir1800-universalk9.17.06.01prd18.SPA.bin
BSI = 0
RET_2_RCALTS =
RANDOM_NUM = 231998661

Router#
```

If your configuration register was set to hex value 0x0 or 0x1820, reload operation will bring you to the ROMmon mode command prompt (rommon 1>). Invoking the set command at the prompt (rommon 1> set) will display the same information as "show romvar" above in IOS/XE exec mode.

```
rommon 1 > set
PS1=rommon ! >
THRPUT=
LICENSE_BOOT_LEVEL=
RET_2_RTS=
DEVICE_MANAGED_MODE=autonomous
BOOT=bootflash:/ir1800-universalk9.17.06.01prd18.SPA.bin,12;
BOOT_STAGED=bootflash:ir1800-universalk9.17.06.01prd18.SPA.bin
BSI=0
RANDOM_NUM=231998661
RET_2_RCALTS=1626821700
```

Commonly Used ROM Monitor Commands

The following table summarizes the commands commonly used in ROM Monitor. For specific instructions on using these commands, refer to the relevant procedure in this document.

Table 22: Commonly Used ROM Monitor Commands

ROMMON Command	Description
boot image	Manually boots a Cisco IOS XE software image.
boot image –o config-file-path	Manually boots the Cisco IOS XE software with a temporary alternative administration configuration file.
confreg	Changes the config-register setting.
dev	Displays the available local storage devices.
dir	Displays the files on a storage device.
reset	Resets the node.
set	Displays the currently set ROM Monitor environmental settings.
sync	Saves the new ROM Monitor environmental settings.
unset	Removes an environmental variable setting.

Examples

The following example shows what appears when you enter the **?** command on a router:

```
rommon 1 > ?
alias               set and display aliases command
boot                boot up an external process
confreg             configuration register utility
dev                 list the device table
dir                 list files in file system
help                monitor builtin command help
history             monitor command history
meminfo             main memory information
repeat              repeat a monitor command
reset               system reset
set                 display the monitor variables
showmon             display currently selected ROM monitor
sync                write monitor environment to NVRAM
token               display board's unique token identifier
unalias             unset an alias
unset               unset a monitor variable
```

Changing the ROM Monitor Prompt

You can change the prompt in ROM Monitor mode by using the **PS1=** command as shown in the following example:

```
rommon 8 > PS1="IR1800 rommon ! > "
IR1800 rommon 9 >
```

Changing the prompt is useful if you are working with multiple routers in ROM Monitor at the same time. This example specifies that the prompt should be “IR1800 rommon ”, followed by the line number, and then followed by “ > “ by the line number.

Displaying the Configuration Register Setting

To display the current configuration register setting, enter the **confreg** command without parameters as follows:

```
rommon > confreg
Configuration Summary
(Virtual Configuration Register: )
enabled are:
[ 0 ] break/abort has effect
[ 1 ] console baud: 9600
 boot:...... the ROM Monitor
do you wish to change the configuration? y/n  [n]:
```

The configuration register setting is labeled *Virtual Configuration Register*. Enter the **no** command to avoid changing the configuration register setting.

Environment Variable Settings

The ROM Monitor environment variables define the attributes of the ROM Monitor. Environmental variables are entered like commands and are always followed by the equal sign (=). Environment variable settings are entered in capital letters, followed by a definition. For example:

```
IP_ADDRESS=10.0.0.2
```

Under normal operating conditions, you do not need to modify these variables. They are cleared or set only when you need to make changes to the way ROM Monitor operates.

This section includes the following topics:

Frequently Used Environmental Variables

The following table shows the main ROM Monitor environmental variables. For instructions on how to use these variables, see the relevant instructions in this document. The IR1800 boot loader does not support netboot, so any setting like the following environment variables are not used:

- IP_ADDRESS
- IP_SUBNET_MASK
- DEFAULT_GATEWAY
- TFTP_SERVER
- TFTP_FILE

Table 23: Frequently Used ROM Monitor Environmental Variables

Environmental variable	Description
BOOT=path/file	Identifies the boot software for a node. This variable is usually set automatically when the router boots.

Displaying Environment Variable Settings

To display the current environment variable settings, enter the **set** command :

```
rommon 1 > showmon
System Bootstrap, Version 3.9(REL), RELEASE SOFTWARE
Copyright (c) 1994-2021 by cisco Systems, Inc.

IR1833-K9 platform with 4194304 Kbytes of main memory

MCU Version - Bootloader: 22, App: 4D
MCU is in application mode.
```

Entering Environment Variable Settings

Environment variable settings are entered in capital letters, followed by a definition. The following example shows the environmental variables that can be configured in ROMmon mode.

```
rommon 1 > confreg 0x0
rommon 1> BOOT_WDOG = DISABLE
rommon 1> BOOT = IR1800-K9_image_name
```

Saving Environment Variable Settings

To save the current environment variable settings, enter the **sync** command:

```
rommon > sync
```

Note Environmental values that are not saved with the **sync** command are discarded whenever the system is reset or booted.

Exiting ROM Monitor Mode

To exit ROM Monitor mode, you must change the configuration register and reset the router.

Procedure

	Command or Action	Purpose
Step 1	**confreg** **Example:** `rommon 1> confreg`	Initiates the configuration register configuration prompts.
Step 2	Respond to each prompt as instructed.	See the example that follows this procedure for more information.
Step 3	**reset** **Example:** `rommon 2> reset`	Resets and initializes the router.

ROMMON Configuration Example

```
rommon 3 > confreg
Configuration Summary
(Virtual Configuration Register: 0x0)
enabled are:
 [ 0 ] break/abort has effect
 [ 1 ] console baud: 9600
 boot: ...... the ROM Monitor
do you wish to change the configuration? y/n  [n]:  y
 enable  "diagnostic mode"? y/n  [n]:
 enable  "use net in IP bcast address"? y/n  [n]:
 enable  "load rom after netboot fails"? y/n  [n]:
 enable  "use all zero broadcast"? y/n  [n]:
 disable "break/abort has effect"? y/n  [n]:
 enable  "ignore system config info"? y/n  [n]:
 change console baud rate? y/n  [n]:
change the boot characteristics? y/n  [n]:
Configuration Summary
(Virtual Configuration Register: 0x0)
enabled are:
 [ 0 ] break/abort has effect
 [ 1 ] console baud: 9600
 boot: ...... the ROM Monitor
do you wish to change the configuration? y/n  [n]:
```

Upgrading the ROMmon for a Router

ROMmon upgrade on the IR1800-K9 router is automatically done when the image is booted. The latest version of the ROMmon is bundled with the IOSXE image. An algorithm detects if the current running version is older than the bundled version, if so, it is automatically upgraded. If the current running version is equal to the bundled version no upgrade is executed. For every successful upgrade, the router is automatically rebooted in order for the new version to get loaded and executed.

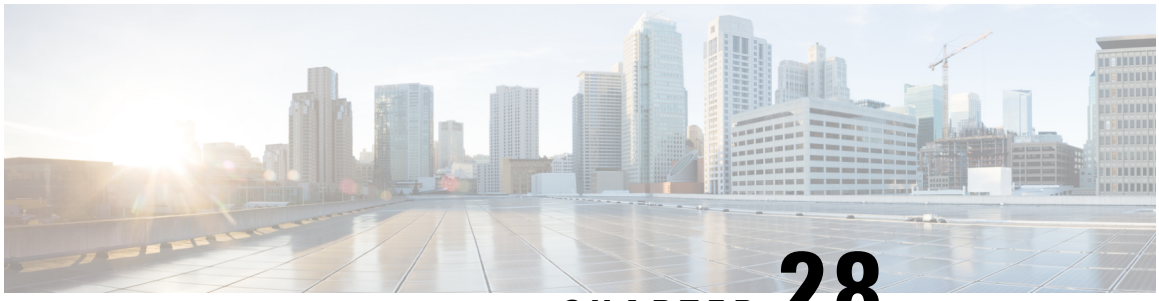
Procedure

Step 1 (Optional) Run the **show rom-monitor** *slot* command on the router to see the current release numbers of ROMmon on the hardware. See the Checking the Current ROMMON Version, on page 320 for information about interpreting the output of the command that you run.

Step 2 If autoboot has not been enabled by using the **config-register 0x2102** command, run the **boot** *filesystem:/file-location* command at the ROMmon prompt to boot the Cisco IOS XE image, where *filesystem:/file-location* is the path to the consolidated package file. The ROMmon upgrade is not permanent for any piece of hardware until the Cisco IOS XE image is booted.

Step 3 Run the **enable** command at the user prompt to enter the privileged EXEC mode after the boot is complete.

Step 4 Run the **show rom-monitor** *slot* command to verify whether the ROMmon has been upgraded.



CHAPTER 28

Connected Grid NMS Agent (CGNA) Support

This chapter contains the following sections:

- Connected Grid NMS Agent (CGNA) Support, on page 327
- CGNA Overview, on page 327
- WebSocket Support, on page 328

Connected Grid NMS Agent (CGNA) Support

CGNA is a subsystem running on IOS which provides connectivity and manageability with network management system like FND and IOT Operations Dashboard. Complete documentation sets for IoT Field Network Director and IoT Industrial Network Director can be found in the following links:

https://www.cisco.com/c/en/us/support/cloud-systems-management/iot-field-network-director/series.html

https://www.cisco.com/c/en/us/support/cloud-systems-management/industrial-network-director/series.html#%7Etab-documents

CGNA Overview

The majority of features are implemented on the IR1800 including basic profile, execution profile, heartbeat profile, and firmware download. Some features like bspatch will not be imported on the IR1800 due to the limitation on the IOS image.

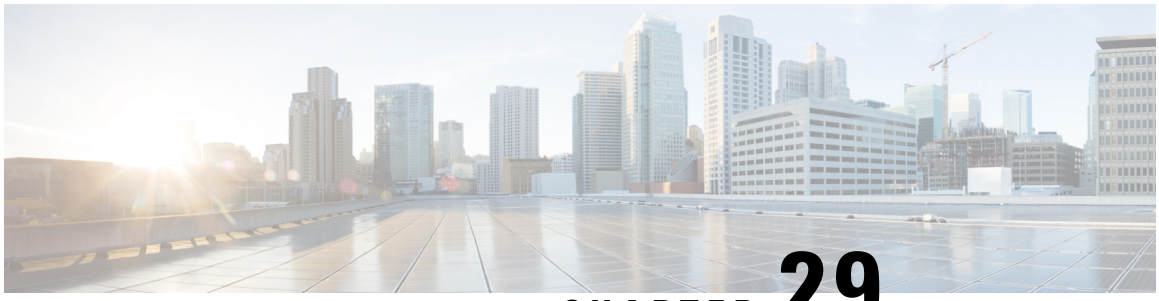
CGNA has four different profiles for different use cases. They are:

- Profile - This is the basic profile. It can be configured to support tunnel provision, registration, and periodic update.
- Execution profile - This can be configured to execute CLI commands
- Heartbeat profile - This can be configured to send heartbeat messages
- Transport profile - This can be configured to provide different transport protocol for the basic profile to use. Currently only supports WebSocket.

CGNA also provides CLIs for resumable image retrieval from network management system and simple SNMP server for redirecting events to WebSocket connection.

WebSocket Support

Traditional IOS provided the WebSocket service in the IOSd kernel. Polaris IOS-XE moves the WebSocket service out of the IOSd kernel for improved performance. The websocket version will be upgraded to 3.2.3.



CHAPTER 29

CLI Output for the FN980 5G Modem

• Change in CLI output for the FN980 5G modem, on page 329

Change in CLI output for the FN980 5G modem

This release has a different output to the **show cellular 0/x/0 radio band** command. The module will no longer display the 5G-SA band information by default. However, once the 5G-SA has been enabled, the band information will then be displayed.

Command examples

See these command examples using a router running IOS XE 17.13.1 with an FN980 modem.

```
router# show cellular 0/4/0 radio band
LTE bands supported by modem:
- Bands 2 4 5 12 14 26 29 30 46 48 66.
LTE band Preference settings for the active sim(slot 0):
- Bands 2 4 5 12 14 26 29 30 46 48 66.

NR5G NSA bands supported by modem:
- Bands 2 5 12 66 77.
NR5G NSA band Preference settings for the active sim(slot 0):
- Bands 2 5 12 66 77.

3G bands supported by modem:
Index: <none>
3G band Preference settings for the active sim(slot 0):
Index: <none>

==========================================
Band index reference list:

For LTE and 5G, indices 1-128 correspond to bands 1-128.

For 3G, indices 1-64 maps to the 3G bands mentioned against each above.
```

```
router#show cellular 0/4/0 radio hard
*Nov  8 12:13:31.969: Graphit 5G RSRP/RSRQ LTE modem:[1]
Modem Firmware Version = M0H.030202
Host Firmware Version = A0H.000302
Device Model ID =  FN980
International Mobile Subscriber Identity (IMSI) = 001010123456789
International Mobile Equipment Identity (IMEI) = 359661100035795
Integrated Circuit Card ID (ICCID) = 89860000502000180722
Mobile Subscriber Integrated Services
```

```
Digital Network-Number (MSISDN) =
Modem Status = Modem Online
Current Modem Temperature = 40 deg C
PRI version = 1080-114, Carrier = Generic GCF
OEM PRI version = 1080-114
```

```
router# show cellular 0/4/0 radio band
LTE bands supported by modem:
- Bands 1 2 3 4 5 7 8 12 13 14 17 18 19 20 25 26 28 29 30 32 34 38 39 40 41 42 43 46 48 66
 71.
LTE band Preference settings for the active sim(slot 0):
- Bands 1 2 3 4 5 7 8 12 13 14 17 18 19 20 25 26 28 29 30 32 34 38 39 40 41 42 43 46 48 66
 71.

NR5G NSA bands supported by modem:
- Bands 1 2 3 5 7 8 12 20 25 28 38 40 41 48 66 71 77 78 79.
NR5G NSA band Preference settings for the active sim(slot 0):
- Bands 1 2 3 5 7 8 12 20 25 28 38 40 41 48 66 71 77 78 79.

NR5G SA bands supported by modem:
- Bands <none>
NR5G SA band Preference settings for the active sim(slot 0):
- Bands <none>

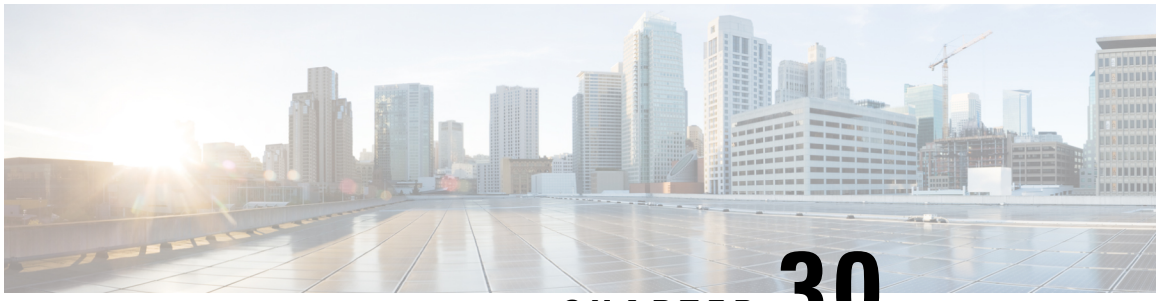
3G bands supported by modem:
Index:
  23 - UMTS Band 1: 2100 MHz (IMT)
  24 - UMTS Band 2: 1900 MHz (PCS A-F)
  26 - UMTS Band 4: 1700 MHz (AWS A-F)
  27 - UMTS Band 5: US 850 MHz  (CLR)
  50 - UMTS Band 8: 900 MHz  (E-GSM)
  51 - UMTS Band 9: Japan 1700 MHz
  61 - UMTS Band 19: 800 MHz (800 Japan)
3G band Preference settings for the active sim(slot 0):
Index:
  23 - UMTS Band 1: 2100 MHz (IMT)
  24 - UMTS Band 2: 1900 MHz (PCS A-F)
  26 - UMTS Band 4: 1700 MHz (AWS A-F)
  27 - UMTS Band 5: US 850 MHz  (CLR)
  50 - UMTS Band 8: 900 MHz  (E-GSM)
  51 - UMTS Band 9: Japan 1700 MHz
  61 - UMTS Band 19: 800 MHz (800 Japan)

==========================================

Band index reference list:

For LTE and 5G, indices 1-128 correspond to bands 1-128.

For 3G, indices 1-64 maps to the 3G bands mentioned against each above.
```



CHAPTER 30

Unified Threat Defence

- Unified Threat Defense (UTD), on page 331

Unified Threat Defense (UTD)

Unified Threat Defense (UTD) is Cisco's premier network security solution which provides a comprehensive suite of security features, such as:

- Enterprise Firewall
- IPS/IDS
- Advanced Malware Protection
- URL Filtering
- DNS Security

UTD is available on the IR1835 router.

IR1835 Limitations

The following are product specific limitations:

- UTD container requires a minimum space of 1.8 GB.
- UTD is supported in both Autonomous mode and Controller Mode, but in Autonomous mode, only IPS/IDS features are supported.
- The UTD configuration supports the Cloud-Low profile only.
- On-Box Web-Filtering Database is not supported.
- SSL proxy is not supported.

License and Supported Features

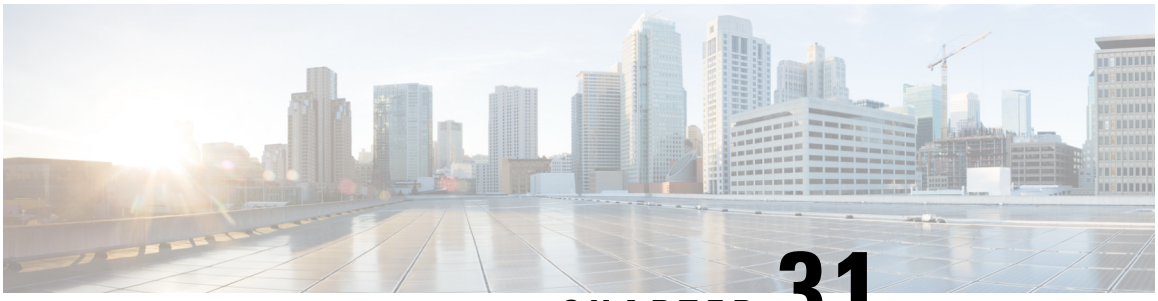
To enable UTD features the DNA Essentials license is required, in addition to Network Essentials. The license is required only in sd-router (autonomous mode).

If Cisco Secure Malware Analytics is also desired, then DNA Advantage license is required, in addition to Network Advantage.

Feature Configuration

Configuration on the IR1835 is the same as on other products. For information please refer to:

- Intrusion Prevention System
- URL Filtering
- Advanced Malware Protection



CHAPTER 31

Support for CAPWAP and WGB Modes on the Cisco Wi-Fi Interface Module

• **Support for CAPWAP and WGB Modes on the Cisco Wi-Fi Interface Module**, on page 333

Support for CAPWAP and WGB Modes on the Cisco Wi-Fi Interface Module

The Cisco Wi-Fi Interface Module (WIM) is a pluggable interface module available for all models of the IR1800 series. The Product Identifier (PID) is WP-WIFI6-x where x signifies the regulatory domain. For more information about WIM, see Cisco Wi-Fi Interface Module (WIM) Configuration Guide..

Cisco IOS XE Release 17.14.1 supports:

1. Switch operation mode between Control and Provisioning of Wireless Access Points (CAPWAP) and Workgroup Bridge (WGB).
2. Factory reset and erase configuration.
3. Configure the radios for WGB uplink and concurrent Root AP mode operations.

The following table summarises the management support for Wi-Fi Module operations in IR1800:

Modes	WIM IOS XE Release	Router IOSXE Release	Support
Control and Provisioning of Wireless Access Points (CAPWAP) Mode	17.11.0.155 and later	17.13.1 and later	Cisco Wireless LAN Controller.
Embedded Wireless Controller (EWC) Mode	17.11.0.155 and later	17.13.1 and later	IOS XE CLI vManage (SDWAN controller mode).
Work Group Bridge (WGB) Mode	17.11.0.155 and later	17.13.1	Cisco IoT Operations Dashboard.
		17.14.1	Cisco IoT Operations Dashboard. IOS XE CLI vManage (SDWAN and SD-Routing modes).

Management Support for Cisco WIM in CAPWAP Mode

- When operating in CAPWAP mode, the module functions as an Access Point managed by an external Cisco IOS XE Wireless LAN Controller, acquiring an IP address through DHCP and discovering the controller using Layer 3, DHCP, DNS, or IP subnet broadcast.
- Configuration of DHCP server and Switch Virtual Interface (SVI) on the router for WIM is required for the CAPWAP mode for WIM to discover and communicate with Wireless LAN controller.

Note The mode change from CAPWAP mode to WGB mode is supported only when the module is in its factory default configuration.

For more information, see Control And Provisioning of Wireless Access Points (CAPWAP).

Management Support for Cisco WIM in EWC Mode

- The Wi-Fi module acts as a Cisco IOS XE Wireless LAN Controller in Embedded Wireless Controller (EWC) mode, supporting configuration from IOS XE release 17.13.1.

Note The Wi-Fi module in EWC mode does not support changing to CAPWAP or WGB mode.

For more information, see Wireless LAN Controller.

For more information, see EWC Mode.

IPv6 Support

From Release 17.16.1a, vManage and Yang model supports both static and dynamic IPv6 configuration in WGB mode.

You can configure IPv6 on the WIM in WGB mode from the router to:

- obtain IPv6 addresses,
- collect metrics, and
- view configuration details.

Use the `show wireless-bridge clients`, `show wireless-bridge interface ipv4`, and `show wireless-bridge interface ipv6` commands on the router to view client IPv4 and IPv6 configuration details.

For more information on configuring IPv6 address, see Configure IPv6 address.

Management Support for Cisco WIM in WGB Mode

The following section describes the new configuration options available on IR1800 for Deploying Cisco Wi-Fi Interface Module in WGB mode.

Configuring IR1800 for deploying WGB

The following section show how to configure IR1800 for deploying WGB:

Configuring a QoS Profile

Procedure

	Command or Action	Purpose
Step 1	**enable** **Example:** `router# enable`	Enables privileged EXEC mode.
Step 2	**configure terminal** **Example:** `router# configure terminal`	Enters global configuration mode.
Step 3	**wireless-bridge submode** **Example:** `router(config)# wireless-bridge`	Enters wireless-bridge configuration mode.
Step 4	**qos-profile** *qos-profile-name* **{bronze | gold | platinum | silver }** **Example:** `router(config-wl-bridge)# qos-profile test-qos-profile bronze`	Create a QoS profile with one of the levels of QoS policy.
Step 5	**End** **Example:** `router(config-wl-bridge)# end`	Exits wireless-bridge configuration mode and returns to privilege EXEC mode.

Configuring an SSID Profile With Open Authentication Without a QoS Profile Mapped

Procedure

	Command or Action	Purpose
Step 1	**enable** **Example:**	Enters privileged EXEC mode.

	Command or Action	Purpose
	`router# enable`	
Step 2	**configure terminal** **Example:** `router# configure terminal`	Enters global configuration mode.
Step 3	**wireless-bridge submode** **Example:** `router(config)# wireless-bridge`	Enters wireless-bridge configuration mode.
Step 4	**ssid-profile** *ssid-profile-name* **ssid** *ssid-name* | **authentication open** **Example:** `router(config-wl-bridge)# ssid-profile test-ssid-profile ssid test-ssid authentication open`	Create SSID profile with open authentication.
Step 5	End **Example:** `router(config-wl-bridge)# end`	Exits wireless-bridge configuration mode and returns to privilege EXEC mode.

Configuring an SSID Profile With Open Authentication With a QoS Profile Mapped

Procedure

	Command or Action	Purpose
Step 1	**enable** **Example:** `router# enable`	Enters privileged EXEC mode.
Step 2	**configure terminal** **Example:** `router# configure terminal`	Enters global configuration mode.

	Command or Action	Purpose
Step 3	**wireless-bridge submode** **Example:** `router(config)# wireless-bridge`	Enters wireless-bridge configuration mode.
Step 4	**ssid-profile** *ssid-profile-name* **ssid** *ssid-name* **qos-profile** *qos-profile-name* **authentication open** **Example:** `router(config-wl-bridge)# ssid-profile test-ssid-profile ssid test-ssid qos-profile test-qos-profile authentication open`	Create SSID profile with open authentication with a QoS-profile mapped.
Step 5	End **Example:** `router(config-wl-bridge)# end`	Exits wireless-bridge configuration mode and returns to privilege EXEC mode.

Configuring an SSID Profile with WPA2 Personal Authentication Without a QoS Profile Mapped

Procedure

	Command or Action	Purpose
Step 1	**enable** **Example:** `router# enable`	Enters privileged EXEC mode.
Step 2	**configure terminal** **Example:** `router# configure terminal`	Enters global configuration mode.
Step 3	**wireless-bridge submode** **Example:** `router(config)# wireless-bridge`	Enters wireless-bridge configuration mode.

	Command or Action	Purpose
Step 4	**ssid-profile** *ssid-profile-name* **ssid** *ssid-name* **authentication psk key-management wpa2 secret-key {0 | 6 | 7 }** *secret-key* **Example:** `router(config-wl-bridge)# ssid-profile test-ssid-profile ssid test-ssid authentication psk key-management wpa2 secret-key 0 testkey123!`	Create SSID profile with PSK authentication. • 0: Specifies an unencrypted secret key will follow. • 6: Specifies an encrypted secret key will follow. • 7: Specifies a hidden secret key will follow.
Step 5	End **Example:** `router(config-wl-bridge)# end`	Exits wireless-bridge configuration mode and returns to privilege EXEC mode.

Configuring an SSID Profile with WPA2 Personal Authentication With a QoS Profile Mapped

Procedure

	Command or Action	Purpose
Step 1	**enable** **Example:** `router# enable`	Enters privileged EXEC mode.
Step 2	**configure terminal** **Example:** `router# configure terminal`	Enters global configuration mode.
Step 3	**wireless-bridge submode** **Example:** `router(config)# wireless-bridge`	Enters wireless-bridge configuration mode.
Step 4	**ssid-profile** *ssid-profile-name* **ssid** *ssid-name* **qos-profile** *qos-profile-name* **authentication psk key-management wpa2 secret-key {0 | 6 | 7 }** *secret-key* **Example:** `router(config-wl-bridge)# ssid-profile test-ssid-profile ssid test-ssid`	Create SSID profile with PSK authentication with QoS profile mapped. • 0: Specifies an unencrypted secret key will follow. • 6: Specifies an encrypted secret key will follow.

	Command or Action	Purpose
	`qos-profile qos-profile-test authentication psk key-management wpa2 secret-key 0 testkey123!`	• 7: Specifies a hidden secret key will follow.
Step 5	End **Example:** `router(config-wl-bridge)# end`	Exits wireless-bridge configuration mode and returns to privilege EXEC mode.

Configuring a Dot11radio in WGB Mode and Configuring Various Parameters

Procedure

	Command or Action	Purpose
Step 1	**enable** **Example:** `router# enable`	Enters privileged EXEC mode.
Step 2	**configure terminal** **Example:** `router# configure terminal`	Enters global configuration mode.
Step 3	**wireless-bridge submode** **Example:** `router(config)# wireless-bridge`	Enters wireless-bridge configuration mode.
Step 4	**dot11radio {0|1} mode {wgb } ssid-profile** *ssid-profile name* **Example:** `router(config-wl-bridge)# dot11radio 1 mode wgb ssid-profile test-ssid-profile`	Configure a Dot11Radio as WGB.
Step 5	**dot11radio {0|1} {enable | disable }** **Example:** `router(config-wl-bridge)# dot11radio 1 enable`	Enabling Dot11Radio.

	Command or Action	Purpose
Step 6	**dot11radio {0|1} channel** *channel number channel-width* **Example:** `router(config-wl-bridge)# dot11radio 1 channel 40 40`	Configure a Dot11Radio channel details.
Step 7	End **Example:** `router(config-wl-bridge)# end`	Exits wireless-bridge configuration mode and returns to privilege EXEC mode.

Configuring a Dot11Radio in uWGB Mode and Configuring Various Parameters

Procedure

	Command or Action	Purpose
Step 1	**enable** **Example:** `router# enable`	Enables privileged EXEC mode.
Step 2	**configure terminal** **Example:** `router# configure terminal`	Enters global configuration mode.
Step 3	**wireless-bridge submode** **Example:** `router(config)# wireless-bridge`	Enters wireless-bridge configuration mode.
Step 4	**dot11radio {0|1} mode {uwgb }** *H.H.H* **ssid-profile** *ssid-profile name* **Example:** `router(config-wl-bridge)# dot11radio 1 mode uwgb E462.C49F.9AA0 ssid-profile test-ssid-profile`	Configure a Dot11Radio as uWGB.
Step 5	**dot11radio {0|1} {enable | disable }** **Example:**	Enabling Dot11Radio.

	Command or Action	Purpose
	`router(config-wl-bridge)# dot11radio 1 enable`	
Step 6	**dot11radio {0|1} channel** *channel number channel-width* **Example:** `router(config-wl-bridge)# dot11radio 1 channel 40 40`	Configure a Dot11Radio channel details.
Step 7	**End** **Example:** `router(config-wl-bridge)# end`	Exits wireless-bridge configuration mode and returns to privilege EXEC mode.

Configuring a Dot11radio in Root AP Mode and Configuring Various Parameters

Procedure

	Command or Action	Purpose
Step 1	**enable** **Example:** `router# enable`	Enables privileged EXEC mode.
Step 2	**configure terminal** **Example:** `router# configure terminal`	Enters global configuration mode.
Step 3	**wireless-bridge submode** **Example:** `router(config)# wireless-bridge`	Enters wireless-bridge configuration mode.
Step 4	**dot11radio {0|1} mode {root-ap}** **Example:** `router(config-wl-bridge)# dot11radio 0 mode root-ap`	Configure a Dot11Radio as Root AP. The Root AP places the bridge in access point mode. In this mode, the bridge emulates a Cisco Aironet 1100 Series Access Point and accepts associations from client devices.
Step 5	**dot11radio {0|1} {enable | disable }** **Example:** `router(config-wl-bridge)# dot11radio 0 enable`	Enabling Dot11Radio.

	Command or Action	Purpose
Step 6	**dot11radio {0|1} wlan** *wlan-profile-name wlan-id (2-16)* **vlan** *vlan-id (2-4094)* **Example:** `router(config-wl-bridge)# dot11radio 0 wlan test-wlan-profile 4 vlan 400`	Map a WLAN profile to the Dot11Radio in Root AP mode.
Step 7	**dot11radio {0|1} channel** *channel number channel-width* **Example:** `router(config-wl-bridge)# dot11radio 0 channel 5 20`	Configure a Dot11Radio channel details.
Step 8	**End** **Example:** `router(config-wl-bridge)# end`	Exits wireless-bridge configuration mode and returns to privilege EXEC mode. **Note** 1.The above command bridges VLAN creation in the client serving radio to wired0, forwarding wireless client traffic directly to the router. 2.WLAN IDs range from 2 to 16 (supporting a maximum of 15 WLANs). Configurations related to the Root AP will take effect only after toggling the Root AP radio. 3.Enabling Broadcast tagging in WGB will prevent the Root AP from supporting wireless client connections. Broadcast tagging configuration is disabled by default.

Configure IPv4 address

Use this task to configure the IPv4 address on the WIM in WGB mode from the router.

Procedure

Step 1 Use the **enable** command to enable the privileged EXEC mode.

```
router#enable
```

Step 2 Use the **configure terminal** command to enter the global configuration mode.

```
router#configure terminal
```

Step 3 Use the **wireless-bridge** command to enter the wireless-bridge configuration mode.

```
router(config)#wireless-bridge
```

Step 4 Use the **wgb address ipv4 static** *ipaddress netmask gateway* command to configure a static IPv4 address along with the netmask and the default gateway.

```
router(config-wl-bridge)# wgb address ipv4 static 10.10.10.2 255.255.255.0 10.10.10.1
```

Step 5 Use the **wgb address ipv4 dhcp** command to configure DHCP IPv4 address.

```
router(config-wl-bridge)#wgb address ipv4 dhcp
```

Configure IPv6 address

Use this task to configure the IPv6 address on the WIM in WGB mode from the router.

Procedure

Step 1 Use the **enable** command to enable the privileged EXEC mode.

```
router#enable
```

Step 2 Use the **configure terminal** command to enter the global configuration mode.

```
router#configure terminal
```

Step 3 Use the **wireless-bridge** command to enter the wireless-bridge configuration mode.

```
router(config)#wireless-bridge
```

Step 4 Use the **wgb address ipv6 static** *ipaddress netmask gateway* command to configure a static IPv6 address along with the netmask and the default gateway.

```
router(config-wl-bridge)# wgb address ipv6 static 2000:100::10 64  2000:100::1
```

Step 5 Use the **wgb address ipv6 dhcp** command to configure DHCP IPv6 address.

```
router(config-wl-bridge)#wgb address ipv6 dhcp
```

Verify the WGB Mode Configuration, Monitoring Operational Status

Use the following commands to verify the configuration status:

Command: show run-config | sec wireless-bridge

Example:

```
router#show run-config | sec wireless-bridge
```

Use the following commands for monitoring the operational status:

Command: show wireless-bridge status

Example:

```
router#show wireless-bridge status
Module Operating Mode : WGB mode
Module Status         : Module State Ready
```

```
Software Version      : 17.11.0.155
Module Session Status : Login Success
```

Command: show wireless-bridge wlans

Example:

```
router#show wireless-bridge wlans
 wlan  band  oper  vlan  #client  wlan-mode  SSID
 ----  ----  ----  ----  -------  ---------  -----------------
    2  2.4g    up     2        1   downlink  myssid
```

Command: show wireless-bridge clients

Example:

From Release 17.16.1a, Cisco WIM introduces support for IPv6, and both IPv4 and IPv6 addresses are visible in the output.

```
router#show wireless-bridge clients
 Client-MAC-Addr    band  status        wlan  DeviceType  SSID       IPV4 address      IPV6
 address
 -----------------  ----  ------------  ----  ----------  ---------- ----------------
--------------------------
 E4:62:C4:9F:68:CF    5g  Associated      3   wireless    000_bba    30.30.30.2
FE80::EDA6:F758:9C25:8539
```

Note vManage does not support to view client IPv4 and IPv6 configuration details.

Command: Use the **show wireless-bridge interface ipv4** command to view the IPv4 interface details.

Example:

```
router#show wireless-bridge interface ipv4
Interface            IP-Address      Method   Status                 Protocol   Speed
 Duplex
wired0               unassigned      unset    up                     up         1000
 full
auxiliary-client     20.20.20.5      DHCP     up                     up         n/a
 n/a
service-vlan         n/a             n/a      up                     up         n/a
 n/a
apr0v0               n/a             n/a      up                     up         n/a
 n/a
apr1v0               n/a             n/a      up                     up         n/a
 n/a
```

Command: Use the **show wireless-bridge interface ipv6** command to view the IPv6 interface details.

Example:

```
router#show wireless-bridge interface ipv6
srcr2     Link encap:Ethernet  HWaddr E4:62:C4:9F:52:40
          inet addr:20.20.20.5  Bcast:20.20.20.255  Mask:255.255.255.0
          inet6 addr: 3000:100::9d98:c2e:5a8f:1351/128 Scope:Global
          inet6 addr: 3000:100::c132:d20b:b520:3537/128 Scope:Global
          inet6 addr: fe80::18c:5b82:4cef:c56c/64 Scope:Link
          inet6 addr: 3000:100::5996:1093:a134:ea8c/128 Scope:Global
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:5042 errors:0 dropped:0 overruns:0 frame:0
          TX packets:4841 errors:0 dropped:0 overruns:0 carrier:0
```

```
          collisions:0 txqueuelen:0
          RX bytes:565722 (552.4 KiB)  TX bytes:556927 (543.8 KiB)
```

Command: Use the **show wireless-bridge running-configuration** command to view the configuration and operational status

Example:

```
router#show wireless-bridge running-configuration
### WGB Running config - Hostname: device ###

configure ap management add username Cisco1 password $1$$r0JYqf7z0bIzm/v2jcqVp/ secret
$1$$cN9KzIWhwzaBqoL.xelgi1
configure ap address ipv6 dhcp
configure qos profile qos-profile1 gold
configure ssid-profile wlan2 ssid 000_aab authentication open
configure ssid-profile wlan3 ssid 000_bba authentication open
configure dot11Radio 0 mode wgb ssid-profile wlan2
configure dot11Radio 0 enable
configure dot11Radio 1 enable
configure dot11Radio 1 mode root-ap
configure dot11Radio 1 wlan add wlan3 3 vlan 30
configure dot11Radio 1 channel 36 80
configure dot11Radio 1 tx-power 1
```

For additional Information about WGB and Universal Workgroup Bridge (uWGB) configuration, refer the following documents:

- Workgroup Bridge (WGB)
- Cisco Industrial Wireless Workgroup Bridge and Universal WGB Deployment Guide

Additional Commands

Clear Configuration

To clear the configuration on the Wi-Fi module, use the following command:

```
router# wireless-bridge erase
```

Factory Reset

To perform a factory reset on the module, use the following command:

```
router# wireless-bridge factory reset config/default
```

Mode Conversion

To change the operating mode of the module between WGB and CAPWAP modes, use the following command:

```
router# wireless-bridge boot mode capwap/wgb
```

Firmware Upgrade

Firmware upgrade is supported from Unified Client Image version 17.11 and above when running in WGB mode. To upgrade the firmware, the IR1800 requires the TFTP server to be enabled for the module to obtain the image.

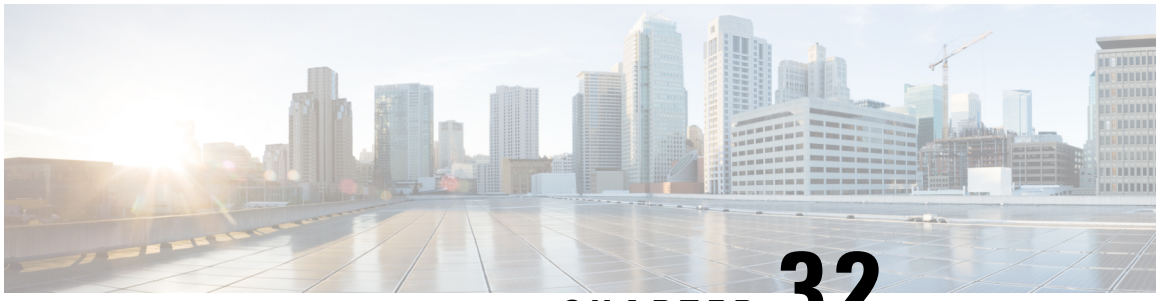
The firmware upgrade process takes about 5-6 minutes to complete. Upon successful upgrade, the Wi-Fi module is automatically reloaded with the new image.

Procedure

	Command or Action	Purpose
Step 1	**enable** **Example:** `router# enable`	Enters privileged EXEC mode.
Step 2	**copy rcp://***local-server*/**ap1g8t-k9c1-tar-k9c1-tar** **Example:** `router# copy rcp://netadmin@172.16.101.101/ap1g8t-k9c1-tar bootflash:ap1g8t-k9c1-tar`	Copy the unified client image(ap1g8t-k9c1-tar) to IR1800. The image will be downloaded to the module from this location.
Step 3	**configure terminal** **Example:** `router# configure terminal`	Enters global configuration mode.

Step 4	**tftp-server** *directory*:*image*
	Example:
	`router# tftp-server bootflash:ap1g8t-k9c1-tar`
	`router# end`
		Configure the image location on the TFTP Server of IR1800 and exit from global configuration mode.

Step 5	**wireless-bridge firmware-upgrade** *ap1g8t-k9c1-tar ip netmask gateway* **Example:** `router(config)# wireless-bridge firmware-upgrade ap1g8t-k9c1-tar 10.10.10.1 255.255.255.0 10.10.10.1`	Start Firmware upgrade.
Step 6	**more bootflash:/od_status**	Firmware upgrade takes about 5 to 6 minutes to complete. Check the **od_status** logs to monitor the progress.



CHAPTER 32

Additional Modem Support for Cellular Pluggable Modules

- Additional Modem Support for CAT 6 and CAT 7 Cellular Pluggable Modules, on page 347
- Additional Modem Support for Cellular Pluggable Modules, on page 348
- 5G Standalone Mode (SA) Support, on page 348

Additional Modem Support for CAT 6 and CAT 7 Cellular Pluggable Modules

The LTE Cat6 Pluggable Interface Modules (PIMs) will be updated with Cat7 modems. The following table shows the product transition:

Table 24: Cat6 to Cat7 Transition

Cat6 (Current)	Cat7 (Refreshed)
Sierra Wireless EM7455/7430	Sierra Wireless EM7411/7421/7431
Cat6 LTE Advanced	Cat7 LTE Advanced

Support for additional modems on the IR1101

The following are the new PIDs that will be available:

- P-LTEA7-NA
- P-LTEA7-EAL
- P-LTEA7-JP
- P-5GS6-R16SA

Support for additional modems on the IR1800

The following are the new PIDs that will be available:

- P-LTEA7-NA

- P-LTEA7-EAL
- P-LTEA7-JP

Important For the new PIDs mentioned above, the following cellular functions have not been tested, and are not supported with IOS XE release 17.13.1 although the CLI commands may permit:

- GNSS/NMEA
- Cellular Dying-Gasp
- eSIM/eUICC support

Note There is no new or changed command line interface with these new modems.

Additional Modem Support for Cellular Pluggable Modules

Cisco IOS-XE Release 17.14.1 enhances connectivity options and throughput on the IR1101 and IR1800 platforms by supporting additional cellular modems:

- CAT 7 Modems:
  - P-LTEA7-NA
  - P-LTEA7-EAL
  - P-LTEA7-JP
- 5G Modem:
  - P-5GS6-R16SA-GL

Note CAT 7 modems support GNSS and NMEA streaming, while currently P-5GS6-R16SA-GL module does not support GPS and NMEA streaming.

5G Standalone Mode (SA) Support

This feature provides 5G Standalone mode (SA) support on the P-5GS6-GL pluggable module. The 5G SA mode support will enable 5G cellular configuration display using Cisco IOS-XE CLI commands.

This feature provides a mechanism in the CLI to select a set of bands for SA mode, as opposed to a single band in previous software releases. The following IOS-XE CLIs have been modified for 5G SA mode support:

- show cellular radio

- show cellular radio details (without carrier aggregation)
- show cellular network

There is also a band selection CLI to select cellular bands.

Show Command Examples

```
Router#show cellular 0/2/0 radio
Radio power mode = Online
5G Rx Channel Number = 632544
5G Tx Channel Number = 632544
5G-SA Band = 78
Bandwidth = 20 MHz
Current 5G RSSI = -60 dBm
Current 5G RSRP = -71 dBm
Current 5G RSRQ = -11 dB
Current 5G SNR = 34.5 dB
Physical Cell Id = 500
Radio Access Technology(RAT) Preference = AUTO
Radio Access Technology(RAT) Selected = 5GNR-SA
```

```
Router#show cellular 0/4/0 radio detail
 Modem Radio is Online
 Main 0 Antenna details:
 RSSI = -38 dBm
 RSRP = -48 dBm
 Diversity 0 Antenna details:
 RSSI = -47 dBm
 RSRP = -58 dBm
```

```
Router#show cellular 0/4/0 network
Current System Time = Sun Jan 6 0:4:36 1980
Current Service Status = Normal
Current Service = Packet switched
Current Roaming Status = Home
Network Selection Mode = Automatic
Network = Test PLMN 1-1
Mobile Country Code (MCC) = 1
Mobile Network Code (MNC) = 1
Packet switch domain(PS) state = Attached
Tracking Area Code (TAC) = 1
Cell ID = 1024
Negotiated network MTU = 1500
```

Band Selection Command Example

The **lte modem band-select** CLI can be used to enable bands that the user wishes to use and subscribe to. By default SA bands are not available.



Important If you wish to use SA bands, the **lte modem band-select** command **MUST** be used as part of the configuration.

The following is an example of the command:

```
conf t
controller cellular 0/2/0
lte modem band-select indices umts3g all lte4g all nr5g-NSA all nr5g-SA 78 slot 0
exit
```

The following shows an example of using nr5g-sa band 48:

```
lte modem band-select indices umts3g "23" lte4g "7" nr5g-nsa "12" nr5g-sa "48" slot 0
```

Note In the above example, umts3g band 23, lte4g band 7, and nr5g-nsa band 12 are not available in the area, which means the modem will only attach to nr5g-nsa band 48.

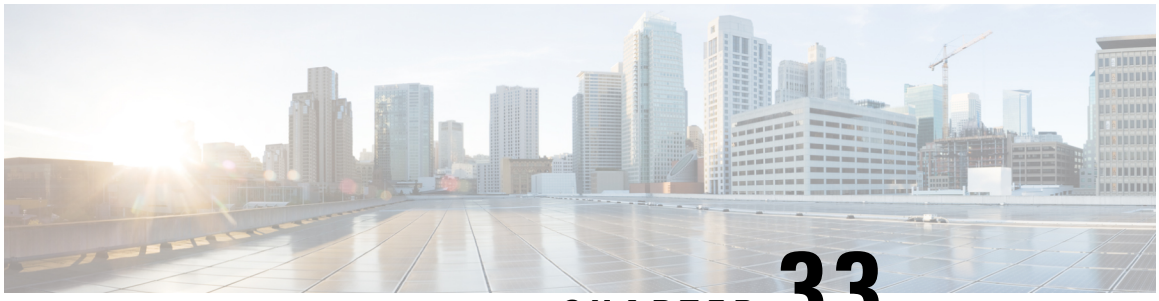
The following shows an example of using nr5g-sa band 78:

```
lte modem band-select indices umts3g "23" lte4g "7" nr5g-nsa "78" nr5g-sa "none" slot 0
```

Note In the above example, umts3g band 23 and lte4g band 7 are not available in the area, and nr5g-sa bands are turned off which means the modem will only attach to nr5g-nsa band 78.

Limitations

none is an invalid option for lte4g, and nr5g-nsa.



CHAPTER 33

Support for P-LTE-450 Pluggable Interface Module

- Support for P-LTE-450 Pluggable Interface Module, on page 351
- Enhancing P-LTE-450 Network Management Through SNMP MIB Objects, on page 352
- VRF for the P-LTE-450 Module, on page 353
- Upgrade the P-LTE-450 PIM, on page 355

Support for P-LTE-450 Pluggable Interface Module

The Cisco Catalyst IR1800 Rugged Series Router now supports the 450MHz category P Long-Term Evolution (LTE) Pluggable Interface Module (PIM), referred to as P-LTE-450. The LTE-450 PIM uses the 450MHz frequency to address LTE use cases primarily targeting utility, public safety, and critical infrastructure maintained by public organizations in Europe and other countries. The module supports only Band 31 and 72, which are the LTE operating frequency bands. Due to hardware limitations, P-LTE-450 is supported only on slot 0/4.

Unlike regular LTE modules, the P-LTE-450 has some differences on the IOS-XE platform. Some of the key differences are:

- IP pass through is on Gigabit Ethernet interfaces rather than cellular interfaces.
- Troubleshooting commands are from the Web user interface of third-party hardware.

Note

- Throughout the user documentation, you can see the module referred to as P-LTE-450, the Cisco product name. The module is designed and manufactured by Intelliport, which refers to it as the IPS-701. Both names are present in the documentation.
- Starting with Cisco IOS XE 26.1.1, the **dhcp-default-gateway assign** and **dhcp-default-gateway dont-assign** CLI commands are supported on P-LTE-450 sub-interfaces.

The show command output displays interface details:

```
router#show run | sec 0/4/0
interface GigabitEthernet0/4/0
ip address dhcp
negotiation auto
```

```
ipv6 dhcp client request vendor
ipv6 address autoconfig
ipv6 enable
interface GigabitEthernet0/4/0.2
encapsulation dot1Q 2
ip address dhcp
ipv6 dhcp client request vendor
ipv6 address autoconfig


router#show ip int br
Interface IP-Address OK? Method Status Protocol
GigabitEthernet0/0/0 10.195.236.86 YES NVRAM up up
GigabitEthernet0/1/0 unassigned YES unset down down
GigabitEthernet0/1/1 unassigned YES unset down down
GigabitEthernet0/1/2 unassigned YES unset up up
GigabitEthernet0/1/3 unassigned YES unset down down
Wl0/1/4 unassigned YES unset up up
GigabitEthernet0/4/0 192.168.200.194 YES DHCP up up
GigabitEthernet0/4/0.2 192.168.4.6 YES DHCP up up
GigabitEthernet0/4/0.3 192.168.5.2 YES DHCP up up
Async0/2/0 unassigned YES unset up down
Vlan1 192.168.50.1 YES NVRAM up up
```

For more information on how to configure the router interface for the P-LTE-450 module, See

- Configuring the Router Interface for the P-LTE-450 Module
- Cellular Pluggable Interface Module Configuration Guide
- LTE 450MHz Alliance

Enhancing P-LTE-450 Network Management Through SNMP MIB Objects

In P-LTE-450 networks, routers can provide crucial quality metrics through Simple Network Management Protocol (SNMP) Management Information Base (MIB) objects. Specifically, Cisco IR1101 and IR1800 routers equipped with the P-LTE-450 Pluggable Interface Module (PIM) allows you to integrate seamlessly with existing SNMP-based management tools to report their status and quality metrics.

The router identifies the P-LTE-450 module as an Ethernet interface, specifically Gigabit Ethernet 0/1/0, rather than a cellular interface.. For LTE450 0/1/0 radio parameters to be supported through SNMP, a new MIB file named **Cisco-LTE450-MIB.my** needs to be created. This MIB file includes specific fields to capture radio signal quality metrics.

Supported SNMP MIBs

The **CISCO-LTE450-MIB** SNMP is supported on **Cisco P_ LTE_450 MIB**.

CISCO-LTE450-MIB supports the following MIB fields:

- `**CiscoLte450Entry**`: Main entry point for the MIB
- `**ciscoLte450PortIndex**`: Index for the port (GigabitEthernet0/1/0)
- `**ciscoLte450Rssi**`: RSSI value
- `**ciscoLte450Rsrp**`: RSRP value

- `**ciscoLte450Rsrq**`: RSRQ value
- `**ciscoLte450Sinr**`: SINR value
- `**ciscoLte450RssiMain**`: RSSI value for the main antenna
- `**ciscoLte450RsrpMain**`: RSRP value for the main antenna
- `**ciscoLte450RssiAux**`: RSSI value for the auxiliary antenna
- `**ciscoLte450RsrpAux**`: RSRP value for the auxiliary antenna

View Radio Signal Quality Metrics

The following example displays the signal quality and antenna signal quality of SNMP Cisco 4G LTE 450 configuration:

```
Router#show lte450 0/1/0 radio
Signal quality
==============
RSSI = -88 dBm
RSRP = -114 dBm
RSRQ = -12 dB
SINR = 0 dB

Antenna signal quality
======================
Main RSSI = -87 dBm
Main RSRP = -112 dBm
Aux RSSI = -104 dBm
Aux RSRP = -143 dBm
```

VRF for the P-LTE-450 Module

Virtual Routing Forwarding (VRF) for the LTE450 module creates a separate routing table. This isolates LTE450 traffic, such as monitoring, management, or dedicated data flows. It ensures segregation from other network services on the device.

Starting from IOS-XE 17.18.1a, you can enable VRF to provision LTE450 parameters through the LTE450 management interface. You can enable VRF on 450 PIM management interface to enable HTTP request from router to the webserver of P-LTE-450, and get response from the webserver of P-LTE-450.

Table 25: Feature History Table

Feature	Release Information	Feature Description
VRF for the P-LTE-450 Module	IOS-XE 17.18.1a	You can now enable VRF to provision LTE450 parameters through the LTE450 management interface. You can also enable VRF on 450 PIM management interface to enable HTTP request from router to the webserver of P-LTE-450 and get response from the webserver of P-LTE-450. **Note** The IR1800 router supports the P-LTE-450 module only in slot 4. The IR1101 supports this module in the Base or in the IRM-1100 Expansion Module attached through the Compute Module slot (which is located on the bottom side of the IR1101 Base).

Configure VRF for the P-LTE-450 module

Perform these steps to configure VRF for the P-LTE-450 module.

Procedure

Step 1 Use the **configure** command to enter the global configuration mode.

Example:

```
Router# configure
```

Step 2 Use the **vrf definition** command to create a VRF.

Example:

```
Router(config)# vrf definition lte450_mgmt
```

Step 3 Use the **address-family** command to to set the address family to either IPv4 or IPv6 for traffic routing.

Example:

```
Router(config-vrf-lte450_mgmt)# address-family ipv4
```

Step 4 Use the **exit-address-family** command to exit from the address family configuration mode.

Example:

```
Router(config-vrf-lte450_mgmt-address-family)# exit-address-family
```

Step 5 Use the **exit** command to exit from the VRF configuration mode.

Example:

```
Router(config-vrf-lte450_mgmt)# exit
```

Step 6 Use the **interface** command to configure LTE450 management interface.

Example:

```
Router(config)# interface GigabitEthernet0/1/0
```

Step 7 Use the **description** command to add a description to the interface.

Example:

```
Router(config-if)# description LTE450 Management Interface
```

Step 8 Use the **vrf forwarding** command to associate the LTE 450 management interface with a VRF for traffic routing.

Example:

```
Router(config-if)# vrf forwarding lte450_mgmt
```

Step 9 Use the **ip address dhcp** command for the main interface to retrieve the management address.

Example:

```
Router(config-if)# ip address dhcp
```

Step 10 Use the **negotiation auto** command to configure auto-negotiation to handle different speeds on hardware.

Example:

```
Router(config-if)# negotiation auto
```

Step 11 Use the **lte450 band** command to configure the LTE 450 MHz modem band value.

Example:

```
Router(config-if)# lte450 band 31
/* Possible values are 31 or 72*/
```

Step 12 Use the **end** commad to save and exit the configuration.

Example:

```
Router(config-if)# end
```

Upgrade the P-LTE-450 PIM

Use this task to upgrade the P-LTE-450 PIM firmware on the Cisco Catalyst IR1101 router.

Before you begin

- Ensure the firmware image file is successfully copied to the router's flash memory before initiating the upgrade.

- You can use the **hw-module** command to trigger the firmware update process for the specific PIM slot and subslot.
- Firmware downgrade is not supported by the P-LTE-450 module software.
- You must set the username and password on the module before you perform the firmware upgrade. The credentials can be found on the sticker attached to the module.
- The PIM and Modem firmware can be found in the intelliport website.

Procedure

Step 1 Use the **upgrade hw-module subslot** <*slot*> / <*subslot*> **lte450 pim bootflash:** / <*filename*> command for executing firmware upgrade of P-LTE-450 PIM.

Example:

```
upgrade hw-module subslot 0/1 lte450 pim bootflash:/pim-firmware-filename
```

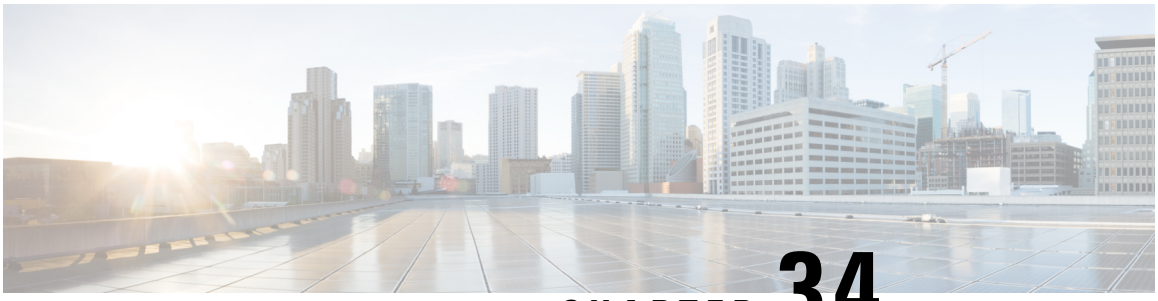
Step 2 Use the **upgrade hw-module subslot** <*slot*> / <*subslot*> **lte450 modem bootflash:** / <*filename*> command for executing firmware upgrade of modem.

Example:

```
upgrade hw-module subslot 0/1 lte450 modem bootflash:/modem-firmware-filename
```

Note
These delays occur during the upgrade proceedure till it completes:

- Firmware upgrade: The upgrade process typically takes 5–7 minutes to complete.
- Automatic reload: Following the upgrade, the module automatically initiates a reload, which requires an additional 3–5 minutes.
- For post-upgrade delay: Once the module is online there is an additional delay of 2–3 minutes before getting the router prompt.
- Total wait time: The total wait time from the start to end of the upgrade procedure is 10-12 minutes.
- Once PIM module is up, check FW **sh lte450 0/x/0 hardware**.



CHAPTER 34

Support for P-5GS6-RC-GL PIM

- Support for P-5GS6-RC-GL PIM, on page 357

Support for P-5GS6-RC-GL PIM

P-5GS6-RC-GL is a 3GPP Pluggable Interface Module (PIM) designed for low-performance and cost-sensitive 5G applications. It utilizes the Cisco EM8695 Reduced Capability (RedCap) cellular modem to provide 5G-SA RedCap and 4G LTE connectivity.

The P-5GS6-RC-GL PIM provides an affordable, future-proof 5G connectivity solution for primary or secondary WAN applications, maintaining full feature parity with other existing Cisco 5G PIMs and routers.

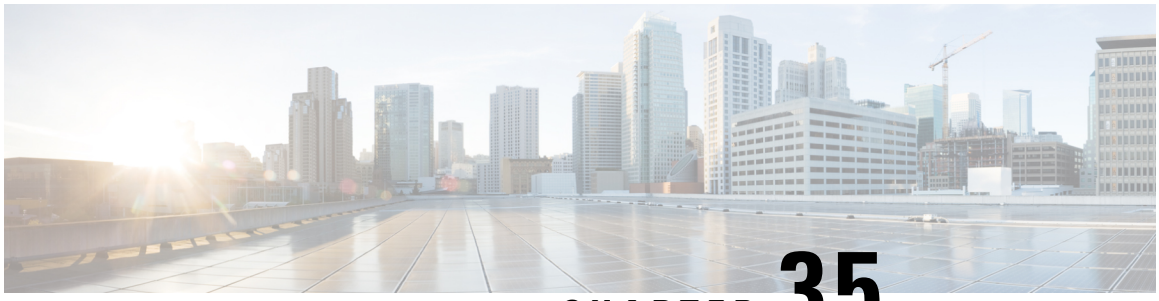
Benefits of P-5GS6-RC-GL PIM

The benefits of using the P-5GS6-RC-GL PIM include:

- It delivers 4G LTE and 5G-SA RedCap connectivity suited for low-performance requirements.
- It provides a reliable migration path for customers as cellular networks evolve.
- It allows existing users to upgrade to 5G-SA RedCap technology without changing operational workflows or management interfaces.
- It addresses the need for affordable, long-term cellular connectivity.

Note The P-5GS6-RC-GL PIM provides only 5G-SA RedCap connectivity when offered by cellular base stations, not 5G NSA or SA. It falls back to 4G when 5G-SA RedCap is not available.

Support for P-5GS6-RC-GL PIM



CHAPTER 35

Cellular Boot Time Improvements

• Cellular boot time improvements, on page 359

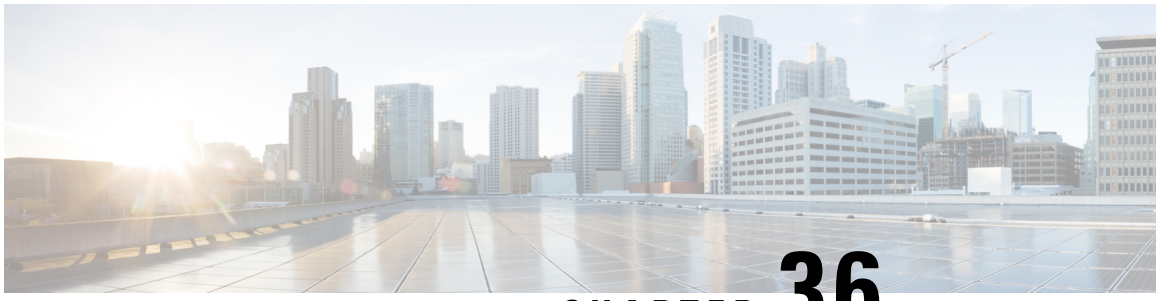
Cellular boot time improvements

Cellular boot time improvements are performance enhancements that

- reduce the time required for cellular interfaces to become operational after router startup
- decrease cellular link up-time by approximately 20% compared to previous releases, and
- enable faster traffic passing capabilities following system initialization.

Performance improvements

In previous releases, the cellular interface was taking approximately two and a half minutes to come up and pass traffic after the router booted up. The cellular link up-time has been improved by approximately 20% in IOS-XE release 17.9.1.



CHAPTER 36

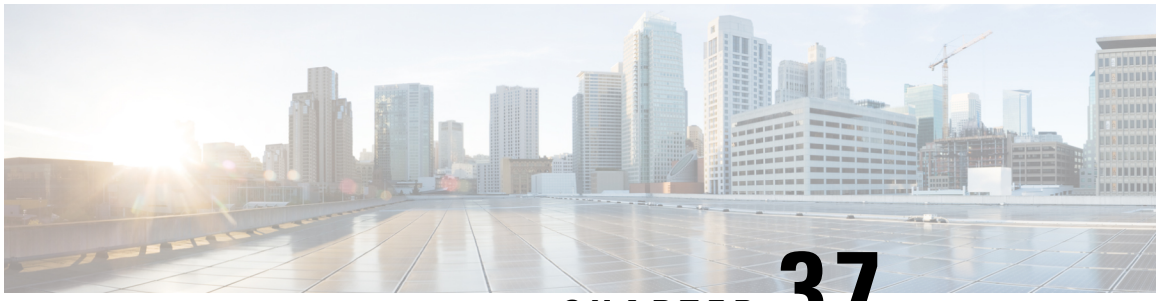
Digital Subscriber Line (DSL) SFP Support on the IR1800

• Digital Subscriber Line (DSL) SFP Support on the IR1800, on page 361

Digital Subscriber Line (DSL) SFP Support on the IR1800

The IR1800 now supports the DSL SFP in the same manner as the IR1101. For complete details, see the Chapter: Digital Subscriber Line in the IR1101 Configuration Guide.

Digital Subscriber Line (DSL) SFP Support on the IR1800



CHAPTER 37

LoRaWAN Pluggable Interface Module Support

This chapter contains the following sections:

- LoRaWAN Pluggable Interface Module Support, on page 363

LoRaWAN Pluggable Interface Module Support

These release adds support for the LoRaWAN Pluggable Interface Module which was first available on the IR1101.

Note This is a software parity release only. The LoRaWAN Pluggable Interface Module is neither orderable or hardware deployment ready for the IR1800 until the product is announced. Please reach out to your Cisco contact for any additional info.

The Cisco LoRaWAN Pluggable Interface Module supports eight channels of LoRa connectivity.

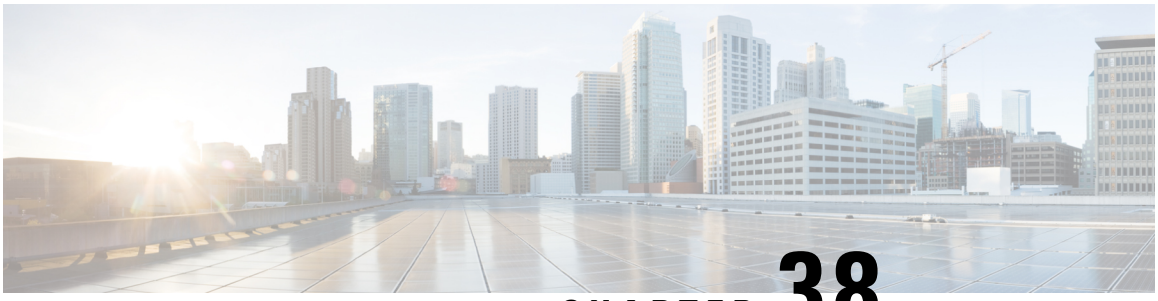
There are two different P-LPWA modules:

- The P-LPWA-900 is designed for RF regional profile US915, AS923 and AU915 as defined by the LoRa Alliance RF regional profile specifications.
- The P-LPWA-800 is designed for the EU868, IND865 and RU864 RF regional profile as defined by the LoRa Alliance RF regional profile specifications.

The Cisco LoRaWAN pluggable modules can be managed by command line interface (CLI), or the Cisco IOS XE Web User Interface (WebUI).

Details on installation, configuration, and regulatory information are found in the Cisco LoRaWAN Pluggable Interface Module Installation and Configuration Guide.

LoRaWAN Pluggable Interface Module Support



CHAPTER 38

Cisco SD-WAN Support

This chapter contains the following:

- Cisco SD-WAN Overview, on page 365
- vManage Support for the WP-WIFI6-x Module, on page 366
- SD-WAN, on page 367
- Related Documentation, on page 367
- vManage Support for EWC Mode on the Cisco Wi-Fi Interface Module, on page 368

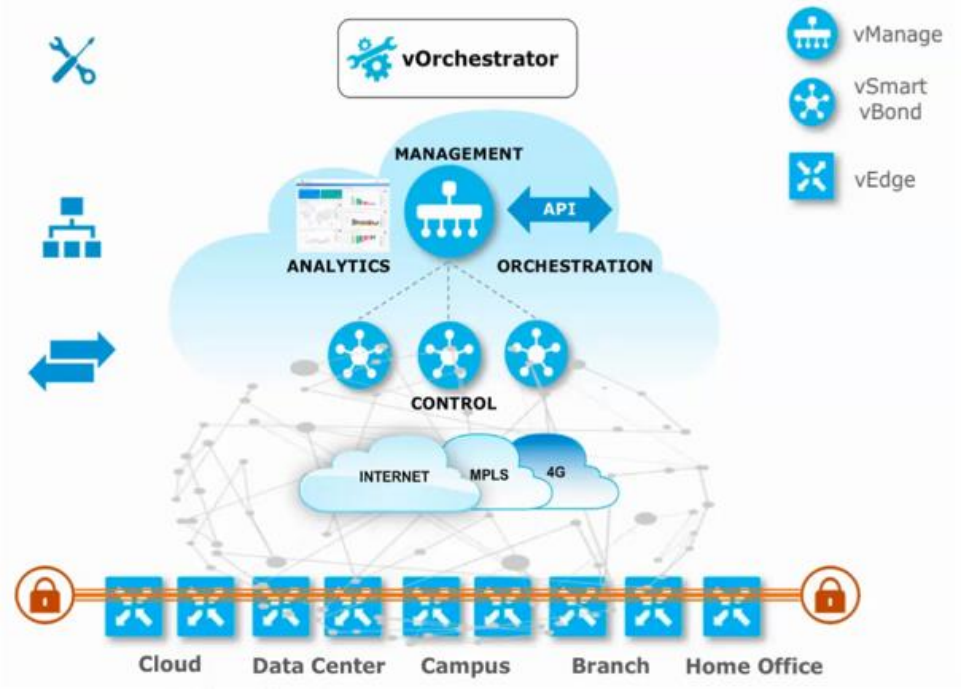
Cisco SD-WAN Overview

Cisco SD-WAN is a cloud-first architecture that separates data and control planes, managed through the Cisco vManage console. You can quickly establish an SD-WAN overlay fabric to connect data centers, branches, campuses, and co-location facilities to improve network speed, security, and efficiency.

Cisco SDWAN adopts a cloud based solution, it consists of vOrchestrator, vManage, vSmart and vEdge.

- vOrchestrator is responsible for launching all controllers VMs in the cloud.
- vManage is the management plane for the overall SDWAN solution. It uses netconf/YANG to talk to vEdge devices.
- vSmart is the control plane for the overall SDWAN solution. It talks to the vEdge device, acts as the route reflector, key reflector, and policy engine.
- vEdge is the data plane of the overall SDWAN solution. The IR1800 platform talks to vSmart, vManage, as part of the SDWAN network.

The follow diagram shows the high level architecture of SDWAN:



While Cisco SD-WAN is a cloud-first architecture, some of the components can be deployed on-premisis. Refer to the Cisco SD-WAN landing page for further information on the capabilities of SD-WAN.

Starting with IOS XE release 17.3.2, the IOS XE image can be configured as controller mode to run SD-WAN. A single universalk9 image is used to deploy Cisco IOS XE SD-WAN and Cisco IOS XE functionality. This universalk9 image supports two modes - Autonomous mode (for Cisco IOS XE features) and Controller mode (for Cisco SD-WAN features).

Access the Cisco IOS XE and Cisco IOS XE SD-WAN functionality through Autonomous and Controller execution modes, respectively. The Autonomous mode is the default mode for the router and includes the Cisco IOS XE functionality. To access Cisco IOS XE SD-WAN functionality, switch to the Controller mode. You can use the existing Plug and Play Workflow to determine the mode of the device. See the Cisco SD-WAN Getting Started Guide for further information.

Note The PnP process works on either Gi0/0/0 or Cellular.

vManage Support for the WP-WIFI6-x Module

This release will enable configuration and monitoring of the WP-WIFI6-x module through vManage from SDWAN. This applies only when the module is running in EWC mode.

Note

- vManage is now called Cisco Catalyst SD-WAN Manager.
- Cisco Catalyst SD-WAN Manager only supports SD-WAN UX 1.0, which is now replaced by SD-WAN UX 2.0.
- EWC mode is at End-of-Life and is not recommended for new deployments. For more information, see End-of-Sale and End-of-Life Announcement for the Cisco Wireless EWC on Access Point.

For further information about vManage, see the product landing page here: https://www.cisco.com/site/us/en/products/networking/wan/vmanage/index.html

SD-WAN

SD-WAN RA is a networking solution that

- combines IOS-XE SD-WAN and IOS-XE FlexVPN Remote Access Server features
- is supported on IoT routers with IOS XE 17.13.1, and
- provides remote access capabilities where all IoT devices only support the SD-WAN RA Client.

Additional documentation

Note All IoT devices only support the SD-WAN RA Client.

Information on SD-WAN Remote Access can be found in Cisco Catalyst SD-WAN Remote Access.

Additional documentation for SDWAN/vManage is available at these links:

- User Documentation for Cisco IOS XE Catalyst SD-WAN Release 17
- Cisco Catalyst SD-WAN
- Cisco SD-WAN Support Information
- Cisco vManage Monitor Overview
- Managing the SD-Routing Device Using Cisco SD-WAN Manager

Related Documentation

Cisco SDWAN documentation is available from the following sources:

https://www.cisco.com/c/en/us/support/routers/sd-wan/tsd-products-support-series-home.html

https://sdwan-docs.cisco.com/Product_Documentation/Software_Features

All of the technical documentation for Cisco SD-WAN can be found here:

https://www.cisco.com/c/en/us/support/routers/sd-wan/tsd-products-support-series-home.html

vManage Support for EWC Mode on the Cisco Wi-Fi Interface Module

The Cisco Wi-Fi Interface Module (WIM), is a pluggable interface available for all models of the IR1800 series. The PID is WP-WIFI6-x where x signifies the regulatory domain.

vManage support for EWC mode on the WIM module allows the user to configure the module in EWC mode with wlan profiles, radio profiles, and management details of the EWC from the router in SDWAN mode. The WIM is configured from vManage using feature template “ISR1K/IR18 Wireless” and verify the show wireless-lan commands in vManage.

With this release of IOS XE, vManage support has been added for the EWC Controller ONLY.

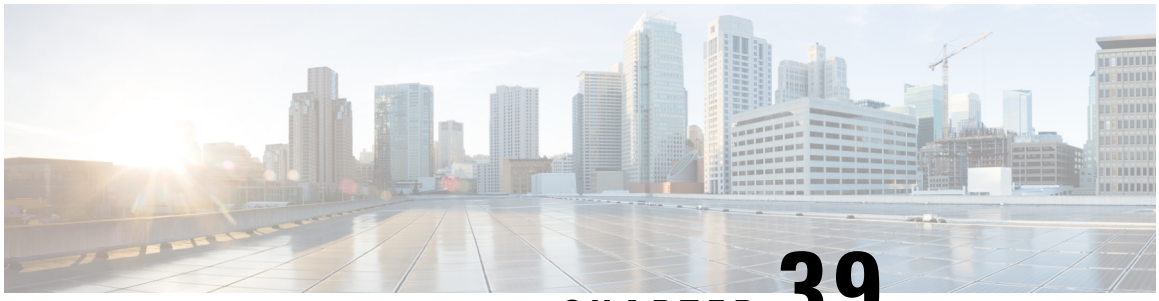
Note

- vManage is now called Cisco Catalyst SD-WAN Manager.
- Cisco Catalyst SD-WAN Manager only supports SD-WAN UX 1.0, which is now replaced by SD-WAN UX 2.0.
- EWC mode is at End-of-Life and is not recommended for new deployments. For more information, see End-of-Sale and End-of-Life Announcement for the Cisco Wireless EWC on Access Point.

Additional Documentation

Additional documentation for SDWAN/vManage is available at the following links:

- User Documentation for Cisco IOS XE Catalyst SD-WAN Release 17
- Cisco Catalyst SD-WAN
- Cisco SD-WAN Support Information
- Cisco vManage Monitor Overview
- Managing the SD-Routing Device Using Cisco SD-WAN Manager



CHAPTER 39

Troubleshooting

This section contains the following:

- Troubleshooting, on page 369
- Understanding Diagnostic Mode, on page 369
- Before Contacting Cisco or Your Reseller, on page 370
- show interfaces Troubleshooting Command, on page 370
- Software Upgrade Methods, on page 371
- Change the Configuration Register, on page 371
- Recovering a Lost Password, on page 374

Troubleshooting

This section describes the troubleshooting scenarios.

Before troubleshooting a software problem, you must connect a PC to the router via the console port. With a connected PC, you can view status messages from the router and enter commands to troubleshoot a problem.

You can also remotely access the interface by using Telnet. The Telnet option assumes that the interface is up and running.

Understanding Diagnostic Mode

The router boots up or accesses diagnostic mode in the following scenarios:

- The IOS process or processes fail, in some scenarios. In other scenarios, the system resets when the IOS process or processes fail.
- A user-configured access policy was configured using the **transport-map** command that directs the user into the diagnostic mode.
- A send break signal (**Ctrl-C** or **Ctrl-Shift-6**) was entered while accessing the router, and the router was configured to enter diagnostic mode when a break signal was sent.

In the diagnostic mode, a subset of the commands that are available in user EXEC mode are made available to the users. Among other things, these commands can be used to:

- Inspect various states on the router, including the IOS state.

- Replace or roll back the configuration.
- Provide methods of restarting the IOS or other processes.
- Reboot hardware, such as the entire router, a module, or possibly other hardware components.
- Transfer files into or off of the router using remote access methods such as FTP, TFTP, and SCP.

The diagnostic mode provides a more comprehensive user interface for troubleshooting than previous routers, which relied on limited access methods during failures, such as ROMMON, to diagnose and troubleshoot Cisco IOS problems. The diagnostic mode commands can work when the Cisco IOS process is not working properly. These commands are also available in privileged EXEC mode on the router when the router is working normally.

Before Contacting Cisco or Your Reseller

If you cannot locate the source of a problem, contact your local reseller for advice. Before you call, you should have the following information ready:

- Chassis type and serial number
- Maintenance agreement or warranty information
- Type of software and version number
- Date you received the hardware
- Brief description of the problem
- Brief description of the steps you have taken to isolate the problem

show interfaces Troubleshooting Command

Use the **show interfaces** command to display the status of all physical ports and logical interfaces on the router. Describe messages in the command output.

The IR1800 supports the following interfaces:

GigabitEthernet 0/0/0

GigabitEthernet 0/1/0 to 0/1/3

Cellular 0/4/0, 0/4/1, 0/5/0, and 0/5/1

Async 0/2/0 and 0/2/1

usbflash0:

msata

Alarm input alarm contact 0

Software Upgrade Methods

Several methods are available for upgrading software on the Cisco IR1800 Routers, including:

- Copy the new software image to flash memory over LAN or WAN when the existing Cisco IOS software image is in use.
- Copy the new software image to flash memory over the LAN while the boot image (ROM monitor) is operating.
- Copy the new software image over the console port while in ROM monitor mode.
- From ROM monitor mode, boot the router from a software image that is loaded on a TFTP server. To use this method, the TFTP server must be on the same LAN as the router.

Change the Configuration Register

To change a configuration register, follow these steps:

Procedure

Step 1 Connect a PC to the CONSOLE port on the router.

Step 2 At the privileged EXEC prompt (*router_name* #), enter the **show version** command to display the existing configuration register value (shown in bold at the bottom of this output example):

Example:

```
Router# show version
Cisco IOS XE Software, Version 17.06.01prd23
Cisco IOS Software [Bengaluru], ISR Software (ARMV8EL_LINUX_IOSD-UNIVERSALK9_IOT-M), Version
 17.6.1prd23, RELEASE SOFTWARE (fc1)
Technical Support: http://www.cisco.com/techsupport
Copyright (c) 1986-2021 by Cisco Systems, Inc.
Compiled Tue 20-Jul-21 02:28 by mcpre


Cisco IOS-XE software, Copyright (c) 2005-2021 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: 3.9(REL)

IR1833 uptime is 13 hours, 6 minutes
Uptime for this control processor is 13 hours, 9 minutes
System returned to ROM by Firmware Upgrade
System image file is "bootflash:ir1800-universalk9.17.06.01prd23.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.


Technology Package License Information:

------------------------------------------------------------------
Technology Type Technology-package Technology-package
Current Next Reboot
------------------------------------------------------------------
Smart License Perpetual network-advantage network-advantage
Smart License Subscription None None

The current throughput level is 50000 kbps


Smart Licensing Status: Registration Not Applicable/Not Applicable

cisco IR1833-K9 (1RU) processor with 470123K/6147K bytes of memory.
Processor board ID FCW2447P0EB
Router operating mode: Autonomous
MCU bootloader version: 0x22
MCU application version: 0x4d
1 Virtual Ethernet interface
6 Gigabit Ethernet interfaces
2 Serial interfaces
1 terminal line
4 Cellular interfaces
32768K bytes of non-volatile configuration memory.
3988088K bytes of physical memory.
7475198K bytes of Bootflash at bootflash:.

Configuration register is 0x2102

Router#
```

Step 3 Record the setting of the configuration register.

Step 4 To enable the break setting (indicated by the value of bit 8 in the configuration register), enter the **config-register** <*value*>command from privileged EXEC mode.

- Break enabled—Bit 8 is set to 0.
- Break disabled (default setting)—Bit 8 is set to 1.

Configuring the Configuration Register for Autoboot

Note Altering the configuration register is only for advanced troubleshooting and should only be done with guidance from Cisco support.

The configuration register can be used to change router behavior. This includes controlling how the router boots. Set the configuration register to 0x0 to boot into ROM, by using one of the following commands:

- In Cisco IOS configuration mode, use the **config-reg** 0x0 command.
- From the ROMMON prompt, use the **confreg** 0x0 command.

Note Setting the configuration register to 0x2102 will set the router to autoboot the Cisco IOS XE software.

Reset the Router

To reset the router, follow these steps:

Procedure

Step 1 If break is disabled, turn the router off (O), wait 5 seconds, and turn it on (|) again. Within 60 seconds, press the **Break** key. The terminal displays the ROM monitor prompt.

Note
Some terminal keyboards have a key labeled *Break* . If your keyboard does not have a Break key, see the documentation that came with the terminal for instructions on how to send a break.

Step 2 Press break. The terminal displays the following prompt:

Example:

```
rommon 2>
```

Step 3 Enter **confreg 0x2142** to reset the configuration register:

Example:

```
rommon 2> confreg 0x142
```

Step 4 Sync the configuration changes with **sync** command.

Example:

```
rommon 2>sync
```

Step 5 Initialize the router by entering the **reload** command:

Example:

```
rommon 2>reload
```

The router cycles its power, and the configuration register is set to 0x2142. The router uses the boot ROM system image, indicated by the system configuration dialog:

Example:

```
--- System Configuration Dialog ---
```

Step 6 Enter **no** in response to the prompts until the following message is displayed:

Example:

```
Press RETURN to get started!
```

Step 7 Press **Return**. The following prompt appears:

Example:

```
Router>
```

Step 8 Enter the enable command to enter enable mode. Configuration changes can be made only in enable mode:

Example:

```
Router> enable
```

The prompt changes to the privileged EXEC prompt:

Example:

```
Router#
```

Step 9 Enter the **show startup-config** command to display an enable password in the configuration file:

Example:

```
Router# show startup-config
```

What to do next

If you are recovering an enable password, do not perform the steps in the Reset the Password and Save Your Changes section. Instead, complete the password recovery process by performing the steps in the Reset the Configuration Register Value section.

If you are recovering an enable secret password, it is not displayed in the **show startup-config** command output. Complete the password recovery process by performing the steps in the Reset the Password and Save Your Changes section.

Recovering a Lost Password

To recover a lost enable or lost enable-secret password, refer to the following sections:

1. Change the Configuration Register
2. Reset the Router
3. Reset the Password and Save your Changes (for lost enable secret passwords only)
4. Reset the Configuration Register Value.

5. If you have performed a **write erase**, or used the reset button, you will need to add the license.

```
IR1800#config term
IR1800#license smart reservation
```

Note Any vlan interfaces will have to be recreated, and it is possible you may need to re-generate certs.

Note Recovering a lost password is only possible when you are connected to the router through the console port. These procedures cannot be performed through a Telnet session.

Tip See the “Hot Tips” section on Cisco.com for additional information on replacing enable secret passwords.

Reset the Password and Save Your Changes

To reset your password and save the changes, follow these steps:

Procedure

Step 1 Enter the **configure terminal** command to enter global configuration mode:

Example:

```
Router# configure terminal
```

Step 2 Enter the **enable secret** command to reset the enable secret password in the router:

Example:

```
Router(config)# enable secret
password
```

Step 3 Enter **exit** to exit global configuration mode:

Example:

```
Router(config)# exit
```

Step 4 Save your configuration changes:

Example:

```
Router# copy running-config startup-config
```

Reset the Configuration Register Value

To reset the configuration register value after you have recovered or reconfigured a password, follow these steps:

Procedure

Step 1 Enter the **configure terminal** command to enter global configuration mode:

Example:

```
Router# configure terminal
```

Step 2 Enter the **configure register** command and the original configuration register value that you recorded.

Example:

```
Router(config)# config-reg
value
```

Step 3 Enter **exit** to exit configuration mode:

Example:

```
Router(config)# exit
```

Note
To return to the configuration being used before you recovered the lost enable password, do not save the configuration changes before rebooting the router.

Step 4 Reboot the router, and enter the recovered password.

Configuring a Console Port Transport Map

This task describes how to configure a transport map for a console port interface on the router.

Procedure

	Command or Action	Purpose
Step 1	**enable**	Enables privileged EXEC mode.
	Example:	Enter your password if prompted.

	Command or Action	Purpose
	`Router> enable`	
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**transport-map type console** *transport-map-name* **Example:** `Router(config)# transport-map type console consolehandler`	Creates and names a transport map for handling console connections, and enters transport map configuration mode.
Step 4	**connection wait** [**allow** [**interruptible**] | **none** [**disconnect**]] **Example:** `Router(config-tmap)# connection wait none`	Specifies how a console connection will be handled using this transport map. • **allow interruptible**—The console connection waits for a Cisco IOS VTY line to become available, and also allows users to enter diagnostic mode by interrupting a console connection that is waiting for a Cisco IOS VTY line to become available. This is the default setting. **Note** Users can interrupt a waiting connection by entering **Ctrl-C** or **Ctrl-Shift-6**. • **none**—The console connection immediately enters diagnostic mode.

Step 5	(Optional) **banner** [**diagnostic** | **wait**] *banner-message*
	Example:
	`Router(config-tmap)# banner diagnostic X`
	`Enter TEXT message. End with the character 'X'.`
	`--Welcome to Diagnostic Mode--`
	`X`
	`Router(config-tmap)#`
		(Optional) Creates a banner message that will be seen by users entering diagnostic mode or waiting for the Cisco IOS VTY line because of the console transport map configuration.
		• **diagnostic**—Creates a banner message seen by users directed to diagnostic mode because of the console transport map configuration.
		Note Users can interrupt a waiting connection by entering **Ctrl-C** or **Ctrl-Shift-6**.
		• **wait**—Creates a banner message seen by users waiting for Cisco IOS VTY to become available.

	Command or Action	Purpose
		• *banner-message*—Banner message, which begins and ends with the same delimiting character.
Step 6	**exit** **Example:** `Router(config-tmap)# exit`	Exits transport map configuration mode to re-enter global configuration mode.
Step 7	**transport type console** *console-line-number* **input** *transport-map-name* **Example:** `Router(config)# transport type console 0 input consolehandler`	Applies the settings defined in the transport map to the console interface. The *transport-map-name* for this command must match the *transport-map-name* defined in the **transport-map type console** command.

Examples

The following example shows how to create a transport map to set console port access policies and attach to console port 0:

```
Router(config)# transport-map type console consolehandler
Router(config-tmap)# connection wait allow interruptible
Router(config-tmap)# banner diagnostic X
Enter TEXT message. End with the character 'X'.
--Welcome to diagnostic mode--
X
Router(config-tmap)# banner wait X
Enter TEXT message. End with the character 'X'.
Waiting for IOS vty line
X
Router(config-tmap)# exit
Router(config)# transport type console 0 input consolehandler
```

Viewing Console Port, SSH, and Telnet Handling Configurations

Use the following commands to view console port, SSH, and Telnet handling configurations:

- **show transport-map**
- **show platform software configuration access policy**

Use the **show transport-map** command to view transport map configurations.

show transport-map [**all** | **name** *transport-map-name* | **type** [**console**]]

This command can be used either in user EXEC mode or privileged EXEC mode.

Example

The following example shows transport maps that are configured on the router: console port (`consolehandler`):

```
Router# show transport-map all
Transport Map:
Name: consolehandler Type: Console Transport

Connection:
Wait option: Wait Allow Interruptable Wait banner:

Waiting for the IOS CLI bshell banner:
Welcome to Diagnostic Mode



Router# show transport-map type console
Transport Map:
Name: consolehandler


REVIEW DRAFT - CISCO CONFIDENTIAL

Type: Console Transport

Connection:
Wait option: Wait Allow Interruptable Wait banner:

Waiting for the IOS CLI Bshell banner:
Welcome to Diagnostic Mode



Router# show transport-map type persistent ssh
Transport Map:
Name: consolehandler Type: Console Transport

Connection:
Wait option: Wait Allow Interruptable Wait banner:

Waiting for the IOS CLI Bshell banner:
Welcome to Diagnostic Mode
```

Use the **show platform software configuration access policy** command to view the current configurations for handling the incoming console port, SSH, and Telnet connections. The output of this command provides the current wait policy for each type of connection (Telnet, SSH, and console), as well as information on the currently configured banners.

Unlike the **show transport-map** command, the **show platform software configuration access policy** command is available in diagnostic mode so that it can be entered in scenarios where you need transport map configuration information, but cannot access the Cisco IOS CLI.

Example

The following example shows the **show platform software configuration access policy** command.

```
Router# show platform software configuration access policy
The current access-policies

Method : telnet
```

```
Rule : wait with interrupt Shell banner:
Welcome to Diagnostic Mode

Wait banner :
Waiting for IOS Process

Method : ssh Rule : wait Shell banner: Wait banner :

Method : console
Rule : wait with interrupt Shell banner:
Wait banner :
```

Using the factory reset Commands

The **factory reset** commands are used to remove all the customer specific data on a router/switch that has been added. The data can be configuration, log files, boot variables, core files, and so on.

The **factory-reset all** command erases the bootflash, nvram, rommon variables, licenses, and logs.

Caution Use of the factory reset command should not be done lightly. All customer configurations will be ddeleted and the platform will boot up as if new from the factory.

Note factory-reset all does not work if IOS-XE is running in controller mode. Please refer to SDWAN configuration information.

```
Router#factory-reset all
The factory reset operation is irreversible for all operations. Are you sure? [confirm]
*Enter*

*May 12 09:55:45.831: %SYS-5-RELOAD: Reload requested by Exec. Reload Reason: Factory Reset.

***Return to ROMMON Prompt
```

Boot Sequence after Factory Reset

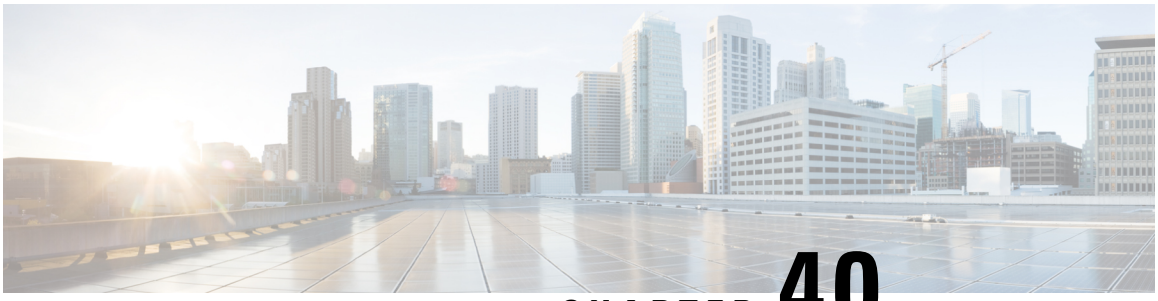
Booting the image:

- The bootloader attempts to boot “golden.bin” from the bootflash: partition
- If no “golden.bin” is present, then boot the first image.

Loading the configuration:

- IOS looks for “golden.cfg” file on nvram: partition and applies it upon booting.
- If no “golden.cfg” is present on nvram: then IOS looks for “golden.cfg” file on bootflash: partition and applies it upon booting.

- If no “golden.cfg” is present on bootflash: then configurations are erased and Software Configuration dialog is used.



CHAPTER 40

System Messages

This chapter contains the following sections:

- Information About Process Management, on page 383
- How to Find Error Message Details, on page 383

Information About Process Management

You can access system messages by logging in to the console through Telnet protocol and monitoring your system components remotely from any workstation that supports the Telnet protocol.

Starting and monitoring software is referred to as process management. The process management infrastructure for a router is platform independent, and error messages are consistent across platforms running on Cisco IOS XE. You do not have to be directly involved in process management, but we recommend that you read the system messages that refer to process failures and other issues.

How to Find Error Message Details

To show further details about a process management or a syslog error message, enter the error message into the Error Message Decoder tool at: https://www.cisco.com/cgi-bin/Support/Errordecoder/index.cgi.

For example, enter the message `%PMAN-0-PROCESS_NOTIFICATION` into the tool to view an explanation of the error message and the recommended action to be taken.

The following are examples of the description and the recommended action displayed by the Error Message Decoder tool for some of the error messages.

Error Message: `%PMAN-0-PROCESS_NOTIFICATION : The process lifecycle notification component failed because [chars]`

Explanation	Recommended Action

The process lifecycle notification component failed, preventing proper detection of a process start and stop. This problem is likely the result of a software defect in the software subpackage.	Note the time of the message and investigate the kernel error message logs to learn more about the problem and see if it is correctable. If the problem cannot be corrected or the logs are not helpful, copy the error message exactly as it appears on the console along with the output of the **show tech-support** command and provide the gathered information to a Cisco technical support representative.

Error Message: `%PMAN-0-PROCFAILCRIT A critical process [chars] has failed (rc [dec])`

Explanation	Recommended Action
A process important to the functioning of the router has failed.	Note the time of the message and investigate the error message logs to learn more about the problem. If the problem persists, copy the message exactly as it appears on the console or in the system log. Research and attempt to resolve the issue using the tools and utilities provided at: http://www.cisco.com/tac. With some messages, these tools and utilities will supply clarifying information. Search for resolved software issues using the Bug Search Tool at: https://bst.cloudapps.cisco.com/bugsearch. If you still require assistance, open a case with the Technical Assistance Center at: http://tools.cisco.com/ServiceRequestTool/create/, or contact your Cisco technical support representative and provide the representative with the information you have gathered. Attach the following information to your case in nonzipped, plain-text (.txt) format: the output of the **show logging** and **show tech-support** commands and your pertinent troubleshooting logs.

Error Message: `%PMAN-3-PROCFAILOPT An optional process [chars] has failed (rc [dec])`

Explanation	Recommended Action

A process that does not affect the forwarding of traffic has failed.	Note the time of the message and investigate the kernel error message logs to learn more about the problem. Although traffic will still be forwarded after receiving this message, certain functions on the router may be disabled because of this message and the error should be investigated. If the logs are not helpful or indicate a problem you cannot correct, copy the message exactly as it appears on the console or in the system log. Research and attempt to resolve the issue using the tools and utilities provided at http://www.cisco.com/tac. With some messages, these tools and utilities will supply clarifying information. Search for resolved software issues using the Bug Search Tool at: https://bst.cloudapps.cisco.com/bugsearch. If you still require assistance, open a case with the Technical Assistance Center at: http://tools.cisco.com/ServiceRequestTool/create/, or contact your Cisco technical support representative and provide the representative with the information you have gathered. Attach the following information to your case in nonzipped, plain-text (.txt) format: the output of the **show logging** and **show tech-support** commands and your pertinent troubleshooting logs.

Error Message: `%PMAN-3-PROCFAIL The process [chars] has failed (rc [dec])`

Explanation	Recommended Action
The process has failed as the result of an error.	This message will appear with other messages related to the process. Check the other messages to determine the reason for the failures and see if corrective action can be taken. If the problem persists, copy the message exactly as it appears on the console or in the system log. Research and attempt to resolve the issue using the tools and utilities provided at: http://www.cisco.com/tac. With some messages, these tools and utilities will supply clarifying information. Search for resolved software issues using the Bug Search Tool at: https://bst.cloudapps.cisco.com/bugsearch. If you still require assistance, open a case with the Technical Assistance Center at: http://tools.cisco.com/ServiceRequestTool/create/, or contact your Cisco technical support representative and provide the representative with the information you have gathered. Attach the following information to your case in nonzipped, plain-text (.txt) format: the output of the **show logging** and **show tech-support** commands and your pertinent troubleshooting logs.

Error Message: `%PMAN-3-PROCFAIL_IGNORE [chars] process exits and failures are being ignored due to debug settings. Normal router functionality will be affected. Critical router functions like RP switchover, router reload, FRU resets, etc. may not function properly.`

Explanation	Recommended Action
A process failure is being ignored due to the user-configured debug settings.	If this behavior is desired and the debug settings are set according to a user's preference, no action is needed. If the appearance of this message is viewed as a problem, change the debug settings. The router is not expected to behave normally with this debug setting. Functionalities such as SSO switchover, router reloads, FRU resets, and so on will be affected. This setting should only be used in a debug scenario. It is not normal to run the router with this setting.

Error Message: `%PMAN-3-PROCHOLDDOWN The process [chars] has been helddown (rc [dec])`

Explanation	Recommended Action
The process was restarted too many times with repeated failures and has been placed in the hold-down state.	This message will appear with other messages related to the process. Check the other messages to determine the reason for the failures and see if corrective action can be taken. If the problem persists, copy the message exactly as it appears on the console or in the system log. Research and attempt to resolve the issue using the tools and utilities provided at: http://www.cisco.com/tac. With some messages, these tools and utilities will supply clarifying information. Search for resolved software issues using the Bug Search Tool at: https://bst.cloudapps.cisco.com/bugsearch. If you still require assistance, open a case with the Technical Assistance Center at: http://tools.cisco.com/ServiceRequestTool/create/, or contact your Cisco technical support representative and provide the representative with the information you have gathered. Attach the following information to your case in nonzipped, plain-text (.txt) format: the output of the **show logging** and **show tech-support** commands and your pertinent troubleshooting logs.

Error Message: `%PMAN-3-RELOAD_RP_SB_NOT_READY : Reloading: [chars]`

Explanation	Recommended Action
The route processor is being reloaded because there is no ready standby instance.	Ensure that the reload is not due to an error condition.

Error Message: `%PMAN-3-RELOAD_RP : Reloading: [chars]`

Explanation	Recommended Action

The RP is being reloaded.	Ensure that the reload is not due to an error condition. If it is due to an error condition, collect information requested by the other log messages.

Error Message: `%PMAN-3-RELOAD_SYSTEM : Reloading: [chars]`

Explanation	Recommended Action
The system is being reloaded.	Ensure that the reload is not due to an error condition. If it is due to an error condition, collect information requested by the other log messages.

Error Message: `%PMAN-3-PROC_BAD_EXECUTABLE : Bad executable or permission problem with process [chars]`

Explanation	Recommended Action
The executable file used for the process is bad or has permission problem.	Ensure that the named executable is replaced with the correct executable.

Error Message: `%PMAN-3-PROC_BAD_COMMAND:Non-existent executable or bad library used for process <process name>`

Explanation	Recommended Action
The executable file used for the process is missing, or a dependent library is bad.	Ensure that the named executable is present and the dependent libraries are good.

Error Message: `%PMAN-3-PROC_EMPTY_EXEC_FILE : Empty executable used for process [chars]`

Explanation	Recommended Action
The executable file used for the process is empty.	Ensure that the named executable is non-zero in size.

Error Message: `%PMAN-5-EXITACTION : Process manager is exiting: [chars]`

Explanation	Recommended Action
The process manager is exiting.	Ensure that the process manager is not exiting due to an error condition. If it is due to an error condition, collect information requested by the other log messages.

Error Message: `%PMAN-6-PROCSHUT : The process [chars] has shutdown`

Explanation	Recommended Action
The process has gracefully shut down.	No user action is necessary. This message is provided for informational purposes only.

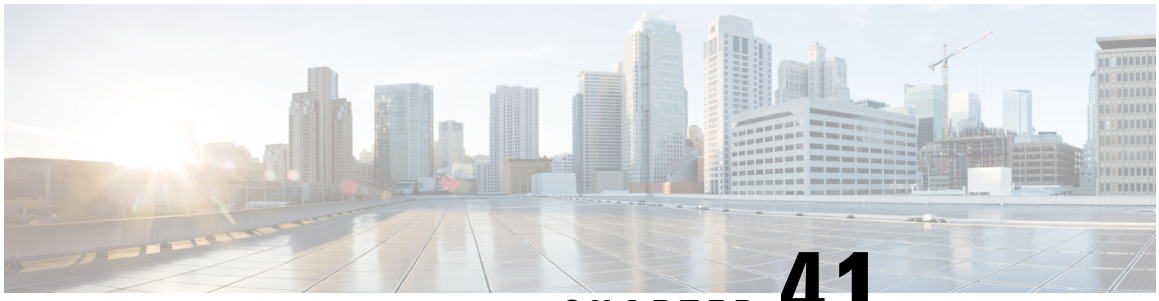
Error Message: `%PMAN-6-PROCSTART : The process [chars] has started`

Explanation	Recommended Action

The process has launched and is operating properly.	No user action is necessary. This message is provided for informational purposes only.

Error Message: `%PMAN-6-PROCSTATELESS : The process [chars] is restarting stateless`

Explanation	Recommended Action
The process has requested a stateless restart.	No user action is necessary. This message is provided for informational purposes only.



CHAPTER 41

Environmental Monitoring

This chapter contains the following sections:

- Environmental Monitoring and Reporting Functions, on page 389
- Environmental Monitoring Functions, on page 389
- Environmental Reporting Functions, on page 391
- SNMP Polling of Temperature OID, on page 398
- Additional References, on page 398
- Technical Assistance, on page 399

Environmental Monitoring and Reporting Functions

Monitoring and reporting functions allow you to maintain normal system operation by identifying and resolving adverse conditions prior to loss of operation.

- Environmental Monitoring Functions, on page 389
- Environmental Reporting Functions, on page 391

Environmental Monitoring Functions

The router provides a robust environment-monitoring system with several sensors that monitor the system temperatures. The following are some of the key functions of the environmental monitoring system:

- Monitoring temperature of CPUs and Motherboard
- Recording abnormal events and generating notifications
- Monitoring Simple Network Management Protocol (SNMP) traps
- Generating and collecting Onboard Failure Logging (OBFL) data
- Sending call home event notifications
- Logging system error messages
- Displaying present settings and status

Environmental monitoring functions use sensors to monitor the temperature of the cooling air as it moves through the chassis.

The router is expected to meet the following environmental operating conditions

- Non-operating Temperature: -40°F to 158°F (-40°C to 70°C)
- Non-operating Humidity: 5 to 95% relative humidity (non-condensing)
- Operating Temperature:

  -40° to 140°F (-40° to 60°C) in a sealed NEMA cabinet with no airflow

  -40° to 158°F (-40° to 70°C) in a vented cabinet with 40 lfm of air

  -40° to 167°F (-40° to 75°C) in a forced air enclosure with 200 lfm of air
- Operating Humidity: 10% to 95% relative humidity (non-condensing)
- Operating Altitude: -500 to 5,000 feet. Derate max operating temperature 1.5°C per 1000 feet.

The following table displays the levels of status conditions used by the environmental monitoring system.

Table 26: Levels of Status Conditions Used by the Environmental Monitoring System

Status Level	Description
Normal	All monitored parameters are within normal tolerance.
Warning	The system has exceeded a specified threshold. The system continues to operate, but operator action is recommended to bring the system back to a normal state.
Critical	An out-of-tolerance temperature or voltage condition exists. Although the system continues to operate, it is approaching shutdown. Immediate operator action is required.

The environmental monitoring system sends system messages to the console, for example, when the conditions described here are met:

Temperature and Voltage Exceed Max/Min Thresholds

The following example shows the warning messages indicating the maximum and minimum thresholds of the temperature or voltage:

```
Warnings :
--------
For all the temperature sensors (name starting with “Temp:”) above,
the critical warning threshold is 100C (100C and higher)
the warning threshold is 80C (range from 80C to 99C)
the low warning threshold is 1C (range from -inf to 1C).

For all voltage sensors (names starting with "V:"),
the high warning threshold starts at that voltage +10%. (voltage + 10% is warning)
the low warning threshold starts at the voltage -10%. (voltage - 10% is warning)
```

Environmental Reporting Functions

You can retrieve and display environmental status reports using the following commands:

- **show diag all eeprom**
- **show environment**
- **show environment all**
- **show inventory**
- **show platform**
- **show platform diag**
- **show platform software status control-processor**
- **show diag slot R0 eeprom detail**
- **show version**
- **show power**

These commands show the current values of parameters such as temperature and voltage.

The environmental monitoring system updates the values of these parameters every 60 seconds. Brief examples of these commands are shown below:

show diag all eeprom

```
Router# show diag all eeprom
MIDPLANE EEPROM data:

Product Identifier (PID) : IR1800-K9
Version Identifier (VID) : V00
PCB Serial Number : FOC21482ZQF
PCB Serial Number : FOC214822CK
PCB Serial Number : FOC21482SY7
Top Assy. Part Number : 68-6479-01
Top Assy. Revision : 13
Hardware Revision : 0.2
Asset ID :
CLEI Code : UNASSIGNED
Power/Fan Module P0 EEPROM data is not initialized

Power/Fan Module P1 EEPROM data is not initialized

Slot R0 EEPROM data:

Product Identifier (PID) : IR1800-K9
Version Identifier (VID) : V00
PCB Serial Number : FOC21482ZQF
PCB Serial Number : FOC214822CK
PCB Serial Number : FOC21482SY7
Top Assy. Part Number : 68-6479-01
Top Assy. Revision : 13
Hardware Revision : 0.2
CLEI Code : UNASSIGNED
```

```
Slot F0 EEPROM data:

Product Identifier (PID) : IR1800-K9
Version Identifier (VID) : V00
PCB Serial Number : FOC21482ZQF
PCB Serial Number : FOC214822CK
PCB Serial Number : FOC21482SY7
Top Assy. Part Number : 68-6479-01
Top Assy. Revision : 13
Hardware Revision : 0.2
CLEI Code : UNASSIGNED
Slot 0 EEPROM data:

Product Identifier (PID) : IR1800-K9
Version Identifier (VID) : V00
PCB Serial Number : FOC21482ZQF
PCB Serial Number : FOC214822CK
PCB Serial Number : FOC21482SY7
Top Assy. Part Number : 68-6479-01
Top Assy. Revision : 13
Hardware Revision : 0.2
CLEI Code : UNASSIGNED
SPA EEPROM data for subslot 0/0:

Product Identifier (PID) : IR1800-ES-5
Version Identifier (VID) : V01
PCB Serial Number :
Top Assy. Part Number : 68-2236-01
Top Assy. Revision : A0
Hardware Revision : 2.2
CLEI Code : CNUIAHSAAA
SPA EEPROM data for subslot 0/1 is not available

SPA EEPROM data for subslot 0/2 is not available

SPA EEPROM data for subslot 0/3 is not available

SPA EEPROM data for subslot 0/4 is not available

SPA EEPROM data for subslot 0/5 is not available

Router#
```

show environment:

```
Router# show environment
Number of Critical alarms: 0
Number of Major alarms: 0
Number of Minor alarms: 0

Slot Sensor Current State Reading Threshold(Minor,Major,Critical,Shutdown)
---------- ------------- --------------- ------------ ---------------------------------------
R0 Temp: LM75BXXX Normal 43 Celsius (75 ,80 ,90 ,na )(Celsius)

Router#
```

show environment all:

```
Router# show environment all
Sensor List: Environmental Monitoring
Sensor Location State Reading
Temp: LM75BXXX R0 Normal 48 Celsius
```

show inventory:

```
Router# show inventory
++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
INFO: Please use "show license UDI" to get serial number for licensing.
++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++

NAME: "Chassis", DESCR: "Cisco Catalyst IR1833 Rugged Series Router"
PID: IR1833-K9 , VID: V00 , SN: FCW2447P0EB

NAME: "Power Supply Module 0", DESCR: "Cisco IR1800 DC Power Supply"
PID: PWR-12V , VID: , SN:

NAME: "GE-POE Module", DESCR: "POE Module for On Board GE for Cisco IR183X"
PID: IR1800-I-POE , VID: V00 , SN: FOC24382K4W

NAME: "module 0", DESCR: "Cisco IR-1833-K9 Built-In NIM controller"
PID: IR1833-K9 , VID: , SN:

NAME: "NIM subslot 0/3", DESCR: "Cisco Wide Pluggable Form Factor WIFI6 AP Module"
PID: WP-WIFI6-B , VID: V00 , SN: FOC24490FEP

NAME: "NIM subslot 0/4", DESCR: "P-LTEA-LA Module"
PID: P-LTEA-LA , VID: V01 , SN: FOC22287JMC

NAME: "Modem on Cellular0/4/0", DESCR: "Sierra Wireless EM7430"
PID: EM7430 , VID: 1.0 , SN: 355813070165276

NAME: "NIM subslot 0/5", DESCR: "P-LTEA-LA Module"
PID: P-LTEA-LA , VID: V01 , SN: FOC22287JLZ

NAME: "Modem on Cellular0/5/0", DESCR: "Sierra Wireless EM7430"
PID: EM7430 , VID: 1.0 , SN: 355813070165524

NAME: "NIM subslot 0/0", DESCR: "Front Panel 1 port Gigabitethernet Module"
PID: IR1833-1x1GE , VID: V01 , SN:

NAME: "NIM subslot 0/1", DESCR: "IR1833-ES-4"
PID: IR1833-ES-4 , VID: V01 , SN:

NAME: "module R0", DESCR: "Cisco IR1833-K9 motherboard"
PID: IR1833-K9 , VID: V00 , SN: FOC24384177

NAME: "module F0", DESCR: "Cisco IR1833-K9 Forwarding Processor"
PID: IR1833-K9 , VID: , SN:
```

show platform:

```
Router# show platform
Chassis type: IR1833-K9

Slot Type State Insert time (ago)
--------- ------------------- --------------------- -----------------
0 IR1833-K9 ok 00:04:03
0/0 IR1833-1x1GE ok 00:01:22
0/1 IR1833-ES-4 ok 00:01:22
0/3 WP-WIFI6-B ok 00:01:22
0/4 P-LTEA-LA ok 00:01:21
0/5 P-LTEA-LA ok 00:01:21
R0 IR1833-K9 ok, active 00:04:03
F0 IR1833-K9 ok, active 00:04:03
P0 PWR-12V ok 00:02:00
GE-POE IR1800-I-POE ok 00:02:00
```

show platform diag:

```
Router# show platform diag
Chassis type: IR1833-K9

Slot: 0, IR1833-K9
Running state : ok
Internal state : online
Internal operational state : ok
Physical insert detect time : 00:01:09 (00:04:38 ago)
Software declared up time : 00:03:19 (00:02:28 ago)
CPLD version :
Firmware version : 3.9(REL)

Sub-slot: 0/0, IR1833-1x1GE
Operational status : ok
Internal state : inserted
Physical insert detect time : 00:03:51 (00:01:57 ago)
Logical insert detect time : 00:03:51 (00:01:57 ago)

Sub-slot: 0/1, IR1833-ES-4
Operational status : ok
Internal state : inserted
Physical insert detect time : 00:03:51 (00:01:57 ago)
Logical insert detect time : 00:03:51 (00:01:57 ago)

Sub-slot: 0/3, WP-WIFI6
Operational status : ok
Internal state : inserted
Physical insert detect time : 00:03:51 (00:01:57 ago)
Logical insert detect time : 00:03:51 (00:01:57 ago)

Sub-slot: 0/4, P-LTEA-LA
Operational status : ok
Internal state : inserted
Physical insert detect time : 00:03:51 (00:01:56 ago)
Logical insert detect time : 00:03:51 (00:01:56 ago)

Sub-slot: 0/5, P-LTEA-LA
Operational status : ok
Internal state : inserted
```

```
Physical insert detect time : 00:03:51 (00:01:56 ago)
Logical insert detect time : 00:03:51 (00:01:56 ago)

Slot: R0, IR1833-K9
Running state : ok, active
Internal state : online
Internal operational state : ok
Physical insert detect time : 00:01:09 (00:04:38 ago)
Software declared up time : 00:01:09 (00:04:38 ago)
CPLD version : 00000000
Firmware version : 3.9(REL)

Slot: F0, IR1833-K9
Running state : ok, active
Internal state : online
Internal operational state : ok
Physical insert detect time : 00:01:09 (00:04:38 ago)
Software declared up time : 00:03:30 (00:02:17 ago)
Hardware ready signal time : 00:00:00 (never ago)
Packet ready signal time : 00:03:36 (00:02:11 ago)
CPLD version : 00000000
Firmware version : 3.9(REL)

Slot: P0, PWR-12V
State : ok
Physical insert detect time : 00:03:13 (00:02:35 ago)

Slot: GE-POE, IR1800-I-POE
State : ok
Physical insert detect time : 00:03:13 (00:02:35 ago)
```

show platform software status control-processor:

```
Router# show platform software status control-processor
RP0: online, statistics updated 9 seconds ago
Load Average: healthy
1-Min: 0.32, status: healthy, under 5.00
5-Min: 0.33, status: healthy, under 5.00
15-Min: 0.35, status: healthy, under 5.00
Memory (kb): healthy
Total: 3959840
Used: 2894588 (73%), status: healthy
Free: 1065252 (27%)
Committed: 2435656 (62%), under 90%
Per-core Statistics
CPU0: CPU Utilization (percentage of time spent)
User: 0.50, System: 0.91, Nice: 0.00, Idle: 98.07
IRQ: 0.40, SIRQ: 0.10, IOwait: 0.00
CPU1: CPU Utilization (percentage of time spent)
User: 0.81, System: 0.30, Nice: 0.00, Idle: 98.48
IRQ: 0.20, SIRQ: 0.20, IOwait: 0.00
CPU2: CPU Utilization (percentage of time spent)
User: 0.81, System: 2.65, Nice: 0.00, Idle: 95.41
IRQ: 1.12, SIRQ: 0.00, IOwait: 0.00
CPU3: CPU Utilization (percentage of time spent)
User: 7.66, System: 17.05, Nice: 0.00, Idle: 70.58
IRQ: 4.59, SIRQ: 0.10, IOwait: 0.00
Router#
```

show diag slot R0 eeprom detail:

```
Router# show diag slot R0 eeprom detail
Slot R0 EEPROM data:
EEPROM version : 4
Compatible Type : 0xFF
Controller Type : 3457
Hardware Revision : 0.2
PCB Part Number : 73-18820-03
Board Revision : 02
Deviation Number : 0
Fab Version : 02
PCB Serial Number : FOC22106KKH
Top Assy. Part Number : 68-6479-03
Top Assy. Revision : 04
Chassis Serial Number : FCW2213TH07
Deviation Number : 0
RMA Test History : 00
RMA Number : 0-0-0-0
RMA History : 00
Product Identifier (PID) : IR1800-K9
Version Identifier (VID) : V00
CLEI Code : UNASSIGNED
Manufacturing Test Data : 00 00 00 00 00 00 00 00
Field Diagnostics Data : 00 00 00 00 00 00 00 00
Chassis MAC Address : 682c.7b4d.7880
MAC Address block size : 128
Asset ID :
Asset Alias :
PCB Part Number : 73-18821-03
Board Revision : 03
Deviation Number : 0
Fab Version : 02
PCB Serial Number : FOC22106KHD
PCB Part Number : 73-19117-02
Board Revision : 02
Deviation Number : 0
Fab Version : 01
PCB Serial Number : FOC22106KJ9
Asset ID :
Router#
```

show version:

```
Router# show version
Cisco IOS XE Software, Version 17.06.01prd23
Cisco IOS Software [Bengaluru], ISR Software (ARMV8EL_LINUX_IOSD-UNIVERSALK9_IOT-M), Version
 17.6.1prd23, RELEASE SOFTWARE (fc1)
Technical Support: http://www.cisco.com/techsupport
Copyright (c) 1986-2021 by Cisco Systems, Inc.
Compiled Tue 20-Jul-21 02:28 by mcpre

Cisco IOS-XE software, Copyright (c) 2005-2021 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: 3.9(REL)

IR1833 uptime is 13 hours, 6 minutes
Uptime for this control processor is 13 hours, 9 minutes
System returned to ROM by Firmware Upgrade
System image file is "bootflash:ir1800-universalk9.17.06.01prd23.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.

Technology Package License Information:
------------------------------------------------------------------
Technology Type Technology-package Technology-package
Current Next Reboot
------------------------------------------------------------------
Smart License Perpetual network-advantage network-advantage
Smart License Subscription None None

The current throughput level is 50000 kbps
Smart Licensing Status: Registration Not Applicable/Not Applicable
cisco IR1833-K9 (1RU) processor with 470123K/6147K bytes of memory.
Processor board ID FCW2447P0EB
Router operating mode: Autonomous
MCU bootloader version: 0x22
MCU application version: 0x4d
1 Virtual Ethernet interface
6 Gigabit Ethernet interfaces
2 Serial interfaces
1 terminal line
4 Cellular interfaces
32768K bytes of non-volatile configuration memory.
3988088K bytes of physical memory.
7475198K bytes of Bootflash at bootflash:.
Configuration register is 0x2102
```

show power:

```
Router# show power
Main PSU :
Total Power Consumed: 8.16 Watts
Router#
```

SNMP Polling of Temperature OID

Support has been added for SNMP MIB to be able to return values from temperature sensors. The output should look similar to the **show environment** CLI.

The output of a **show environment**:

```
router#show environment

Number of Critical alarms:  0
Number of Major alarms:     0
Number of Minor alarms:     0

 Slot        Sensor          Current State   Reading
Threshold(Minor,Major,Critical,Shutdown)
 ----------  --------------  --------------- ------------
---------------------------------------
 R0          Temp: TS1       Normal          42    Celsius      (75 ,80 ,90 ,na )(Celsius)
 R0          Temp: TS2       Normal          37    Celsius      (75 ,80 ,90 ,na )(Celsius)
```

The output from an snmpwalk would look similar to this:

```
[root@sg-centos-hv ~]# snmpwalk -v 2c -c public 33.33.33.204 1.3.6.1.4.1.9.9.13.1.3.1
SNMPv2-SMI::enterprises.9.9.13.1.3.1.2.1 = STRING: "Sensor 1"
SNMPv2-SMI::enterprises.9.9.13.1.3.1.3.1 = Gauge32: 48
SNMPv2-SMI::enterprises.9.9.13.1.3.1.4.1 = INTEGER: 93
SNMPv2-SMI::enterprises.9.9.13.1.3.1.5.1 = INTEGER: 0
SNMPv2-SMI::enterprises.9.9.13.1.3.1.6.1 = INTEGER: 1
SNMPv2-SMI::enterprises.9.9.13.1.3.1.7.1 = INTEGER: 0
```

The ciscoEnvMonTemperatureStatusEntry oid is 1.3.6.1.4.1.9.9.13.1.3.1:

- ciscoEnvMonTemperatureStatusIndex (.1)
- ciscoEnvMonTemperatureStatusDescr (.2)
- ciscoEnvMonTemperatureStatusValue (.3)
- ciscoEnvMonTemperatureThreshold (.4)
- ciscoEnvMonTemperatureLastShutdown (.5)
- ciscoEnvMonTemperatureStatus (.6)

Additional References

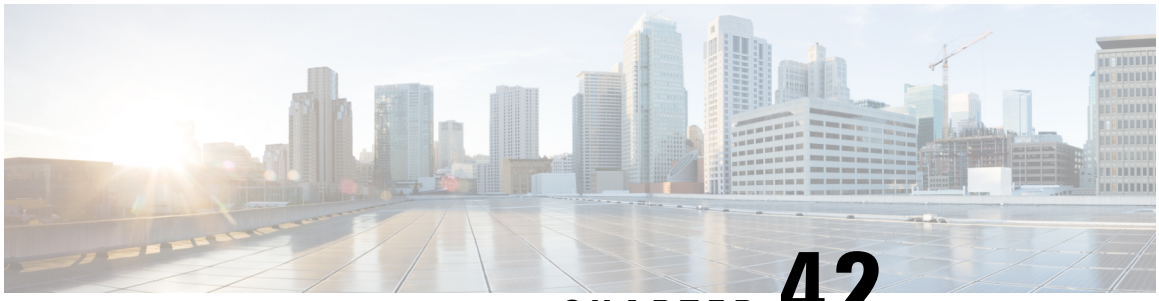
The following sections provide references related to the power efficiency management feature.

MIBs

MIBs	MIBs Link
CISCO-ENTITY-FRU-CONTROL-MIB	To locate and download MIBs for selected platforms, Cisco IOS releases, and feature sets, use the Cisco MIB Locator at: http://www.cisco.com/go/mibs.

Technical Assistance

Description	Link
The Cisco Support website provides extensive online resources, including documentation and tools for troubleshooting and resolving technical issues with Cisco products and technologies. To receive security and technical information about your products, you can subscribe to various services, such as the Product Alert Tool (accessed from Field Notices), the Cisco Technical Services Newsletter, and Really Simple Syndication (RSS) Feeds. Access to most tools on the Cisco Support website requires a Cisco.com user ID and password.	http://www.cisco.com/cisco/web/support/index.html



CHAPTER 42

Process Health Monitoring

This chapter describes how to manage and monitor the health of various components of your router. It contains the following sections:

- Monitoring Control Plane Resources, on page 401
- Monitoring Hardware Using Alarms, on page 410
- Network Management System Alerts a Network Administrator when an Alarm is Reported Through SNMP, on page 411

Monitoring Control Plane Resources

The following sections explain the details of memory and CPU monitoring from the perspective of the Cisco IOS process and the overall control plane:

Avoiding Problems Through Regular Monitoring

Processes should provide monitoring and notification of their status/health to ensure correct operation. When a process fails, a syslog error message is displayed and either the process is restarted or the router is rebooted. A syslog error message is displayed when a monitor detects that a process is stuck or has crashed. If the process can be restarted, it is restarted; else, the router is restarted.

Monitoring system resources enables you to detect potential problems before they occur, thus avoiding outages. It also establishes a baseline for a normal system load. You can use this information as a basis for comparison, when you upgrade hardware or software to see if the upgrade has affected resource usage.

Cisco IOS Process Resources

You can view CPU utilization statistics on active processes and see the amount of memory being used in these processes using the **show memory** command and the **show process cpu** command. These commands provide a representation of memory and CPU utilization from the perspective of only the Cisco IOS process; they do not include information for resources on the entire platform. When the **show memory** command is used in a system with 4 GB RAM running a single Cisco IOS process, the following memory usage is displayed:

```
Router# show memory
Tracekey : 1#b93c0f1c0d5d16ddc3ab8e54342a8dd5

               Head    Total(b)     Used(b)     Free(b)   Lowest(b)  Largest(b)
Processor  7F5F358048   1997625144   290200588   1707424556   487419128   1509949348
```

```
reserve P  7F5F3580A0      102404          92      102312      102312      102312
lsmpi_io  7F5D9901A8     6295128     6294304         824         824         412
Dynamic heap limit(MB) 1440      Use(MB) 0

          Processor memory

Address      Bytes      Prev       Next    Ref   PrevF     NextF          what
   Alloc PC
7F5F358048 0000102408 00000000 7F5F3710A8 001  -------- -------- *Init*
:555F22A000+5E5691C
7F5F3710A8 0000000056 7F5F358048 7F5F371138 001  -------- -------- *Init*
:555F22A000+5E56938
7F5F371138 0000008224 7F5F3710A8 7F5F3731B0 001  -------- -------- *Init*
:555F22A000+5E56958
7F5F3731B0 0000000296 7F5F371138 7F5F373330 001  -------- -------- *Init*
:555F22A000+86B9178
7F5F373330 0000000568 7F5F3731B0 7F5F3735C0 001  -------- -------- *Init*
:555F22A000+86BEE5C
7F5F3735C0 0000032776 7F5F373330 7F5F37B620 001  -------- -------- Managed Chunk Q
:555F22A000+86AAC38
7F5F37B620 0000000056 7F5F3735C0 7F5F37B6B0 001  -------- -------- *Init*
:555F22A000+5EA29DC
7F5F37B6B0 0000032776 7F5F37B620 7F5F383710 001  -------- -------- Queue Pair - Q
:555F22A000+86D3A28
7F5F383710 0000012808 7F5F37B6B0 7F5F386970 001  -------- -------- *Init*
:555F22A000+11EFF930
7F5F386970 0000032776 7F5F383710 7F5F38E9D0 001  -------- -------- List Elements
:555F22A000+86798AC
7F5F38E9D0 0000032776 7F5F386970 7F5F396A30 001  -------- -------- List Headers
:555F22A000+86798EC
7F5F396A30 0000032776 7F5F38E9D0 7F5F39EA90 001  -------- -------- IOSXE Process S
:555F22A000+984FEE0
7F5F39EA90 0000032776 7F5F396A30 7F5F3A6AF0 001  -------- -------- IOSXE Queue Pro
:555F22A000+984FF24
7F5F3A6AF0 0000065544 7F5F39EA90 7F5F3B6B50 001  -------- -------- IOSXE Queue Bal
:555F22A000+984FF68
7F5F3B6B50 0000000328 7F5F3A6AF0 7F5F3B6CF0 001  -------- -------- *Init*
:555F22A000+11EF88AC
7F5F3B6CF0 0000000328 7F5F3B6B50 7F5F3B6E90 001  -------- -------- *Init*
:555F22A000+11EF88AC
7F5F3B6E90 0000000192 7F5F3B6CF0 7F5F3B6FA8 001  -------- -------- SDB String
:555F22A000+8629F60
7F5F3B6FA8 0000036872 7F5F3B6E90 7F5F3C0008 001  -------- -------- *Init*
:555F22A000+98482F4
7F5F3C0008 0000010008 7F5F3B6FA8 7F5F3C2778 001  -------- -------- Platform VM Pag
:555F22A000+986FC68
7F5F3C2778 0000002008 7F5F3C0008 7F5F3C2FA8 001  -------- -------- *Init*
iosd_crb_crankshaft_unix:7F83050000+6D850
7F5F3C2FA8 0000200712 7F5F3C2778 7F5F3F4008 001  -------- -------- Interrupt Stack
:555F22A000+98482F4
7F5F3F4008 0000000328 7F5F3C2FA8 7F5F3F41A8 001  -------- -------- *Init*
:555F22A000+11EF88AC
7F5F3F41A8 0000002008 7F5F3F4008 7F5F3F49D8 001  -------- -------- Watcher Message
:555F22A000+86DE21C
7F5F3F49D8 0000000360 7F5F3F41A8 7F5F3F4B98 001  -------- -------- Process Events
:555F22A000+86D94A4
7F5F3F4B98 0000000328 7F5F3F49D8 7F5F3F4D38 001  -------- -------- *Init*
:555F22A000+11EF88AC
7F5F3F4D38 0000000184 7F5F3F4B98 7F5F3F4E48 001  -------- -------- *Init*
:555F22A000+86C945C
7F5F3F4E48 0000000264 7F5F3F4D38 7F5F3F4FA8 001  -------- -------- *Init*
:555F22A000+86C945C
7F5F3F4FA8 0000036872 7F5F3F4E48 7F5F3FE008 001  -------- -------- *Init*
:555F22A000+98482F4
```

```
7F5F3FE008 0000000328 7F5F3F4FA8 7F5F3FE1A8 001  -------- -------- *Init*
:555F22A000+11EF88AC
7F5F3FE1A8 0000001504 7F5F3FE008 7F5F3FE7E0 001  -------- -------- Reg Function Se
:555F22A000+868AE50
7F5F3FE7E0 0000001504 7F5F3FE1A8 7F5F3FEE18 001  -------- -------- Reg Function Se
:555F22A000+868AEE0
7F5F3FEE18 0000000064 7F5F3FE7E0 7F5F3FEEB0 001  -------- -------- Parser Linkage
:555F22A000+5D98838
7F5F3FEEB0 0000000160 7F5F3FEE18 7F5F3FEFA8 001  -------- -------- *Init*
:555F22A000+530A17C
7F5F3FEFA8 0000036872 7F5F3FEEB0 7F5F408008 001  -------- -------- *Init*
:555F22A000+98482F4
7F5F408008 0000000328 7F5F3FEFA8 7F5F4081A8 001  -------- -------- *Init*
:555F22A000+11EF88AC
7F5F4081A8 0000000760 7F5F408008 7F5F4084F8 001  -------- -------- *Init*
:555F22A000+868BC18
7F5F4084F8 0000000576 7F5F4081A8 7F5F408790 001  -------- -------- *Init*
:555F22A000+868BC18
7F5F408790 0000000400 7F5F4084F8 7F5F408978 001  -------- -------- *Init*
:555F22A000+868BC18
7F5F408978 0000000488 7F5F408790 7F5F408BB8 001  -------- -------- *Init*
:555F22A000+868BC18
7F5F408BB8 0000000920 7F5F408978 7F5F408FA8 001  -------- -------- *Init*
:555F22A000+868BC18
7F5F408FA8 0000200712 7F5F408BB8 7F5F43A008 001  -------- -------- Interrupt Stack
:555F22A000+98482F4
7F5F43A008 0000000968 7F5F408FA8 7F5F43A428 001  -------- -------- *Init*
iosd_crb_crankshaft_unix:7F83050000+378C0
7F5F43A428 0000000280 7F5F43A008 7F5F43A598 001  -------- -------- *Init*
:555F22A000+A3CE294
7F5F43A598 0000000896 7F5F43A428 7F5F43A970 001  -------- -------- Watched Message
:555F22A000+86DE1E8
7F5F43A970 0000001320 7F5F43A598 7F5F43AEF0 001  -------- -------- Process
:555F22A000+86E50BC
7F5F43AEF0 0000000096 7F5F43A970 7F5F43AFA8 001  -------- -------- *Init*
:555F22A000+868BC18
7F5F43AFA8 0000036872 7F5F43AEF0 7F5F444008 001  -------- -------- *Init*
:555F22A000+98482F4
7F5F444008 0000003008 7F5F43AFA8 7F5F444C20 001  -------- -------- Watched Semapho
:555F22A000+86DE180
7F5F444C20 0000000360 7F5F444008 7F5F444DE0 001  -------- -------- Process Events
:555F22A000+86D94A4
7F5F444DE0 0000000368 7F5F444C20 7F5F444FA8 001  -------- -------- *Init*
:555F22A000+11EF88AC
7F5F444FA8 0000200712 7F5F444DE0 7F5F476008 001  -------- -------- Interrupt Stack
:555F22A000+98482F4
7F5F476008 0000002336 7F5F444FA8 7F5F476980 001  -------- -------- Process Array
:555F22A000+86E4F94
7F5F476980 0000000184 7F5F476008 7F5F476A90 001  -------- -------- *Init*
:555F22A000+86C945C
7F5F476A90 0000000184 7F5F476980 7F5F476BA0 001  -------- -------- *Init*
:555F22A000+86C945C
7F5F476BA0 0000000184 7F5F476A90 7F5F476CB0 001  -------- -------- *Init*
:555F22A000+86C945C
7F5F476CB0 0000000184 7F5F476BA0 7F5F476DC0 001  -------- -------- *Init*
:555F22A000+86C945C
7F5F476DC0 0000000184 7F5F476CB0 7F5F476ED0 001  -------- -------- *Init*
:555F22A000+86C945C
7F5F476ED0 0000000128 7F5F476DC0 7F5F476FA8 001  -------- -------- *Init*
:555F22A000+868BC18
7F5F476FA8 0000036872 7F5F476ED0 7F5F480008 001  -------- -------- *Init*
:555F22A000+98482F4
7F5F480008 0000001320 7F5F476FA8 7F5F480588 001  -------- -------- Process
:555F22A000+86E50BC
```

```
7F5F480588 0000000184 7F5F480008 7F5F480698 001  -------- -------- *Init*
:555F22A000+86C945C
7F5F480698 0000000184 7F5F480588 7F5F4807A8 001  -------- -------- *Init*
:555F22A000+86C945C
7F5F4807A8 0000000184 7F5F480698 7F5F4808B8 001  -------- -------- *Init*
:555F22A000+86C945C
7F5F4808B8 0000000184 7F5F4807A8 7F5F4809C8 001  -------- -------- *Init*
:555F22A000+86C945C
7F5F4809C8 0000000184 7F5F4808B8 7F5F480AD8 001  -------- -------- *Init*
:555F22A000+86C945C
7F5F480AD8 0000000184 7F5F4809C8 7F5F480BE8 001  -------- -------- *Init*
:555F22A000+86C945C
7F5F480BE8 0000000184 7F5F480AD8 7F5F480CF8 001  -------- -------- *Init*
:555F22A000+86C945C
7F5F480CF8 0000000096 7F5F480BE8 7F5F480DB0 001  -------- -------- *Init*
:555F22A000+86C940C
7F5F480DB0 0000000096 7F5F480CF8 7F5F480E68 001  -------- -------- Init
:555F22A000+862A110
7F5F480E68 0000000232 7F5F480DB0 7F5F480FA8 001  -------- -------- *Init*
:555F22A000+60E2660
7F5F480FA8 0000200712 7F5F480E68 7F5F4B2008 001  -------- -------- Interrupt Stack
:555F22A000+98482F4
7F5F4B2008 0000003008 7F5F480FA8 7F5F4B2C20 001  -------- -------- Reg Function Li
:555F22A000+868AE80
7F5F4B2C20 0000000064 7F5F4B2008 7F5F4B2CB8 001  -------- -------- Parser Linkage
:555F22A000+5D98838
7F5F4B2CB8 0000000064 7F5F4B2C20 7F5F4B2D50 001  -------- -------- Parser Linkage
:555F22A000+5D98A78
7F5F4B2D50 0000000080 7F5F4B2CB8 7F5F4B2DF8 001  -------- -------- Init
:555F22A000+5E87AC0
7F5F4B2DF8 0000000200 7F5F4B2D50 7F5F4B2F18 001  -------- -------- Init
:555F22A000+5E87AC0
7F5F4B2F18 0000000056 7F5F4B2DF8 7F5F4B2FA8 001  -------- -------- Init
:555F22A000+54DD60C
7F5F4B2FA8 0000036872 7F5F4B2F18 7F5F4BC008 001  -------- -------- *Init*
:555F22A000+98482F4
7F5F4BC008 0000001504 7F5F4B2FA8 7F5F4BC640 001  -------- -------- Reg Function Ca
:555F22A000+868AF10
7F5F4BC640 0000000224 7F5F4BC008 7F5F4BC778 001  -------- -------- *Init*
:555F22A000+868BC18
7F5F4BC778 0000000224 7F5F4BC640 7F5F4BC8B0 001  -------- -------- *Init*
:555F22A000+868BC18
7F5F4BC8B0 0000000328 7F5F4BC778 7F5F4BCA50 001  -------- -------- *Init*
:555F22A000+868BC18
7F5F4BCA50 0000000328 7F5F4BC8B0 7F5F4BCBF0 001  -------- -------- *Init*
:555F22A000+868BC18
7F5F4BCBF0 0000000328 7F5F4BCA50 7F5F4BCD90 001  -------- -------- *Init*
:555F22A000+868BC18
7F5F4BCD90 0000000216 7F5F4BCBF0 7F5F4BCEC0 001  -------- -------- Init
:555F22A000+5E87AC0
7F5F4BCEC0 0000000144 7F5F4BCD90 7F5F4BCFA8 001  -------- -------- Init
:555F22A000+530F7A4
7F5F4BCFA8 0000200712 7F5F4BCEC0 7F5F4EE008 001  -------- -------- Interrupt Stack
:555F22A000+98482F4
7F5F4EE008 0000006888 7F5F4BCFA8 7F5F4EFB48 001  -------- -------- TTY data
:555F22A000+85C9E44
7F5F4EFB48 0000004104 7F5F4EE008 7F5F4F0BA8 001  -------- -------- TTY Input Buf
:555F22A000+85CBD60
7F5F4F0BA8 0000004104 7F5F4EFB48 7F5F4F1C08 001  -------- -------- TTY Output Buf
:555F22A000+85CBDB0
7F5F4F1C08 0000024584 7F5F4F0BA8 7F5F4F7C68 001  -------- -------- proc_hist_lmt_v
:555F22A000+BA3B718
7F5F4F7C68 0000008200 7F5F4F1C08 7F5F4F9CC8 001  -------- -------- proc_hist_lmt_v
:555F22A000+BA3B74C
```

```
7F5F4F9CC8 0000008200 7F5F4F7C68 7F5F4FBD28 001  -------- -------- proc_hist_lmt_v
:555F22A000+BA3B784
7F5F4FBD28 0000005008 7F5F4F9CC8 7F5F4FD110 001  -------- -------- messages
:555F22A000+86DE040
7F5F4FD110 0000005008 7F5F4FBD28 7F5F4FE4F8 001  -------- -------- Watched message
:555F22A000+86DE078
7F5F4FE4F8 0000020008 7F5F4FD110 7F5F503378 001  -------- -------- Watched Queue
:555F22A000+86DE0AC
7F5F503378 0000065544 7F5F4FE4F8 7F5F5133D8 001  -------- -------- Watched Queue I
:555F22A000+86DE0E4
7F5F5133D8 0000020008 7F5F503378 7F5F518258 001  -------- -------- Watched Boolean
:555F22A000+86DE118
7F5F518258 0000020008 7F5F5133D8 7F5F51D0D8 001  -------- -------- Watched Bitfiel
:555F22A000+86DE14C
7F5F51D0D8 0000010008 7F5F518258 7F5F51F848 001  -------- -------- Watcher Info
:555F22A000+86DE1B4
7F5F51F848 0000010008 7F5F51D0D8 7F5F521FB8 001  -------- -------- Read/Write Lock
:555F22A000+86DE250
7F5F521FB8 0000001232 7F5F51F848 7F5F5224E0 001  -------- -------- *Init*
:555F22A000+868BC18
7F5F5224E0 0000000064 7F5F521FB8 7F5F522578 001  -------- -------- Init
:555F22A000+8D8F3A0
7F5F522578 0000002008 7F5F5224E0 7F5F522DA8 001  -------- -------- Injected msg CB
:555F22A000+ACBBF08
7F5F522DA8 0000000064 7F5F522578 7F5F522E40 001  -------- -------- SDB String
:555F22A000+8629F60
7F5F522E40 0000000056 7F5F522DA8 7F5F522ED0 001  -------- -------- *Init*
:555F22A000+5EA29DC
7F5F522ED0 0000000128 7F5F522E40 7F5F522FA8 001  -------- -------- XOS_MEM_XDT
:555F22A000+894FE1C
7F5F522FA8 0000028104 7F5F522ED0 7F5F529DC8 001  -------- -------- Process Stack
:555F22A000+98482F4
7F5F529DC8 0000000096 7F5F522FA8 7F5F529E80 001  -------- -------- Init
:555F22A000+862A110
7F5F529E80 0000000208 7F5F529DC8 7F5F529FA8 001  -------- -------- *Init*
:555F22A000+86C8D58
7F5F529FA8 0000016104 7F5F529E80 7F5F52DEE8 001  -------- -------- Process Stack
:555F22A000+98482F4
7F5F52DEE8 0000032776 7F5F529FA8 7F5F535F48 001  -------- -------- List Elements
:555F22A000+8679DCC
7F5F535F48 0000032776 7F5F52DEE8 7F5F53DFA8 001  -------- -------- List Elements
:555F22A000+8679DCC
7F5F53DFA8 0000032776 7F5F535F48 7F5F546008 001  -------- -------- List Elements
:555F22A000+8679DCC
7F5F546008 0000032776 7F5F53DFA8 7F5F54E068 001  -------- -------- List Elements
:555F22A000+8679DCC
7F5F54E068 0000032776 7F5F546008 7F5F5560C8 001  -------- -------- List Elements
:555F22A000+8679DCC
7F5F5560C8 0000032776 7F5F54E068 7F5F55E128 001  -------- -------- List Elements
:555F22A000+8679DCC
7F5F55E128 0000032776 7F5F5560C8 7F5F566188 001  -------- -------- List Elements
:555F22A000+8679DCC
7F5F566188 0000032776 7F5F55E128 7F5F56E1E8 001  -------- -------- List Elements
:555F22A000+8679DCC
7F5F56E1E8 0000005008 7F5F566188 7F5F56F5D0 001  -------- -------- Reg Function 12
:555F22A000+868AE1C
7F5F56F5D0 0000020008 7F5F56E1E8 7F5F574450 001  -------- -------- Subsys Malloc I
:555F22A000+86875C8
7F5F574450 0000001176 7F5F56F5D0 7F5F574940 001  -------- -------- SPA variable ms
:555F22A000+B9B3700
7F5F574940 0000000920 7F5F574450 7F5F574D30 001  -------- -------- SAMsgThread
:555F22A000+54E49DC
7F5F574D30 0000000064 7F5F574940 7F5F574DC8 001  -------- -------- Parser Linkage
:555F22A000+5D98A78
```

```
7F5F574DC8 0000000064 7F5F574D30 7F5F574E60 001  -------- -------- Parser Linkage
:555F22A000+5D98838
7F5F574E60 0000000064 7F5F574DC8 7F5F574EF8 001  -------- -------- Parser Linkage
:555F22A000+5D98A78
7F5F574EF8 0000000088 7F5F574E60 7F5F574FA8 001  -------- -------- Init
:555F22A000+54DD60C
7F5F574FA8 0000000968 7F5F574EF8 7F5F5753C8 001  -------- -------- Crypto CA
:555F22A000+8DC26C0
7F5F5753C8 0000000216 7F5F574FA8 7F5F5754F8 001  -------- -------- Crypto CA
:555F22A000+8DC2588
7F5F5754F8 0000002648 7F5F5753C8 7F5F575FA8 000  7F609C3C28 7F69C82D38 (coalesced)
:555F22A000+52BC2B8
7F5F575FA8 0000028104 7F5F5754F8 7F5F57CDC8 001  -------- -------- Process Stack
:555F22A000+98482F4
7F5F57CDC8 0000013112 7F5F575FA8 7F5F580158 001  -------- -------- SAMsgThread
:555F22A000+5360944
7F5F580158 0000004728 7F5F57CDC8 7F5F581428 001  -------- -------- *Packet Data*
:555F22A000+AF448CC
7F5F581428 0000000968 7F5F580158 7F5F581848 001  -------- -------- Exec
:555F22A000+5D9C004
7F5F581848 0000000600 7F5F581428 7F5F581AF8 001  -------- -------- Ether OAM subbl
:555F22A000+962A100
7F5F581AF8 0000000056 7F5F581848 7F5F581B88 000  7F69F8B620 7F6A906D40 (fragment)
:555F22A000+962A100
7F5F581B88 0000005008 7F5F581AF8 7F5F582F70 001  -------- -------- Reg Function iL
:555F22A000+868AEB0
7F5F582F70 0000065544 7F5F581B88 7F5F592FD0 001  -------- -------- Registry Call S
:555F22A000+8690EFC
7F5F592FD0 0000002584 7F5F582F70 7F5F593A40 001  -------- -------- *Init*
:555F22A000+868BC18
7F5F593A40 0000002080 7F5F592FD0 7F5F5942B8 001  -------- -------- *Init*
:555F22A000+ACB41D8
7F5F5942B8 0000002600 7F5F593A40 7F5F594D38 001  -------- -------- *Init*
:555F22A000+ACB41D8
7F5F594D38 0000020008 7F5F5942B8 7F5F599BB8 001  -------- -------- Peer uid cb chu
:555F22A000+ACBBE98
```

The **show process cpu** command displays Cisco IOS CPU utilization average:

```
Router# show process cpu
CPU utilization for five seconds: 0%/0%; one minute: 0%; five minutes: 0%
PID Runtime(ms)     Invoked      uSecs   5Sec   1Min   5Min TTY Process
   1           8          31        258  0.00%  0.00%  0.00%   0 Chunk Manager
   2           8       10141          0  0.00%  0.00%  0.00%   0 Load Meter
   3           0           1          0  0.00%  0.00%  0.00%   0 PKI Trustpool
   4           0           1          0  0.00%  0.00%  0.00%   0 Retransmission o
   5           0           1          0  0.00%  0.00%  0.00%   0 IPC ISSU Dispatc
   6          28          13       2153  0.00%  0.00%  0.00%   0 RF Slave Main Th
   7           0           1          0  0.00%  0.00%  0.00%   0 EDDRI_MAIN
   8           0           1          0  0.00%  0.00%  0.00%   0 RO Notify Timers
   9       40648        7844       5182  0.00%  0.06%  0.05%   0 Check heaps
  10          16         845         18  0.00%  0.00%  0.00%   0 Pool Manager
  11           0           1          0  0.00%  0.00%  0.00%   0 DiscardQ Backgro
  12           0           2          0  0.00%  0.00%  0.00%   0 Timers
  13           0         176          0  0.00%  0.00%  0.00%   0 WATCH_AFS
  14           0           1          0  0.00%  0.00%  0.00%   0 MEMLEAK PROCESS
  15           0           1          0  0.00%  0.00%  0.00%   0 ARP Input
  16           4       52892          0  0.00%  0.00%  0.00%   0 ARP Background
  17           0           2          0  0.00%  0.00%  0.00%   0 ATM Idle Timer
  18           0           1          0  0.00%  0.00%  0.00%   0 ATM ASYNC PROC
  19           0           1          0  0.00%  0.00%  0.00%   0 CEF MIB API
```

```
  20           0           1           0  0.00%  0.00%  0.00%   0 AAA_SERVER_DEADT
  21           0           1           0  0.00%  0.00%  0.00%   0 Policy Manager
  22           0           2           0  0.00%  0.00%  0.00%   0 DDR Timers
  23          16          10        1600  0.00%  0.00%  0.00%   0 Entity MIB API
  24         120          27        4444  0.00%  0.00%  0.00%   0 PrstVbl
  25           0           2           0  0.00%  0.00%  0.00%   0 Serial Backgroun
  26           0           1           0  0.00%  0.00%  0.00%   0 RMI RM Notify Wa
  27           0           2           0  0.00%  0.00%  0.00%   0 ATM AutoVC Perio
  28           0           2           0  0.00%  0.00%  0.00%   0 ATM VC Auto Crea
  29           4       25354           0  0.00%  0.00%  0.00%   0 IOSXE heartbeat
  30           0          86           0  0.00%  0.00%  0.00%   0 Btrace time base
  31           0          10           0  0.00%  0.00%  0.00%   0 DB Lock Manager
  32           4       50697           0  0.00%  0.00%  0.00%   0 GraphIt
  33           0           1           0  0.00%  0.00%  0.00%   0 DB Notification
  34           0           1           0  0.00%  0.00%  0.00%   0 IPC Apps Task
  35           0           1           0  0.00%  0.00%  0.00%   0 ifIndex Receive
  36           0       10142           0  0.00%  0.00%  0.00%   0 IPC Event Notifi
  37           0       49518           0  0.00%  0.00%  0.00%   0 IPC Mcast Pendin
  38           0           1           0  0.00%  0.00%  0.00%   0 Platform appsess
  39           0         846           0  0.00%  0.00%  0.00%   0 IPC Dynamic Cach
  40           0       10142           0  0.00%  0.00%  0.00%   0 IPC Service NonC
  41           0           1           0  0.00%  0.00%  0.00%   0 IPC Zone Manager
  42           8       49518           0  0.00%  0.00%  0.00%   0 IPC Periodic Tim
  43           0       49518           0  0.00%  0.00%  0.00%   0 IPC Deferred Por
  44           0           1           0  0.00%  0.00%  0.00%   0 IPC Process leve
  45           0           1           0  0.00%  0.00%  0.00%   0 IPC Seat Manager
```

show process cpu platform sorted

```
CPU utilization for five seconds:  9%, one minute: 10%, five minutes: 10%
Core 0: CPU utilization for five seconds:  3%, one minute:  4%, five minutes:  4%
Core 1: CPU utilization for five seconds:  4%, one minute:  4%, five minutes:  4%
Core 2: CPU utilization for five seconds:  2%, one minute:  2%, five minutes:  2%
Core 3: CPU utilization for five seconds: 38%, one minute: 38%, five minutes: 38%
   Pid    PPid    5Sec    1Min    5Min  Status        Size  Name
--------------------------------------------------------------------------------
18700   18679     44%     44%     44%  S           235192  qfp-ucode-avent
  5226    5216      2%      3%      3%  S           697124  linux_iosd-imag
24238   24231      1%      1%      1%  S             8288  ngiolite
18412   18398      1%      1%      1%  S           135696  fman_fp_image
30574       2      0%      0%      0%  S                0  kworker/0:3
24231   16366      0%      0%      0%  S             2460  pman
24025   23998      0%      0%      0%  S             3392  nginx
24024   23998      0%      0%      0%  S             4300  nginx
23998   23990      0%      0%      0%  S             7944  nginx
23990    4251      0%      0%      0%  S             2460  pman
23605   23599      0%      0%      0%  S             7988  ngiolite
23599   16366      0%      0%      0%  S             2464  pman
23330   23309      0%      0%      0%  S            39600  iomd
23309   16366      0%      0%      0%  S             2460  pman
21981   15002      0%      0%      0%  S              416  sleep
21935   21906      0%      0%      0%  S            38680  iomd
21906   16366      0%      0%      0%  S             2460  pman
21830   13884      0%      0%      0%  S              416  sleep
21694       2      0%      0%      0%  S                0  kworker/0:0
21042       2      0%      0%      0%  S                0  kworker/u8:4
21041       2      0%      0%      0%  S                0  kworker/u8:3
21040       2      0%      0%      0%  S                0  kworker/u8:0
20737       2      0%      0%      0%  S                0  kworker/1:3
20731       2      0%      0%      0%  S                0  SarIosdMond
20574   20548      0%      0%      0%  S            12004  btman
20548   16921      0%      0%      0%  S             2432  pman
20180   20146      0%      0%      0%  S            17428  cman_fp
20146   16921      0%      0%      0%  S             2432  pman
20135   20105      0%      0%      0%  S            12228  btman
```

```
20105   16366      0%      0%      0%  S             2432  pman
20093       2      0%      0%      0%  S                0  kworker/0:1
19819   19796      0%      0%      0%  S           107992  cpp_cp_svr
19796   16921      0%      0%      0%  S             2436  pman
19549   19528      0%      0%      0%  S            18948  cmcc
19541   19512      0%      0%      0%  S            35124  cpp_driver
19528   16366      0%      0%      0%  S             2432  pman
19512   16921      0%      0%      0%  S             2432  pman
19280   19243      0%      0%      0%  S            38708  cpp_ha_top_leve
19243   16921      0%      0%      0%  S             2436  pman
18966   18959      0%      0%      0%  S            49916  cpp_sp_svr
18959   16921      0%      0%      0%  S             2436  pman
18877   18862      0%      0%      0%  S             5780  pttcd
18862    4251      0%      0%      0%  S             2432  pman
18856       2      0%      0%      0%  S                0  kworker/1:1
18711   18691      0%      0%      0%  S            10352  hman
18691   16366      0%      0%      0%  S             2432  pman
18679   16921      0%      0%      0%  S             2436  pman
18517   18495      0%      0%      0%  S            60720  pubd
18495    4251      0%      0%      0%  S             2432  pman
18398   16921      0%      0%      0%  S             2432  pman
18211       2      0%      0%      0%  S                0  kworker/0:2
18140   18120      0%      0%      0%  S            10352  hman
18120   16921      0%      0%      0%  S             2436  pman
17448   16921      0%      0%      0%  S              428  inotifywait
17253       2      0%      0%      0%  S                0  kworker/1:0
17204       1      0%      0%      0%  S             2064  rotee
16921       1      0%      0%      0%  S             5512  pvp.sh
16744   16366      0%      0%      0%  S              428  inotifywait
16582       1      0%      0%      0%  S             2060  rotee
16366       1      0%      0%      0%  S             5512  pvp.sh
15627       2      0%      0%      0%  S                0  bioset
15626       2      0%      0%      0%  S                0  dmcrypt_write
15625       2      0%      0%      0%  S                0  kcryptd
15624       2      0%      0%      0%  S                0  kcryptd_io
15623       2      0%      0%      0%  S                0  bioset
15621       2      0%      0%      0%  S                0  kdmflush
15618       2      0%      0%      0%  S                0  loop1
15563       2      0%      0%      0%  S                0  ext4-rsv-conver
15562       2      0%      0%      0%  S                0  jbd2/mmcblk0p1-
15023       2      0%      0%      0%  S                0  kworker/u8:1
15002   14992      0%      0%      0%  S             1440  sort_files_by_i
14992    4251      0%      0%      0%  S             2416  pman
13884   13874      0%      0%      0%  S             2816  flash_check.sh
```

Overall Control Plane Resources

Control plane memory and CPU utilization on each control processor allows you to keep a tab on the overall control plane resources. You can use the **show platform software status control-processor brief** command (summary view) or the **show platform software status control-processor command** (detailed view) to view control plane memory and CPU utilization information.

All control processors should show status, Healthy. Other possible status values are Warning and Critical. Warning indicates that the router is operational, but that the operating level should be reviewed. Critical implies that the router is nearing failure.

If you see a Warning or Critical status, take the following actions:

- Reduce the static and dynamic loads on the system by reducing the number of elements in the configuration or by limiting the capacity for dynamic services.

- Reduce the number of routes and adjacencies, limit the number of ACLs and other rules, reduce the number of VLANs, and so on.

The following sections describe the fields in the **show platform software status control-processor** command output.

Load Average

Load average represents the process queue or process contention for CPU resources. For example, on a single-core processor, an instantaneous load of 7 would mean that seven processes are ready to run, one of which is currently running. On a dual-core processor, a load of 7 would mean that seven processes are ready to run, two of which are currently running.

Memory Utilization

Memory utilization is represented by the following fields:

- Total—Total system memory
- Used—Consumed memory
- Free—Available memory
- Committed—Virtual memory committed to processes

CPU Utilization

CPU utilization is an indication of the percentage of time the CPU is busy, and is represented by the following fields:

- CPU—Allocated processor
- User—Non-Linux kernel processes
- System—Linux kernel process
- Nice—Low-priority processes
- Idle—Percentage of time the CPU was inactive
- IRQ—Interrupts
- SIRQ—System Interrupts
- IOwait—Percentage of time CPU was waiting for I/O

Example: show platform software status control-processor Command

The following are some examples of using the **show platform software status control-processor** command:

```
Router# show platform software status control-processor
RP0: online, statistics updated 10 seconds ago
Load Average: healthy
  1-Min: 1.28, status: healthy, under 5.00
  5-Min: 0.74, status: healthy, under 5.00
  15-Min: 0.78, status: healthy, under 5.00
```

```
Memory (kb): healthy
  Total: 8154204
  Used: 2282364 (28%), status: healthy
  Free: 5871840 (72%)
  Committed: 2025108 (25%), under 90%
Per-core Statistics
CPU0: CPU Utilization (percentage of time spent)
  User:  2.46, System:  5.53, Nice:  0.00, Idle: 90.87
  IRQ:  0.82, SIRQ:  0.20, IOwait:  0.10
CPU1: CPU Utilization (percentage of time spent)
  User:  2.24, System:  5.91, Nice:  0.00, Idle: 90.91
  IRQ:  0.71, SIRQ:  0.20, IOwait:  0.00
CPU2: CPU Utilization (percentage of time spent)
  User:  0.50, System:  1.82, Nice:  0.00, Idle: 97.16
  IRQ:  0.50, SIRQ:  0.00, IOwait:  0.00
CPU3: CPU Utilization (percentage of time spent)
  User: 13.03, System: 12.88, Nice:  0.00, Idle: 62.51
  IRQ: 11.55, SIRQ:  0.00, IOwait:  0.00


Router# show platform software status control-processor brief
Load Average
Slot  Status  1-Min  5-Min 15-Min
  RP0 Healthy   0.99   0.72   0.77

Memory (kB)
Slot  Status    Total     Used (Pct)     Free (Pct) Committed (Pct)
  RP0 Healthy  8154204  2281012 (28%)  5873192 (72%)   2032232 (25%)

CPU Utilization
Slot  CPU   User System   Nice   Idle    IRQ   SIRQ IOwait
  RP0    0   1.02   1.84   0.00  96.30   0.61   0.10   0.10
         1   0.72   1.85   0.00  96.60   0.61   0.20   0.00
         2   0.50   1.62   0.00  97.25   0.60   0.00   0.00
         3  11.78  14.28   0.00  62.44  11.34   0.14   0.00


Boot Flash Disk Monitoring

*Aug 24 07:48:31.088 GMT: %FLASH_CHECK-3-DISK_QUOTA: R0/0: flash_check: Flash disk quota
exceeded
[free space is 83820 kB] - Please clean up files on flash1.
```

Monitoring Hardware Using Alarms

This section contains the following:

Router Design and Monitoring Hardware

The router sends alarm notifications when problems are detected, allowing you to monitor the network remotely. You do not need to use **show** commands to poll devices on a routine basis; however, you can perform onsite monitoring if you choose.

BootFlash Disk Monitoring

The bootflash disk must have enough free space to store two core dumps. This condition is monitored, and if the bootflash disk is too small to store two core dumps, a syslog alarm is generated, as shown in the following example:

```
Oct  6 14:10:56.292: %FLASH_CHECK-3-DISK_QUOTA: R0/0: flash_check: Flash disk quota exceeded

[free space is 1429020 kB] - Please clean up files on bootflash.
```

Approaches for Monitoring Hardware Alarms

This section contains the following:

Viewing the Console or Syslog for Alarm Messages

The network administrator can monitor alarm messages by reviewing alarm messages sent to the system console or to a system message log (syslog).

Enabling the logging alarm Command

The **logging alarm** command must be enabled for the system to send alarm messages to a logging device, such as the console or a syslog. This command is not enabled by default.

You can specify the severity level of the alarms to be logged. All the alarms at and above the specified threshold generate alarm messages. For example, the following command sends only critical alarm messages to logging devices:

```
Router(config)# logging alarm critical
```

If alarm severity is not specified, alarm messages for all severity levels are sent to logging devices.

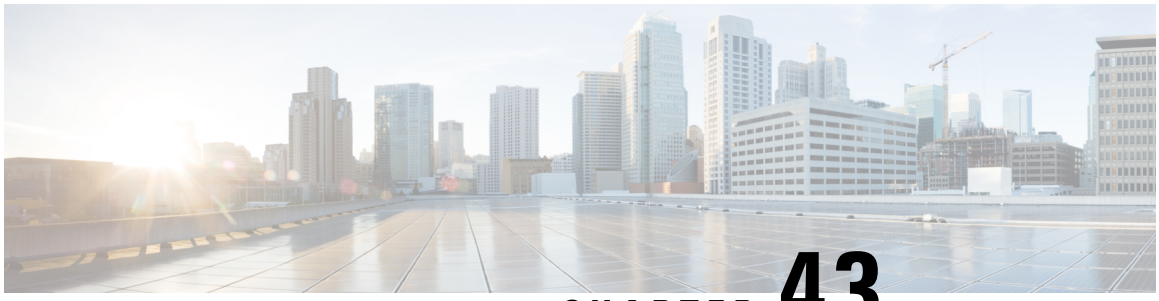
Network Management System Alerts a Network Administrator when an Alarm is Reported Through SNMP

The SNMP is an application-layer protocol that provides a standardized framework and a common language used for monitoring and managing devices in a network.

SNMP provides notification of faults, alarms, and conditions that might affect services. It allows a network administrator to access router information through a network management system (NMS) instead of reviewing logs, polling devices, or reviewing log reports.

To use SNMP to get alarm notification, use the following MIBs:

- ENTITY-MIB, RFC4133 (required for the CISCO-ENTITY-ALARM-MIB, ENTITY-STATE-MIB and CISCO-ENTITY-SENSOR-MIB to work)
- CISCO-ENTITY-ALARM-MIB
- ENTITY-STATE-MIB
- CISCO-ENTITY-SENSOR-MIB (for transceiver environmental alarm information, which is not provided through the CISCO-ENTITY-ALARM-MIB)



CHAPTER 43

WAN Monitoring

This chapter contains the following topics:

- Information About WANMon, on page 413
- Built-in Recovery Actions, on page 413
- Prerequisites, on page 414
- Guidelines and Limitations, on page 414
- Configuring WANMon, on page 415
- Verifying WANMon Configuration, on page 416
- Configuration Examples, on page 417

Information About WANMon

WANMon is a flexible solution to address the WAN link recovery requirements for the following products and interfaces:

- Physical networks: 4G LTE and Ethernet (WAN port)
- Virtual links: Non-crypto map based IPSec tunnels (either legacy or FlexVPN); that is, any IPSec tunnel you configure as an interface.

You enable WANMon to monitor your WAN links and initiate link recovery actions on receipt of link failure triggers.

Built-in Recovery Actions

The following are the three levels of built-in recovery processes specific to the link type:

Link Type	Recovery Actions		
	Level 0 (Immediate)	Level 1 (Active)	Level 2 (Last-Resort)
4G LTE	Clear interface, and then shut/no-shut	Module reload	System reload
Ethernet	Clear interface, and then shut/no-shut	No action taken	System reload

Link Type	Recovery Actions		
	Level 0 (Immediate)	Level 1 (Active)	Level 2 (Last-Resort)
Tunnel	Shut/no-shut	No action taken	System reload

Each level has two time-based thresholds based on which built-in recovery actions are taken. The following are the default settings for each level:

- *threshold* is the wait time in minutes after receipt of a link failure trigger to initiate the recovery action as set in the specified level.
- *mintime* is the frequency to perform the recovery action if the link remains down.

The built-in values are:

Level	threshold	mintime	Description
Level 0	10 min	10 min	Triggers Level 0 actions 10 minutes after the link went down. Repeat no more than every 10 minutes.
Level 1	60 min	60 min	Triggers Level 1 actions 10 minutes after the link went down. Repeat no more than every 60 minutes.
Level 2	480 min	60 min	Triggers Level 2 actions 480 minutes after the link went down. Repeat no more than every 60 minutes.



Note If threshold values are specified as 0, no recovery actions are taken for that level. You can use this to avoid system reload (the built-in Level 2 recovery action) on receipt of a link failure trigger where other WAN links may be operational.

Prerequisites

Ensure that the WANMon module is available. The WANMon module is included in the IOS-XE image as the *tm_wanmon.tcl* policy file.

Guidelines and Limitations

- WANMon automatically performs IP address checking (no user configuration) as required for cellular interfaces.
- For all other interfaces, WANMon never performs IP address checking.
- WANMon indirectly triggers user-specified actions by generating an application event that link resetter applets monitor.

- If your network is live, ensure that you understand the potential impact of any command.

Configuring WANMon

You can enable WANMon on the router and assign WAMMon support to specific interfaces. Optionally, you can override the built-in recovery actions, define custom recovery links, and define an event manager environment policy to set the track object value and disable IP address checking. WANMon is disabled by default.

Procedure

	Command or Action	Purpose
Step 1	**event manager policy** *tm_wanmon.tcl* **authorization bypass**	Enables the WANMon link recovery module. Use **authorization bypass** to avoid authorization for CLIs invoked by this policy.
Step 2	**event manager environment wanmon_if_list** <*instance*> {*interface name* {**ipsla** <*instance*>}}	Configures WANMon for the interfaces in your WAN, and indicates that this is an interface configuration command. **Note** Any environment variable with the prefix wanmon_if_list constitutes an interface configuration. Multiple interfaces are allowed by specifying an instance. Be sure to specify the full interface name (for example, cellular0/4/0 or cellular0/5/0). You can set the IP SLA icmp-echo trigger, if desired. Multiple IP SLA triggers are allowed by specifing an instance. **Note** WANMon only looks at the status of the SLA ID. Even though *icmp-echo* is most common, if needed any other type of SLA probe (for example, *udp-echo*) can be used instead.
Step 3	**event manager environment wanmon_if_listx** {*interface name* {**recovery** *Level0* {*Level1* } *Level2*}}	(Optional) Overrides the built-in thresholds.
Step 4	**publish-event sub-system 798 type 2000 arg1** <*interface name*> **arg2** <*level* >	(Optional) Configures custom recovery actions using link resetter applets. <*interface* > is the full interface name (for example, cellular0/4/0 or cellular0/5/0).

	Command or Action	Purpose
		<*level* > is 0, 1, or 2 to match the desired link recovery action.
Step 5	**{stub** <*track-stub-id* > **}**	(Optional) Allows an event manager environment policy to set the track object value. WANMon can set a track-stub-object value to reflect the link state so that an external applet can track the stub object.
Step 6	**event manager environment wanmon_if_listx** {<*interface name* > {**checkip** <*instance* >}}	(Optional) Disables IP address checking.

What to do next

EXAMPLES

```
event manager policy tm_wanmon.tcl authorization bypass
```

The following examples are Event Manager commands to configure cellular and Ethernet interfaces:

```
event manager environment wanmon_if_list1 {cellular0/4/0 {ipsla 1}}
event manager environment wanmon_if_list2 {GigabitEthernet0/0/0 {ipsla 2}}
```

This example sets custom recovery thresholds:

```
event manager environment wanmon_if_list {cellular0/4/0 {recovery 20 {90 75} 600}
```

Where:

- The Level 0 threshold is set to 20 minutes after the link failure trigger. Level 0 recovery actions are performed for the cellular interface. Repeats indefinitely, no more than every 10 minutes (default).
- Level 1 threshold is set to 90 minutes. Level 1 recovery actions are performed for the cellular interface. Repeats no more frequently than every 75 minutes.
- The Level 2 threshold is set to 600 minutes (10 hours).

The following sets the track-stub-object value to 21:

```
conf t
track 21 stub-object
event manager environment wanmon_if_list {cellular0/4/0 {ipsla 1} {stub 21}
```

Verifying WANMon Configuration

Use the following steps to verify your WANMon configuraion.

Procedure

	Command or Action	Purpose
Step 1	**show event manager policy registered**	Displays the WAN monitoring policy.
Step 2	**show event manager environment**	Displays the interface environment variables set during interface configuration.

What to do next

EXAMPLE

```
show event manager policy registered
1    script    system  multiple   Off   Thu Jan 16 18:44:29 2014  tm_wanmon.tcl

show event manager environment
1 wanmon_if_list   {cell0/4/0 {ipsla 1}}
```

Configuration Examples

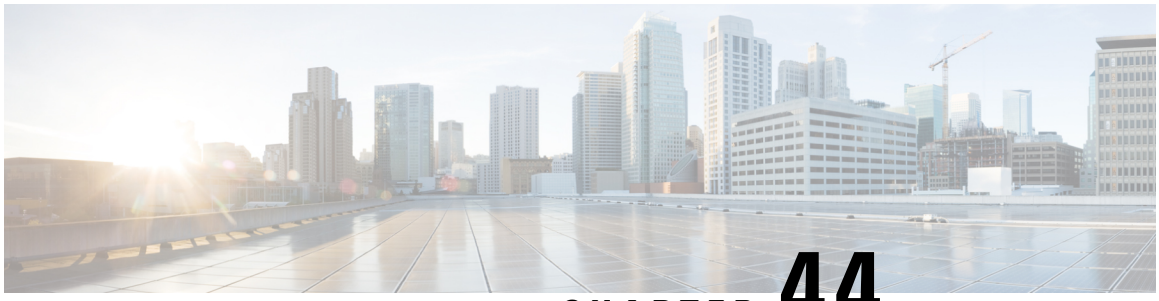
The following examples are provided:

WANMon Cellular Interface Configuration Example

```
track 1 ip sla 1
ip sla 1
 icmp-echo 172.27.166.250
 timeout 6000
 frequency 300
ip sla schedule 1 life forever start-time now
event manager environment wanmon_if_list {cellular0/4/0 {ipsla 1}}
event manager policy tm_wanmon.tcl authorization bypass
```

Multiple WAN Link Monitoring Example

```
track 1 ip sla 1
track 21 stub-object
ip sla 1
 icmp-echo 172.27.166.250
 timeout 6000
 frequency 300
ip sla schedule 1 life forever start-time now
track 2 ip sla 2
track 22 stub-object
ip sla 2
 icmp-echo 10.27.16.25
 timeout 6000
 frequency 300
ip sla schedule 2 life forever start-time now
event manager environment wanmon_if_list1 {cellular0/4/0 {ipsla 1} {stub 21}}
event manager policy tm_wanmon.tcl authorization bypass
```



CHAPTER 44

Yang Data Models

This chapter contains the following:

- Support for YANG Data Models (Call-home), on page 419
- Yang Data Model Support for Raw Socket Transport, on page 419
- Yang Data Model Support for Scada, on page 420

Support for YANG Data Models (Call-home)

The YANG models supported for the call-home feature are similar to the earlier releases of Cisco-IOS-XE, and the same is supported on the release of IOS-XE on IR1800. The following references are available for earlier YANG models:

https://github.com/YangModels/yang/tree/master/vendor/cisco/xe For additional information about call-home for IOS-XE, see the following:

Software Activation Configuration Guide, Cisco IOS XE Release 3S

Yang Data Model Support for Raw Socket Transport

Release 17.2.1 adds support for additional Yang Data Models. These additional models include Raw Socket Transport.

Yang Data Models can be found here:

https://github.com/YangModels/yang/tree/master/vendor/cisco/xe/1721

There are two feature modules available for raw socket that belong to the main Cisco-IOS-XE-native model. They are:

- Cisco-IOS-XE-rawsocket.yang
- Cisco-IOS-XE-rawsocket-oper.yang

The Cisco-IOS-XE-rawsocket-oper.yang module contains a collection of YANG definitions for Raw Socket Transport Configuration commands.

The module has the following corresponding CLI commands:

```
# encapsulation raw-tcp
# encapsulation raw-udp
```

```
# raw-socket packet-length <length>
# raw-socket packet-timer <timer>
# raw-socket special-char <value>
# raw-socket tcp server <port> <ip>
# raw-socket tcp idle-timeout <value>
# raw-socket tcp client <dest-ip> <dest-port>
# raw-socket tcp idle-timeout <timeout>
# raw-socket tcp tcp-session <value>
# raw-socket tcp dscp <value>
# raw-socket udp connection <dest-ip> <dest-port> <local_port>
```

The Cisco-IOS-XE-rawsocket-oper.yang module contains a collection of YANG definitions for Raw Socket Transport operational data.

The module has the following corresponding CLI commands:

```
# show raw udp statistics
# show raw tcp statistics
# show raw tcp session
# show raw udp session
# show raw tcp session local
# show raw udp session local
```

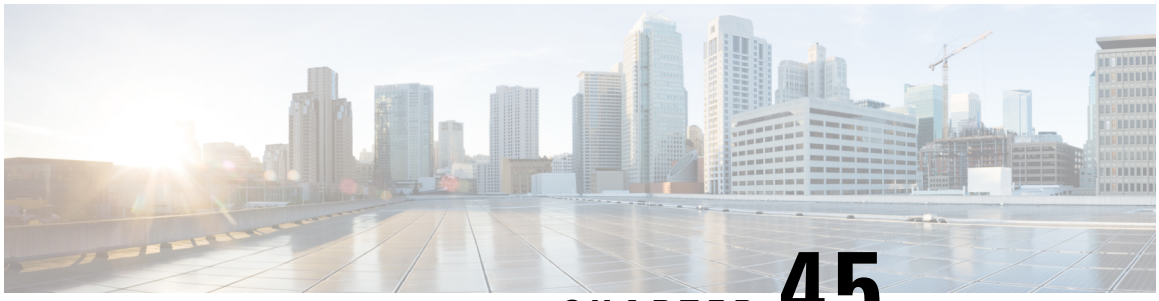
The following is a list of the Dependent Modules:

- Cisco-IOS-XE-native
- Cisco-IOS-XE-features
- ietf-inet-types
- Cisco-IOS-XE-interfaces
- Cisco-IOS-XE-ip
- Cisco-IOS-XE-vlan
- ietf-yang-types @ (any revision)
- cisco-semver

Yang Data Model Support for Scada

The Cisco IOS XE 17.1.1 release introduces support for the Cisco IOS XE YANG model for the Scada System. Previous releases already provided Yang models in other areas.

https://github.com/YangModels/yang/tree/master/vendor/cisco/xe/17111 .



CHAPTER 45

gRPC Network Operations

- gRPC Network Operations Interface updates, on page 421
- GNMI Broker update, on page 421

gRPC Network Operations Interface updates

A gRPC Network Operations Interface (gNOI) update is a gRPC-based microservice that:

- defines interfaces for executing operational commands and procedures on network devices, such as OS Install, Activate, and Verification,
- enables operating system related tasks through os.proto, including OS activation, install, detailed overview, internal OS commands, and summary output of OS operations, and
- provides capabilities to display gnmib detailed state, check gnmib operational statistics, and output modifiers.

GNMI Broker update

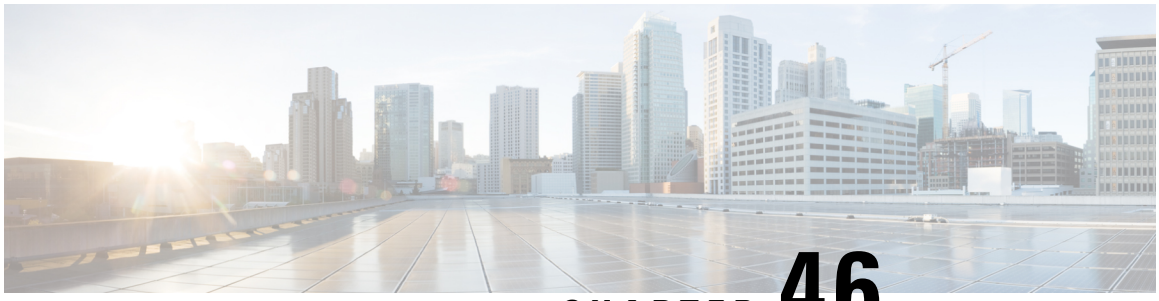
A GNMI Broker update is a software enhancement that:

- extends GNMIB to support the gRPC Network Operations Interface (gNOI) reset.proto service,
- provides functionality for restoring the device to its factory defaults via gRPC, and,
- maintains the current booted image through the reset process.

Service behavior and functionality

When the service is executed, it behaves similarly to the 'factory-reset all' command, and subsequently triggering a reload. Additionally, the service will maintain the current booted image. These additional steps will be taken to comply with the reset.proto service:

- Set the rommon BOOT variable to the current booted image and maintain it through reload following factory-reset,
- Enable autoboot to bring the device up on the current booted image following factory-reset.



CHAPTER 46

Cellular Network Slicing

This chapter contains the following sections:

- Cellular network slicing, on page 423
- Restrictions for cellular network slicing, on page 424
- Configure cellular network slicing, on page 424
- Example for configuring cellular network slicing , on page 425
- Verify cellular network slicing configuration, on page 425

Cellular network slicing

Network slicing refers to partitioning of a physical network into multiple virtual segments known as slices. This solution lets service providers divide a 5G physical network into several distinct virtual slices. A host platform can set up multiple simultaneous network connections, each with its own IP address. It uses slice types that match each host's requirements. This method enables service providers to allocate dedicated resources for different service types, ensuring optimal performance and efficiency.

5G network slicing is supported only on 5G standalone networks. It is not available on 5G non-standalone (NSA) networks. The slice or service types have a specific set of network functions and resources defined by the network provider. Service type defines how a network slice is expected to behave, based on its features and services. These service types are supported in network slicing:

- Enhanced Mobile Broadband (eMBB): for high-bandwidth and low-latency applications.
- Ultra-reliable low latency communications (URLLC): for high reliability and low latency transmission requirements.
- Massive Internet of Things (MIoT): for sending small quantities of data.

Table 27: Feature History Table

Feature name	Release information	Feature description
Cellular Network Slicing	Release 17.18.2	This feature enables service providers to divide a 5G physical network into several distinct virtual slices.

Restrictions for cellular network slicing

These restrictions apply when configuring cellular network slicing.

- You can enable network slicing for the cellular profile only on the P-5GS6-R16SA-GL module when using 5G standalone mode.
- You cannot configure network slicing parameters on profile one or the default profile for the carrier.

Configure cellular network slicing

Follow these steps to configure cellular network slicing.

Procedure

Step 1 Use **config terminal** command to enter the global configuration mode.

Example:

```
Router# configure terminal
```

Step 2 Use the **controller cellular** <*slot*> command to enter the cellular controller configuration mode.

Example:

```
Router(config)# controller cellular x/x/x
```

Step 3 Use the **profile id** *number* **apn** *name* **authentication** *type* **pdn-type** *type* command to associate the profile with the PDN.

Example:

```
Router(config-controller)# profile id 3 apn apn-ns authentication none pdn-type ipv4
```

Step 4 Use the **profile id** *number* **apn** *name* **authentication** *type* **pdn-type** *type* **slice-type** *type* to associate the profile with the slice type.

Example:

```
Router(config-controller)# profile id 3 apn apn-ns authentication none pdn-type ipv4 slice-
 type miot
```

<1-3> slice type *number*: 1(eMBB), 2(URLLC), 3(MioT)

Step 5 Use the **profile id** *number* **apn** *name* **authentication** *type* **pdn-type** *type***slice-type** *type* **slice-differentiator** *number* to associate tslice differentiator with the slice type.

Example:

```
Router(config-controller)# profile id 3 apn apn-ns authentication none pdn-type ipv4 slice-
 type miot slice-differentiator 6
```

slice differentiator *number*: 0-16777214

If slice differentiator is bypassed, then "FF FF FF" is used internally as the slice differentiator to indicate that there is no slice differentiator associated with this slice type.

Example for configuring cellular network slicing

This example shows how to configure a cellular network slicing.

```
Device(config-controller)# profile id 3 apn apn-ns authentication none pdn-type ipv4
        no-overwrite
        slot            Associated sim slot
        slice-type Associated network slice type(SST)

        Device(config-controller)# profile id 3 apn apn-ns authentication none pdn-type
ipv4 slice-type
        <1-3>  Slice type number: 1(eMBB), 2(URLLC), 3(MioT)
        Device(config-controller)# profile id 3 apn apn-ns authentication none pdn-type
ipv4 slice- type embb
        slice-differentiator Associated Slice Differentiator(SD)  <--- if SD is bypassed
here,"FF FF FF" wll be used internally as the SD to indicate that there is no SD associated
 with this SST.
        slot                            Associated sim slot

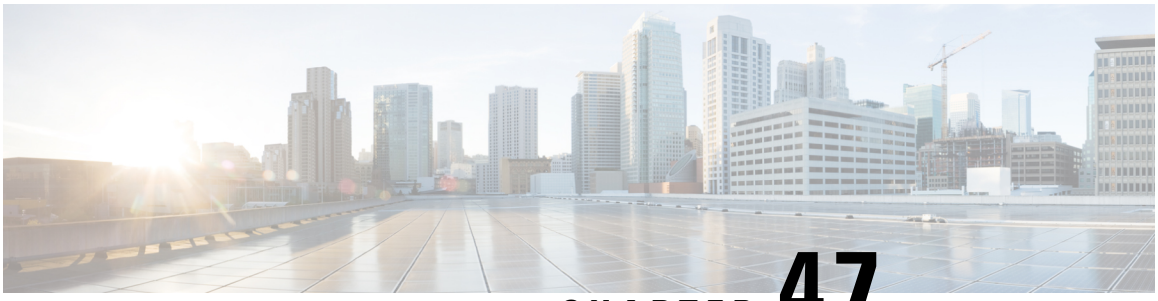
        Device(config-controller)# profile id 3 apn apn-ns authentication none pdn-type
ipv4 slice-type
        embb slice-differentiator
        <0-16777214>  Slice Differentiator
        Device(config-controller)#
        Device(config-controller)#$ pdn ipv6  slice-type miot slice-differentiator 6
        slot  Associated sim slot
```

Verify cellular network slicing configuration

This is a sample output from the **show cellular** command that displays network slicing configuration and slice type details.

```
Device# show cellular profile 3

                Profile 3 = ACTIVE
                --------
                PDP Type = IPv6
                Access Point Name (APN) = TestNSSAI6_MIOT1
                Authentication = None
                S-NSSAI Slice Type is MIOT
                S-NSSAI Slice Differentiator = 6
```



CHAPTER 47

Change of Authorization

This chapter contains the following sections:

- Change of authorization, on page 427

Change of authorization

A Change of authorization (CoA) is a network policy mechanism to change the attributes of an authentication, authorization, and accounting (AAA) session. When a policy changes for a user or user group in AAA, administrators can send the CoA packets from the AAA server such as a Cisco Secure Access Control Server (ACS) to reinitialize authentication and apply the new policy.

CoA also helps in:

- Supporting actions such as session query, reauthentication, termination, port bounce, and port shutdown.
- Enabling dynamic activation or deactivation of service templates.

Change of authorization is part of Identity-Based Networking Services and helps enforce policy changes in real time. This feature allows administrators to respond to changes in user roles, device states, or network conditions.

Table 28: Feature History Table

Feature name	Release information	Feature description
CoA Support	Release 26.1.1	This feature supports Change of Authorization (CoA), which is a network policy mechanism that: • Modifies session attributes for active authentication, authorization, and accounting sessions. actions such as session query and reauthentication termination, port bounce, and port shutdown. • Enables dynamic activation or deactivation of service templates ass part of Identity-Based Networking Services. Change of Authorization enforces policy changes in real time, allowing administrators to respond promptly to changes in user roles, device states, or network conditions.

Information about change of authorization

How change of authorization reauthentication work

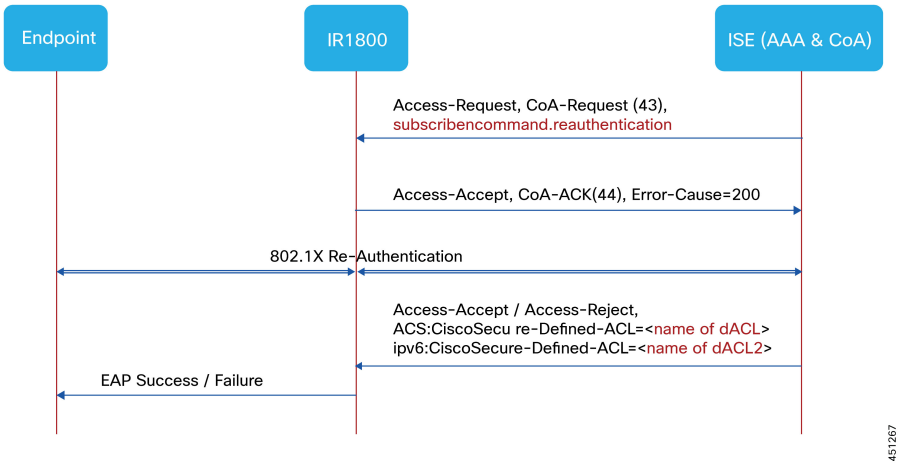
Summary

Change of authorization reauthentication enables dynamic policy changes in AAA sessions after initial authentication.

- When a policy changes for a user or user group in AAA, administrators can send RADIUS CoA packets from the AAA server.
- The AAA server, such as Cisco Identity Services Engine, uses CoA packets to reinitialize authentication and apply the new policy.
- The RADIUS interface provides various primitives that can be used during a CoA event.
- These primitives and their functions are essential for effectively applying new policies to users or groups during a session.

Workflow

Figure 34: Workflow



1. The administrator changes a user or user group policy in the AAA system.
2. The AAA server sends a RADIUS CoA packet to the network device, specifying policy updates.
3. The device receives the CoA packet and reinitializes authentication, applying the new policy.
4. The RADIUS interface returns either a CoA-ACK (acknowledgement) or CoA-NAK (nonacknowledgement) as a response.

Result

By default, the RADIUS interface is enabled on the device. However, some basic configuration is required for the following attributes:

- Security and Password
- Accounting
- CoA acknowledgement (ACK) [CoA-ACK]
- CoA nonacknowledgement (NAK) [CoA-NAK]

What's next

After posture assessment is succeessful, full network access is pushed down to the device for specific client through CoA re-authentication command based on its compliance state derived from last assessment. It is optional to enforce downloadable ACLs with Permit-ALL or limited access to certain resources to corresponding clients. Per-session CoA requests are supported for session identification, session termination, host reauthentication, port shutdown, and port bounce. This model comprises one request (CoA-Request) and two possible response codes:

Change of authorization requests

A Change of authorization is a network policy mechanism that:

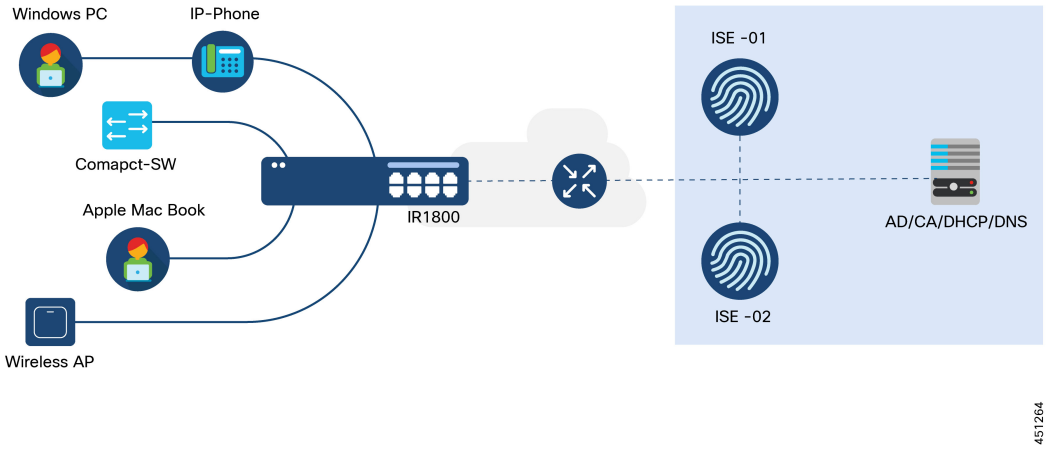
- facilitates endpoint re-authentication based on posture assessment,

- integrates with Cisco AnyConnect and Cisco ISE, version 2.6 and
- enhances security through customizable security policies per client.

Topology diagram of Cisco IR1800 Router as a branch router

The network topology below shows a typical Cisco IR1800 Series Router as a branch router in a network for secure access with ISE and other network services deployed in Campus or Data Center. CoA is critical part of the solution to initiate re-authentication or re-authorization to endpoint's network access based on its posture assessment result. Downloadable ACL is the target or purpose of the entire solution.

Figure 35: Network topology of Cisco IR1800 with ISE and other Network Services



Limitations for change of authorization

When configuring the Change of Authorization feature, ensure that you observe these restrictions:

- Only platforms with SKUs that support TCAM can use downloadable ACLs (DACL) and redirect ACLs with change of authorization.
- The switch ASIC TCAM supports a maximum of 255 IPv4 ACL entries in total.
- Multi-Auth VLAN is not supported with change of authorization.
- VLAN changes may not occur consistently with multiple iterations on the same client interface.

How to configure change of authorization

Create RADIUS server group

Use this procedure to create a RADIUS server group and add a RADIUS server to the group.

Procedure

Step 1 Use the **configure terminal** command to enter global configuration mode.

Example:

```
Device# configure terminal
```

Step 2 Use the **radius server** *server-name* command to configure a RADIUS server.

Example:

```
Device(config)# radius server ise
```

Step 3 Use the **address ipv4** *ip* **auth-port** *port* **acct-port** *port* command to set the RADIUS server's IP address and authentication/accounting ports.

Example:

```
Device(config-radius-server)# address ipv4 10.0.0.10 auth-port 1812 acct-port 1813
```

Step 4 Use the **aaa group server radius** *radius-group* **server name** *RADIUS-server-name* command to create a RADIUS server group and add the RADIUS server to it.

Example:

```
Device(config-radius-server)# aaa group server radius ise-group
 server name ise
```

Step 5 Use the **key** <*secret-key*> command to specify the shared secret key (encrypted) for the RADIUS server.

Example:

```
Device(config-radius-server)# key secret-key
```

What to do next

Configure 802.1X authentication using a RADIUS server

Configure 802.1X authentication using a RADIUS server

Follow this procedure to enable and set up 802.1X authentication using a RADIUS server on a Cisco device.

Before you begin

- Ensure that your device is in global configuration mode.
- Ensure that the RADIUS server is reachable from the device and is configured to accept requests.
- Ensure that a RADIUS server group is already created.

Procedure

Step 1 Use the **dot1x system-auth-control** command to enable 802.1X authentication on the router.

Example:

```
Device(config)# dot1x system-auth-control
```

Step 2 Use the **aaa new-model** command to enable authentication, authorization, and accounting (AAA) globally.

Example:

```
Device(config)# aaa new-model
```

Step 3 Use the **aaa authentication dot1x default group** *radius-server-group-name* command to specify that 802.1X authentication requests use the RADIUS server group by default.

Example:

```
Device(config)# aaa authentication dot1x default group ise-group
```

Step 4 Use the **aaa authorization network default group** *radius-server-group-name* command to specify that 802.1X authorization requests use the RADIUS server group by default.

Example:

```
Device(config)# aaa authorization network default group ise-group
```

Step 5 Use the **aaa authorization auth-proxy default group** *radius-server-group-name* command to specify that the authorization of the authentication proxy requests use RADIUS server group by default, enabling dynamic user access control via the specified RADIUS server group.

Example:

```
Device(config)# aaa authorization auth-proxy default group ise-group
```

Step 6 Use the **aaa accounting auth-proxy default start-stop group** *radius-server-group-name* command to specify that the accounting of the authentication proxy requests use RADIUS server group by default, enabling dynamic user access control via the specified RADIUS server group.

Example:

```
Device(config)# aaa accounting auth-proxy default start-stop group ise-group
```

This command enables the device to generate accounting records when an authentication proxy cache is created and deleted, capturing user session information dynamically via the specified RADIUS server group.

Step 7 Use the **aaa accounting dot1x default start-stop group** *radius-server-group-name* command to enable accounting for 802.1X authentication sessions.

Example:

```
Device(config)# aaa accounting dot1x default start-stop group ise-group
```

This command sends both start and stop accounting records to the specified RADIUS server group, allowing tracking of user authentication sessions on the network.

Step 8 Use the **key** <*secret-key*> command to specify the shared secret key (encrypted) for the RADIUS server.

Example:

```
Device(config-radius-server)# key secret-key
```

Configure 802.1X authentication on an access port

Follow this procedure to configure a switch interface for 802.1X authentication in single-host mode with port security.

Before you begin

- Ensure that your device is in global configuration mode.
- Ensure that global AAA and 802.1X configurations are completed prior to configuring the interface.

Procedure

Step 1 Use the **interface** *interface-type port* command to enter the interface configuration mode for the specified interface.

Example:

```
Device(config)# interface GigabitEthernet0/1/0
```

Step 2 Use the **switchport mode access** command to set the port to access mode for end devices.

Example:

```
Device(config-if)# switchport mode access
```

Step 3 Use the **switchport access vlan** *vlan-id* command to assign the specified VLAN as the access port.

Example:

```
Device(config-if)# switchport access vlan 100
```

Step 4 Use the **authentication order dot1x mab** command to specify the order in which authentication methods are attempted.

Example:

```
Device(config-if)# authentication order dot1x mab
```

In this case, the switch first tries 802.1X (dot1x) authentication, and if that fails or is not supported by the device, it falls back to MAC Authentication Bypass (MAB).

Step 5 Use the **authentication priority dot1x mab** command to set the priority order of authentication methods on a port.

Example:

```
Device(config-if)# authentication priority dot1x mab
```

In this case, the switch first configures the port to prioritize dot1x authentication first, and if that method is interrupted or fails, it will then attempt MAB.

Step 6 Use the **authentication port-control auto** command to require dot1x authentication before granting access.

Example:

```
Device(config-if)# authentication port-control auto
```

Step 7 Use the **mab** command to enable MAC-based authentication on the interface.

Example:

```
Device(config-if)# mab
```

Step 8 Use the **dot1x pae authenticator** command to enable 802.1X authentication in authenticator mode.

Example:

```
Device(config-if)# dot1x pae authenticator
```

Step 9 Use the **access-session host-mode single-host** command to limit the port to one authenticated device at a time.

Example:

```
Device(config-if)# access-session host-mode single-host
```

Step 10 Use the **description** *text* command to add a descriptive label to the interface configuration.

Example:

```
Device(config-if)# description 802.1X Test Port
```

Configure RADIUS change of authorization

Use this procedure to configure RADIUS change of authorization.

Procedure

Step 1 Use the **configure terminal** command to enter global configuration mode.

Example:

```
Device# configure terminal
```

Step 2 Use the **aaa new-model** command to enable authentication, authorization, and accounting (AAA) globally.

Example:

```
Device(config)# aaa new-model
```

Step 3 Use the **aaa server radius dynamic-author** command to enter dynamic authorization local server configuration mode and specify a RADIUS client from which a device accepts Change of Authorization (CoA) and disconnect requests.

Example:

```
Device(config)# aaa server radius dynamic-author
```

This command configures the device as a AAA server to facilitate interaction with an external policy server.

Step 4 Use the **client** { *ip-address* | *name* [**vrf** *vrf-name*]} **server-key** *string* command to configure the RADIUS key to be shared between a device and RADIUS clients.

Example:

```
Device(config-locsvr-da-radius)# client 10.0.0.1 server-key cisco123
```

Step 5 (Optional) Use the **port** *port-number* command to specify the port on which a device listens for RADIUS requests from configured RADIUS clients.

Example:

```
Device(config-locsvr-da-radius)# port 3799
```

Note

The default port for packet of disconnect is 1700.

Configuration examples for change of authorization

RADIUS server status example

This example shows how to check if the RADIUS server is active.

```
Device# show aaa servers
RADIUS: id 1, priority 1, host 10.75.28.231, auth-port 1812, acct-port 1813, hostname host
State: current UP
duration 188755s, previous duration 0s
Dead: total time 0s, count 0
Platform State from SMD: current UP, duration 188755s, previous duration 0s
```

Verification of change of authorization

Successful 802.1X authentication and CoA reauthentication

Verify successful 802.1X authentication and CoA reauthentication

```
Device# show access-session interface GigabitEthernet0/1/0 details
Interface:  GigabitEthernet0/1/0
MAC Address:  00a1.b2c3.d4e5
IPv4 Address: 10.10.10.25
User-Name:  host_8021x_client
Status:  Authorized
Domain:  DATA
Oper host mode:  single-host
Oper control dir:  both
Session timeout:  N/A
Idle timeout:  N/A
Common Session ID:  0A01000A00000012A4E3D6B7
Acct Session ID:  0x00000012
Handle:  0xB4000012
Current Policy:  802.1X_AUTH_POLICY

Local Policies:
Service Template:  DOT1X-DEFAULT (activated)

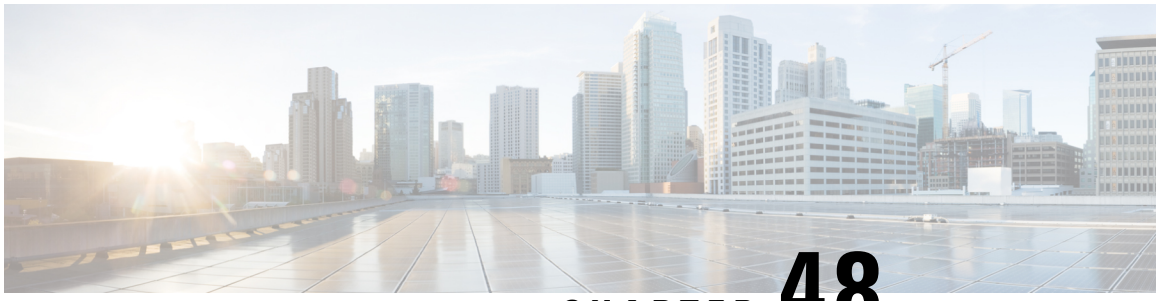
Server Policies:
VLAN Group:  VLAN_10
ACL:  ACL_PERMIT_ALL
URL Redirect ACL:  N/A
```

Troubleshoot RADIUS change of authorization configuration

Use these comands to monitor and troubleshoot the RADIUS Change of Authorization feature.

Table 29: Monitoring and Troubleshooting RADIUS Change of Authorization

Command	Purpose
debug aaa coa	Displays debug information for CoA processing.
debug aaa authentication	Displays debug information related to AAA (Authentication, Authorization, and Accounting) authentication processes. It helps to learn the methods of authentication being used and troubleshoot authentication issues by showing detailed logs of authentication events as they occur.
show aaa attributes protocol radius	Displays the mapping between an authentication, authorization, and accounting (AAA) attribute number and the corresponding AAA attribute name.



CHAPTER 48

Password encryption types for wireless profile configuration

Table 30: Feature History Table

Feature name	Release information	Feature description
Password encryption types for wireless profile configuration	Release 26.2.1	Starting from Cisco IOS XE Release 26.2.1, credentials entered as Type 0 in supported wireless profile configurations are automatically converted to and stored as Type 6 encrypted values. Type 7 credentials are deprecated and are rejected. This behavior applies to WLAN profile passphrases and management credentials in Embedded Wireless Controller mode and to SSID profile secret keys in Workgroup Bridge mode.

- Password encryption types for wireless profile configuration, on page 437
- Prerequisites, on page 438
- Limitations, on page 438
- Configure wireless module in EWC mode, on page 439
- Configure wireless module in WGB mode, on page 439
- Verify EWC autonomous and controller mode configurations, on page 440
- Verify WGB autonomous and controller mode configurations, on page 440

Password encryption types for wireless profile configuration

Starting from the Cisco IOS XE 26.2.1 release, less secure password encryption types such as Type 7 are deprecated from the Cisco Catalyst IR1800 router Wi-Fi modules as part of infrastructure resilience and security compliance. Type 0 passphrases and keys are automatically converted to Type 6 to enhance security.

This feature applies to:

- The WLAN profile passphrase and **wireless-lan** management credential in Embedded Wireless Controller (EWC) mode.
- The **wireless-bridge** SSID profile configuration in Workgroup Bridge (WGB) mode.

EWC mode

EWC mode: WLAN passphrase and management credential CLI auto converts Type 0 to Type 6 in both the controller and autonomous modes. Type 7 encryption is deprecated.

WGB mode

WGB mode: Wireless bridge SSID profile configuration auto converts Type 0 to Type 6 in both controller and autonomous modes. Type 7 encryption is deprecated.

Prerequisites

Consider these points before using password encryption types in wireless profile configuration:

- The IR1800 router must be running Cisco IOS XE 26.2.1 or later.
- A supported Wi-Fi module must be installed on the IR1800.
- Before converting Type 0 encryption to Type 6, you must ensure that the master key is configured for Advanced Encryption Standard (AES) encryption.



Tip You can use the **show running-config** | **section aes** command to verify whether AES encryption is enabled.

Limitations

The limitations of using password encryption types for wireless profiles are:

- Type 7 configuration is deprecated. You can no longer push or configure a Type 7 passphrase or management credential; the attempt is rejected.
- In CLI templates or CLI configuration groups, any backslash '\' character in a Type 6 password must be doubled entered as \ before deployment. Cisco SD-WAN Manager trims a single \ character, so entering it as \\ ensures the device can correctly decrypt the password.



Note Regardless of the input encryption option provided (Type 0 or Type 6), the credential is stored only as Type 6.

Configure wireless module in EWC mode

Use this task to configure the wireless module in EWC mode.

Procedure

Step 1 Use the **configure terminal** command to enter global configuration mode.

Example:

```
Router# configure terminal
```

Step 2 Use the **wireless-lan mgmt credential username** *username* **password** *type password* command to configure the management credential for the wireless LAN in EWC mode.

Example:

```
Router(config)# wireless-lan mgmt credential username <user-name> password ?
0 Specifies an UNENCRYPTED password will follow
6 Specifies an ENCRYPTED password will follow
LINE The UNENCRYPTED (cleartext) user password

Router(config)# wireless-lan mgmt credential username <user-name> password 0 ?
LINE The UNENCRYPTED (cleartext) user password
Router(config)#$n mgmt credential username admin password 0 admin ?
LINE <cr> <cr>
```

Step 3 Use the **wlan-profile** *profile-name* command to enter WLAN profile mode.

Example:

```
Router(config)# wlan-profile wlan
```

Step 4 Use the **passphrase** *type passphrase* command to configure the WLAN profile passphrase.

Example:

```
Router(config-wlan-profile)# passphrase 0 <passphrase>
```

Configure wireless module in WGB mode

Use this task to configure the wireless module in WGB mode.

Procedure

Step 1 Use the **configure terminal** command to enter global configuration mode.

Example:

```
Router# configure terminal
```

Step 2 Use the **wireless-bridge** command to enter the wireless bridge configuration mode in WGB mode.

Example:

```
Router(config)# wireless-bridge
```

Step 3 Use the **ssid-profile** *ssid-profile-name* **ssid** *ssid-name* **authentication psk key-management wpa2 secret-key** *type secret-key* command to configure an SSID profile in the WGB mode.

Example:

```
Router(config-wl-bridge)# ssid-profile test-ssid-profile ssid test-ssid authentication psk
 key-management wpa2 secret-key 0 testkey123!
```

Verify EWC autonomous and controller mode configurations

Use this task to check EWC autonomous and controller mode configurations on IR1800 router.

Procedure

Step 1 Use the **show sd-routing running-config | sec wlan-profile** command to verify EWC autonomous mode running configuration after the configuration push.

Example:

```
Router# show sd-routing running-config | sec wlan-profile
wlan-profile sample_aab
ssid          sample_aab
radio-band    all
vlan-id       200
data-security personal
passphrase 6 BKhQC_Cbc]PeLOb^JNFcHMdhDa^PZbZKRICg
qos-type      silver
```

Step 2 Use the **show sdwan running-config | sec wlan-profile** command to verify EWC controller mode running configuration after the configuration push.

Example:

```
Router# show sdwan running-config | sec wlan-profile
wlan-profile sample_aab
ssid         sample_aab
radio-band   all
vlan-id      200
data-security personal
passphrase 6 BKhQC_Cbc]PeLOb^JNFcHMdhDa^PZbZKRICg
qos-type     silver
```

Verify WGB autonomous and controller mode configurations

Use this task to check WGB autonomous and controller mode configurations on IR1800 router.

Procedure

Step 1 Use the **show sd-routing running-config | sec wireless-bridge** command to verify WGB autonomous mode running configuration after the configuration push.

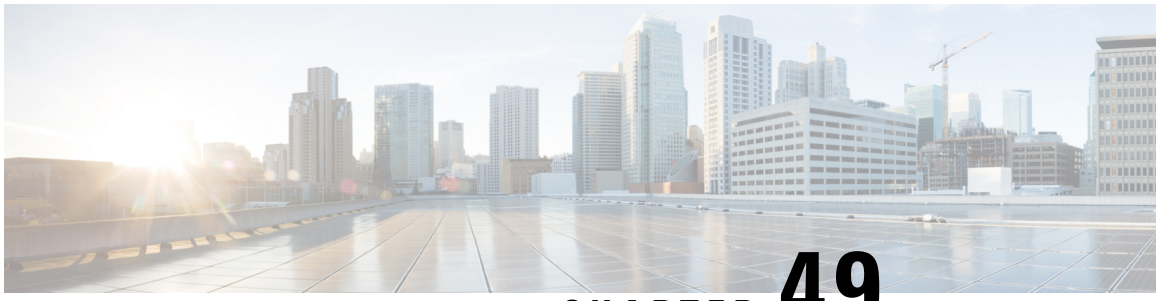
Example:

```
Router# show sd-routing running-config | sec wireless-bridge
wireless-bridge
qos-profile qos1 platinum
ssid-profile sp1 ssid s1 authentication open
ssid-profile sp2 ssid s2 qos-profile qos1 authentication psk key-management wpa2 secret-key
 6 NKTYYi_UHLNWUcKHHCCY_Ni\\CKJO\ShVJ`TWZ`agfRO^AAB
ssid-profile sp3 ssid s3 authentication psk key-management wpa2 secret-key 6
DECURVJaKSPbHF_[FiUgeHFKPhYHFbOi]ASFDbE
ssid-profile sp4 ssid s4 authentication psk key-management wpa2 secret-key 6
gdNdP\SVfbRgRAKOJXZL]XgZWLMZJLVDIUQ]
ssid-profile sp5 ssid s5 authentication psk key-management wpa2 secret-key 6
GXQKSUJ^L_A_dR^RXbTOL[BDb[LeJFYEgVgMDBRE^GFdaAAB
ssid-profile sp6 ssid s6 authentication psk key-management wpa2 secret-key 6
\\McXcUFU[C[\bNWZaJKK[MVRHNQKBBXXHGT
ssid-profile sp7 ssid s7 authentication psk key-management wpa2 secret-key 6
IGhVBh_WOAfKPf_OVTOQAdSHcT^RTa]S\RCN
ssid-profile sp8 ssid s8 authentication psk key-management wpa2 secret-key 6
KX`[E]N[JJEM\LdOcC`FL^ePePZC[eEfQdaN
ssid-profile ssidp1 ssid s1 authentication psk key-management wpa2 secret-key 6
KeeVe`Z_K]_TLIddbfSg_PM]MIPSPafiXJLe
ssid-profile testssid2 ssid ts2 authentication psk key-management wpa2 secret-key 6
IGhVBh_WOAfKPf_OVTOQAdSHcT^RTa]S\RCN
```

Step 2 Use the **show sdwan running-config | sec ssid-profile** command to verify WGB controller mode running configuration after the configuration push.

Example:

```
Router# show sdwan running-config | sec ssid-profile
ssid-profile wlan2 ssid 000_aab_pwdauth qos-profile qos-profile1 authentication psk
key-management wpa2 secret-key 6 bJUWfbBePBSXUFAKOIZDTLUdEgQeOhZKMR\M
ssid-profile wlan3 ssid 000_bba qos-profile qos-profile1 authentication open
```



CHAPTER 49

IKEv2 and IPsec PFS Post Quantum Cryptography

Table 31: Feature History Table

Feature name	Release information	Feature description
IKEv2 and IPsec PFS Post Quantum Cryptography	Release 26.2.1	Cisco IoT routers can combine a traditional Diffie-Hellman key exchange with the NIST-standardized Module-Lattice-Based Key-Encapsulation Mechanism (ML-KEM) to establish keys for IKEv2/IPsec sessions. This hybrid approach helps protect recorded VPN traffic against future harvest-now, decrypt-later attacks.

- IKEv2 and IPsec PFS Post Quantum Cryptography, on page 443
- Benefits of PQC, on page 444
- Prerequisites, on page 444
- Limitations, on page 444
- ML-KEM algorithm and parameter sets, on page 445
- PFS negotiation behavior, on page 446
- Configure an IKEv2 proposal for PQC, on page 446
- Configure Perfect Forward Secrecy with PQC, on page 448
- Verify the PQC hybrid key exchange, on page 449
- Troubleshooting PQC hybrid key exchange, on page 450

IKEv2 and IPsec PFS Post Quantum Cryptography

Quantum-Safe Encryption refers to cryptographic techniques designed to protect network communications against attacks by computers and future cryptographically relevant quantum computers.

Post-Quantum Cryptography (PQC) is part of this broader security approach. PQC includes quantum-resistant algorithms for functions such as key establishment and digital signatures. These algorithms help mitigate harvest-now, decrypt-later attacks, in which an attacker records encrypted traffic and attempts to decrypt it later using a sufficiently capable quantum computer.

PQC does not replace existing public-key algorithms. Instead, it works alongside them to create a quantum-safe hybrid key exchange.

The Module-Lattice-Based Key-Encapsulation Mechanism (ML-KEM) is a National Institute of Standards and Technology (NIST)-standardized post-quantum key-encapsulation mechanism. ML-KEM enables two peers to establish a shared secret that can be used to derive cryptographic session keys.

On supported Cisco routers, ML-KEM is combined with a traditional key-exchange algorithm to provide a hybrid key establishment for IKEv2/IPsec Perfect Forward Secrecy (PFS). The hybrid approach protects the negotiated session as long as at least one of the constituent key-establishment mechanisms remains secure.

For more information on PQC on Cisco routers with IKEv2 sessions, see Security and VPN Configuration Guide.

For more information on Quantum-Safe Encryption and the related protocols, see: https://www.cisco.com/c/en/us/td/docs/routers/ios/config/17-x/sec-vpn/b-security-vpn/m-sec-cfg-quantum-encryption-ppk.html.



Note Postquantum preshared keys (PPKs), including dynamic PPKs obtained through the Secure Key Integration Protocol (SKIP), can be used with ML-KEM-based hybrid key establishment.

Benefits of PQC

You can use the PQC feature to strengthen the security of your IPsec and IKEv2 connections with quantum-safe, future-proof protection for traffic between Cisco IoT routers.

The PQC hybrid key exchange provides these benefits:

- Future-proof security: Protects traffic against future quantum-computer-based attacks.
- Defense in depth: Combines a classical DH algorithm with ML-KEM, so the session stays secure even if one algorithm is broken.
- Smooth migration: Supports phased migration in existing networks through the optional fallback keyword.

Prerequisites

Before you configure the PQC hybrid key exchange, ensure that the following requirements are met:

- Both peers support PQC, or the optional keyword is used to allow fallback during migration.
- IKEv2/IPsec is configured and operational between the peers.

Limitations

The limitations with respect to using PQC are:

- IKEv2 always uses hybrid key exchange: a combination of Diffie-Hellman (DH) and ML-KEM algorithms. Pure-PQC key exchange is not supported for IKEv2.

- The encapsulation keys and ciphertexts used by ML-KEM-768 and ML-KEM-1024 may cause IKEv2 messages to exceed typical path maximum transmission units (MTUs), resulting in IP fragmentation and potential packet drops. To avoid IP fragmentation, enable IKEv2 fragmentation and set the IKEv2 fragmentation MTU to 1400 bytes.
- This feature provides post-quantum key establishment only. It does not provide post-quantum peer authentication. IKEv2 peers continue to use the configured authentication method, such as a preshared key or certificate-based authentication.
- ML-KEM establishes shared keying material; it does not encrypt user traffic. The encryption algorithm configured in the IPsec transform set, such as AES-CBC or AES-GCM, continues to encrypt the user traffic.

Note License, platform, and software support: This feature does not require an additional feature license. ML-KEM-based quantum-safe key establishment is supported on all Cisco IoT routers with Cisco IOS XE Release 26.2.1.

Supported routers

The supported routers are:

- Cisco Catalyst IR1101 Rugged Series Router,
- Cisco Catalyst IR1800 Rugged Series Router,
- Cisco Catalyst IR8340 Rugged Series Router,
- Cisco Catalyst IR8140 Heavy Duty Router, and
- Cisco ESR6300 Embedded Series Router.

Phased migration in Hub-and-Spoke deployments

If an older responder (hub) receives a PQC proposal from upgraded spokes (initiators), the negotiation fails.

To avoid this, it is recommended that you always upgrade the hub first and configure **optional** in the hub's IKEv2 proposal.

When any issue occurs, the responder logs the following errors:

IKEv2-INTERNAL: Transform type 6 invalid.

IKEv2-ERROR: Failed to parse the packet: The peer's proposal is invalid.

ML-KEM algorithm and parameter sets

Cisco IoT routers support the three National Institute of Standards and Technology (NIST)-standardized ML-KEM parameter sets: ML-KEM-512, ML-KEM-768, and ML-KEM-1024. During the key exchange, the peers exchange an encapsulation key (public key) and a ciphertext.

Table 32: ML-KEM parameter sets (sizes in bytes)

Parameter	ML-KEM-512	ML-KEM-768	ML-KEM-1024
Encapsulation key size (public key)	800	1184	1568
Decapsulation key size (private key)	1632	2400	3168
Shared secret key size	32	32	32
Ciphertext size	768	1088	1568

PFS negotiation behavior

Table 33: PFS negotiation behavior for Cisco IOS XE to Cisco IOS XE devices

Initiator/ Responder	no set pfs	set pfs	set pfs <group>	set pfs <group> pqc <mlkem>
set pfs	UP (ML-KEM inherit)	UP (ML-KEM inherit)	UP (DH+ML-KEM inherit)	UP (DH+ML-KEM inherit)
set pfs <*group*>	UP (DH)	DOWN	UP (DH)	DOWN
set pfs <*group*> **pqc** <*mlkem*>	UP (DH+ML-KEM)	UP (DH+ML-KEM inherit)	UP (DH+ML-KEM)	UP (DH+ML-KEM)

Note During initial session establishment, `CHILD_SA` output might display `PFS: N`. After the `CHILD_SA` rekeys, verify the negotiated ML-KEM and DH PFS settings.

Configure an IKEv2 proposal for PQC

Follow these steps to enable ML-KEM algorithm in an IKEv2 proposal.

Procedure

Step 1 Use the **configure terminal** command to enter global configuration mode.

Example:

```
Router# configure terminal
```

Step 2 Use the **crypto ikev2 fragmentation mtu** *mtu-size* command to enable IKEv2 fragmentation and set the IKEv2 fragmentation MTU to 1400 bytes.

Example:

```
Router(config)# crypto ikev2 fragmentation mtu 1400
```

Step 3 Use the **crypto ikev2 proposal** *name* command to create or edit an IKEv2 proposal.

Example:

```
Router(config)# crypto ikev2 proposal prop1
```

Step 4 Use the **encryption** *aes-cbc-key* command to configure AES key in CBC mode.

Example:

```
Router(config-ikev2-proposal)# encryption aes-cbc-256
```

a) Use the **integrity** *integrity-algorithm* command to configure the IKEv2 integrity algorithm.

Example:

```
Router(config-ikev2-proposal)# integrity sha512
```

b) Use the **group** *group-number* command to configure Diffie-Hellman group for classical key exchange used with ML-KEM.

Example:

```
Router(config-ikev2-proposal)# group 19
```

Step 5 Use the **pqc mlkem1024 mlkem768 mlkem512 optional** command to enable one or more ML-KEM algorithm parameter sets.

Example:

```
Router(config-ikev2-proposal)# pqc mlkem1024 mlkem768 mlkem512 optional
```

Note
The proposal enables **pqc mlkem768 optional** by default. The **optional** keyword allows the SA to fall back to a classical (non-PQC) negotiation if the peer does not support PQC, which supports coexistence during phased migration.

Step 6 Use these commands to create the IKEv2 policy and attach the proposal:

a) Use the **crypto ikev2 policy** *policy* command to create the IKEv2 policy.

Example:

```
Router(config)# crypto ikev2 policy pol1
```

b) Use the **match fvrf any** command to apply the IKEv2 policy to negotiations through any front-door VRF (FVRF).

Example:

```
Router(config-ikev2-policy)# match fvrf any
```

c) Use the **proposal** *proposal-name* command to add the PQC-enabled proposal to the IKEv2 policy.

Example:

```
Router(config-ikev2-policy)# proposal prop1
```

The **proposal** *proposal* command makes the PQC-enabled proposal available during IKEv2 negotiation for sessions that match this policy.

Step 7 Use these commands to associate the existing IKEv2 profile with the IPsec profile:

a) Use the **crypto ipsec profile** *profile-name* command to create an IPsec profile.

Example:

```
Router(config)# crypto ipsec profile gen
```

b) Use the **set ikev2-profile** *ikev2-profile-name* command to associate an existing IKEv2 profile with the IPsec profile.

Example:

```
Router(ipsec-profile)# set ikev2-profile prof1
```

Note
The profile and IPsec transform set can already exist, if not you can create it using the create IPsec profile option given in step 7.

Step 8 Use these commands to apply the IPsec profile to the tunnel:

a) Use the **interface tunnel** *tunnel-interface-number* command to create a tunnel interface.

Example:

```
Router(config)# interface tunnel 1
```

b) Use the **tunnel protection ipsec profile** *ipsec-profile-name* command to protect the tunnel with IPsec profile.

Example:

```
Router(config-if)# tunnel protection ipsec profile gen
```

Step 9 Use the **end** command to return to privileged EXEC mode.

Example:

```
Router(config-if)# end
```

Configure Perfect Forward Secrecy with PQC

Perfect Forward Secrecy (PFS) adds extra randomness to derive session keys during each key exchange. With PQC, the key exchange can use DH, ML-KEM, or DH + ML-KEM. PFS is optional and can be disabled.

Follow these steps to configure PFS with PQC.

Procedure

Step 1 Use the **configure terminal** command to enter global configuration mode.

Example:

```
Router# configure terminal
```

Step 2 Use the **crypto ipsec profile** *name* command to enter the IPsec profile.

Example:

```
Router(config)# crypto ipsec profile P1
```

Step 3 Use the **set pfs** *group* **pqc** *mlkem-algorithm* command to enable PFS, specify a DH group, and an ML-KEM algorithm, or both.

Example:

```
Router(ipsec-profile)# set pfs group19 pqc mlkem1024
```

Note
When **set pfs** is used:

- Without PQC: Starting from Cisco IOS XE Release 26.2.1 onwards, the DH group is inherited from the IKEv2 SA.
- With PQC: The DH group is inherited from the IKEv2 SA. Exchanges are optimized for Cisco IOS XE to Cisco IOS XE deployments using an ML-KEM-only IPsec Child Security Association exchange.

Verify the PQC hybrid key exchange

Follow these steps to verify that the PQC hybrid key exchange is active.

Procedure

Step 1 Use the **show crypto ikev2 sa detail** command to verify the IKEv2 SA and the PQC key exchange algorithm.

Example:

```
Router# show crypto ikev2 sa detail

IPv4 Crypto IKEv2 SA

Tunnel-id Local              Remote             fvrf/ivrf   Status
1         10.0.149.203/500   10.0.149.217/500   none/none   READY
Encr: AES-CBC, keysize: 256, PRF: SHA512, Hash: SHA512, DH Grp:19, Auth sign: PSK, Auth
verify: PSK
PQC Key Exchange: ML-KEM-1024
...
Quantum-safe Encryption using PQC: ML-KEM-1024
PEER TYPE: IOS-XE
```

Step 2 Use the **show crypto ipsec sa** command to verify the IPsec SA. The PQC key exchange appears after the first IPsec SA rekey.

Example:

```
Router# show crypto ipsec sa
...
PFS (Y/N): Y, DH group: group19, PQC Key Exchange: ML-KEM-1024
```

Note
The PQC key exchange value in the IPsec SA output appears only after an IPsec SA rekey.

When the show commands display the configured ML-KEM algorithm and the tunnel status is READY, the quantum-safe hybrid key exchange is active.

Troubleshooting PQC hybrid key exchange

Use these debug commands to troubleshoot PQC hybrid key exchange issues on a Cisco IoT router.

Follow these steps to debug the PQC hybrid key exchange.

Procedure

Step 1 Use the **debug crypto ikev2** commands to debug IKEv2 negotiation, errors, packets, and internal events.

Example:

```
Router# debug crypto ikev2
Router# debug crypto ikev2 error
Router# debug crypto ikev2 packet
Router# debug crypto ikev2 internal
```

Step 2 Use the **debug crypto kmi** command to debug Key Management Infrastructure (KMI) events.

Example:

```
Router# debug crypto kmi
```

Step 3 Use the **debug crypto socket** command to debug crypto socket events.

Example:

```
Router# debug crypto socket
```

Step 4 Use the **debug crypto ipsec** commands to debug IPsec negotiation, errors, messages, and state changes.

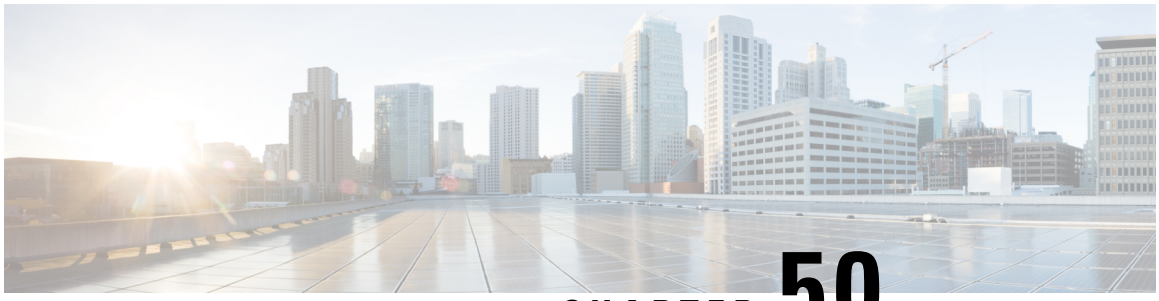
Example:

```
Router# debug crypto ipsec
Router# debug crypto ipsec error
Router# debug crypto ipsec message
Router# debug crypto ipsec state
```

Step 5 Use the **debug tunnel protection** command to debug tunnel protection events.

Example:

```
Router# debug tunnel protection
```



CHAPTER 50

Post Quantum Cryptography for SSHv2 sessions

Table 34: Feature History Table

Feature name	Release information	Feature description
Post Quantum Cryptography for SSHv2 sessions	Release 26.2.1	Cisco IoT routers can combine the Module-Lattice-Based Key-Encapsulation Mechanism (ML-KEM) with classical elliptic-curve key-exchange algorithms to establish keys for SSHv2 sessions.

- Implementation of Post Quantum Cryptography, on page 451
- Overview, on page 452
- Factors to consider for deploying PQC, on page 452
- Protocol requirements, on page 452
- Restrictions for PQC, on page 452
- PQC negotiation scenarios , on page 453
- Configure PQC for SSHv2 sessions, on page 453
- Verify PQC for SSHv2 sessions, on page 454
- Key elements of PQC, on page 454
- Supported platforms, on page 456

Implementation of Post Quantum Cryptography

Cisco utilizes a hybrid approach to implement Post Quantum Cryptography (PQC). The hybrid approach to PQC combines a quantum-resistant algorithm, such as the module-lattice based ML-KEM (e.g., mlkem768), with a classical elliptic curve key exchange algorithm like NIST P-256 or X25519, along with a hash function such as SHA-256.

This combination provides enhanced security by protecting against both classical and quantum attacks while maintaining compatibility and performance with existing cryptographic protocols.

Cisco implements this hybrid approach in protocols like SSH allowing configurations that enforce the use of ML-KEM algorithms for quantum resistance without falling back to classical-only key exchanges.

Overview

PQC refers to cryptographic algorithms designed to be secure against attacks from quantum computers which can break many classical encryption methods. Current widely used encryption methods—such as Diffie-Hellman (DH), and Elliptic Curve Diffie-Hellman (ECDH) — are vulnerable to being broken as quantum computing technology advances.

Implementing quantum-safe encryption now is critical for the long-term confidentiality and integrity of sensitive communications, particularly for government, military, and financial sectors.

Factors to consider for deploying PQC

Review these factors before deploying PQC.

Backward compatibility with classical algorithms
If a device is only configured for PQC and the SSH client does not support PQC, the SSH session will fail and no connection will be established.
Impact on computational time
The key size in PQC implementations may have an impact on the computational time due to the mathematical complexity and processing requirements of quantum-resistant algorithms. Larger key sizes generally increase the computational workload. PQC implementations must consider hardware acceleration and software optimizations to mitigate the increased computational time due to larger key sizes.

Protocol requirements

Protocol requirements

SSHv2 :

To establish secure, encrypted, and remote command-line access to servers and network devices over unsecured network use SSH version 2.0. PQC cannot be implemented on SSHv1 due to severe security vulnerabilities and it does not support the modern, complex key exchange mechanisms required to resist quantum computer attacks.

Key Pairs :

Ensure that RSA key pair or EC key pair for the SSH server is generated using crypto key generate command.

Restrictions for PQC

Implementing PQC is not supported on devices that operate in Federal Information Processing Standards (FIPS) mode.

PQC negotiation scenarios

During the SSH protocol connection setup, the client and server negotiate which cryptographic algorithms to use for securing the session. This negotiation process is a basic example of how algorithm negotiation works in SSH and is applicable to any similar negotiation process.

The algorithms are considered in the order they are configured on the device, meaning the priority or preference of algorithms is determined by the sequence in which they are listed in the configuration. The negotiation proceeds by trying the algorithms in that configured order until a mutually supported algorithm is found and agreed upon for use in the session. The server must support at least one algorithm from the client's list for successful negotiation.

Table 35: sfds

Client configuration	Server configuration	Negotiated algorithm
Classical, Hybrid	Classical, Hybrid	Classical
Hybrid, Classical	Hybrid, Classical	Hybrid
Classical, Hybrid	Hybrid	Hybrid
Hybrid, Classical	Classical	Classical
Classical	Classical, Hybrid	Classical
Hybrid	Classical, Hybrid	Hybrid
Classical	Hybrid	None
Hybrid	Classical	None
Mixed (No overlap)	Mixed (No overlap)	None

Configure PQC for SSHv2 sessions

Use these commands to set up devices as an SSH server and an SSH client. This allows the device to accept connection requests from remote SSH clients, enabling secure remote management of the device through SSH connections.

Note The PQC algorithms are not enabled by default and must be explicitly enabled.

Procedure

Step 1 Use the **configure terminal** command to enter global configuration mode.

Example:

```
Router# configure terminal
```

Step 2 Use the **ip ssh server algorithm kex** *algorithm* command to configure the PQC based key-exchange algorithms that the SSH server supports.

Example:

```
Router(config)# ip ssh server algorithm kex mlkem768nistp256-sha256
```

Step 3 Use the **ip ssh client algorithm kex** *algorithm* command to configure the PQC based key-exchange algorithms that SSH client supports.

Example:

```
Router(config)# ip ssh client algorithm kex mlkem768x25519-sha256
```

This command is used to define the order and selection of key exchange (KEX) algorithms that the SSH client will use during the SSH handshake process.

Verify PQC for SSHv2 sessions

Use this section to understand how to verify PQC implementation for SSHv2.

Command	Purpose	What to look for in output
show ip ssh	Displays the status of the SSH server, its configuration details.	In the output of this command, look for these details that indicate PQC implementation is successful: KEX algorithms: **mlkem1024nistp384-sha384,** **mlkem768nistp256-sha256,** **mlkem768x25519-sha256**

This example shows the response when a PQC algorithm command is entered on a platform that does not support PQC.

```
Router(config)# ip ssh server algorithm kex mlkem1024nistp384-sha384 ecdh-sha2-nistp256
Router(config)#
000205: Jun 18 16:41:43.086: %SSH-3-KEX_PQC_NOT_SUPP: PQ/T hybrid key exchange algorithms
are not supported on this platform.
Router(config)#
```

Key elements of PQC

PQC is implemented by using the algorithms available as part of Module-lattice based Cryptography. Module-lattice based Cryptography is a type of quantum-resistant public-key cryptography that uses structured, high-dimensional grid points (lattices) to secure data. Cisco implements PQC into protocols like SSH using ML-KEM.

These algorithms are resistant to quantum attacks and are used for secure key establishment, encryption, and digital authentication or signatures.

This table lists the areas in which each algorithm can be used:

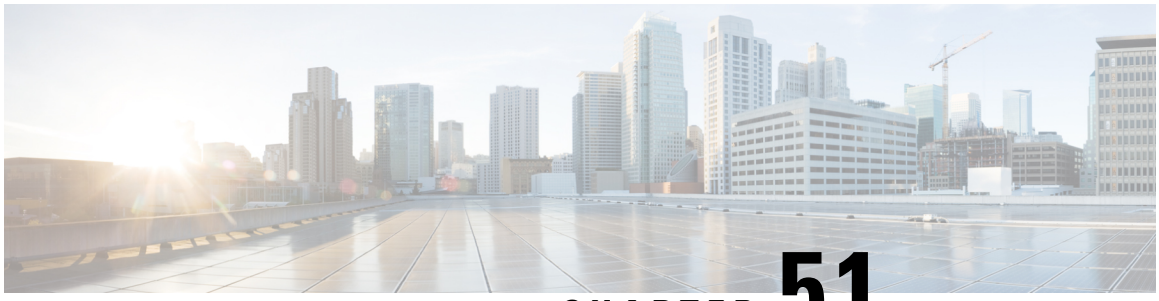
ML-KEM algorithm	What this algorithm means	Security Level
mlkem768x25519-sha256	ML-KEM 768 is a PQC algorithm based on module-lattice cryptography, providing resistance against quantum attacks. X25519 is a widely used classical elliptic curve Diffie-Hellman key exchange algorithm. SHA-256 is the hash function used for cryptographic hashing in the key exchange process. In summary, mlkem768x25519-sha256 is a hybrid PQC key exchange algorithm combining the ML-KEM 768-bit lattice-based scheme with classical X25519 and SHA-256, used to secure key exchanges against quantum and classical threats in Cisco's PQC-enabled network protocols.	The security level of this implementation is balanced and is best suited for deployments where performance metrics are critical.
mlkem768nistp256-sha256	ML-KEM 768 is a PQC algorithm based on module-lattice cryptography, providing resistance against quantum attacks. NIST P-256 is a widely used classical elliptic curve Diffie-Hellman key exchange algorithm standardized by NIST. SHA-256 is the hash function used for cryptographic hashing in the key exchange process. In summary, mlkem768x25519-sha256 is a hybrid PQC key exchange algorithm combining the ML-KEM 768-bit lattice-based scheme with classical X25519 and SHA-256, used to secure key exchanges against quantum and classical threats in Cisco's PQC-enabled network protocols.	The security level of this implementation is good and is best suited for most deployments.

ML-KEM algorithm	What this algorithm means	Security Level
mlkem1024nistp384-sha384	ML-KEM 768 is a PQC algorithm based on module-lattice cryptography, providing resistance against quantum attacks. 1024 is the highest security parameter set, equivalent to the security level of AES-256. NIST P-384 is the traditional (classical) component. It uses Elliptic Curve Diffie-Hellman (ECDH) with the NIST P-384 curve to maintain security against current non-quantum threats. In summary, mlkem1024nistp384-sha384 represents a hybrid cryptographic proposal combining a quantum-resistant ML-KEM algorithm with classical elliptic curve cryptography (NIST P-384) and SHA-384 hashing. This hybrid approach ensures security against both classical and quantum attacks by negotiating keys using both classical and post-quantum algorithms.	The security level of this implementation is the highest and is best suited for deployments where maximum security is needed.

Supported platforms

PQC can be configured only on these platforms in autonomous mode:

- Cisco Catalyst IR1101 Rugged Series Routers,
- Cisco IR1000 Rugged Series Secure Routers
- Cisco Catalyst IR1800 Rugged Series Routers,
- Cisco Catalyst IR8140 Heavy Duty Series Routers,
- Cisco Catalyst IR8340 Rugged Series Routers,
- Cisco Embedded Service 6300 Series Routers.



CHAPTER 51

DHCP Option 82

Table 36: Feature History Table

Feature name	Release information	Feature description
DHCP Option 82	Release 26.2.1	Starting with Cisco IOS XE Release 26.2.1, DHCP servers on routers support DHCP Option 82 relay information. The DHCP server can use the Circuit ID and Remote ID inserted by a DHCP relay agent to select an address pool for a client. This support enables predictable, location-based IP address assignment without requiring administrators to track client MAC addresses.

- DHCP Option 82, on page 457
- Prerequisites, on page 458
- Key considerations, on page 459
- Limitations, on page 459
- DHCP Option 82 packet flow, on page 459
- Hexadecimal pattern and suboption decoding, on page 460
- Configure DHCP Option 82 on server, on page 461
- Configure DHCP address pool, on page 463
- Configure DHCP relay agent, on page 464
- Configure relay chaining on relay agents, on page 465
- Verify DHCP Option 82, on page 466
- Debug DHCP Option 82 logs, on page 467

DHCP Option 82

The Dynamic Host Configuration Protocol (DHCP) Option 82 feature allows the server to allocate dynamic IP addresses based on the relay information sent by the relay agent. It enables predictive IP address assignment by the server based on network switch ports.

DHCP Option 82 enables DHCP relay agents to insert additional information such as Circuit ID, Remote ID and other configuration information into DHCP request messages before forwarding them from the client to the server.

Starting with Cisco IOS XE Release 26.2.1, DHCP servers support DHCP Option 82 on routers. This platform-independent feature includes the following suboptions:

- Circuit ID suboption: Identifies the relay agent's source interface or port where the client is connected.
- Remote ID suboption: Identifies the relay agent.

The DHCP server allows for address reservation based on Option 82 values, enabling consistent IP assignments per interface or location.

Benefits of DHCP Option 82

The use of DHCP Option 82 provides several operational and security advantages for network management such as:

- Predictable IP assignment: Enables static or predictable IP assignment for DHCP clients based on switch ports instead of MAC addresses.
- Enhanced security: Improves IP address management and security.
- Operational efficiency: Eliminates manual intervention or the need to track device MAC addresses.
- Granular control: Allows ISPs and network administrators to map specific user accounts to physical locations, enabling consistent, location-based IP assignments.

Prerequisites

Before configuring DHCP Option 82, you must ensure these requirements are met:

- A relay agent must be configured with global and VLAN-based DHCP snooping enabled.
- A DHCP server is available.
- A DHCP client is connected.
- A relay chaining topology is present wherever applicable.
- The client must be connected to the relay agent through an SVI interface for enabling VLAN-based snooping.
- Class based support must be enabled on the DHCP server.



Note You must know the hexadecimal value of each byte location in DHCP Option 82 to configure the relay-information hexadecimal command.For more information, see Circuit ID and Remote ID decode values

Key considerations

Key considerations for using DHCP Option 82 are:

- DHCP Option 82 support must be enabled on both the relay agent and DHCP server.
- You need to consider whether to use the DHCP Option 82 feature or not for predictable usage of IP assignment for DHCP clients.
- You need to consider whether to enable DHCP Option 82 globally for all interfaces or selectively on specific VLANs or interfaces only.
- You need to consider which information you want to match in the DHCP message received by the server. This can be Circuit ID, Remote ID or both.

Limitations

Limitations while using DHCP Option 82 are:

- Support must be enabled on both the relay agent and the DHCP server for it to work properly.
- If the DHCP server ignores DHCP Option 82, port-based deterministic assignment will not occur.
- It is not suitable for environments where MAC based IP address assignment is required.

DHCP Option 82 packet flow

Summary

DHCP Option 82 enables network administrators to track the location information of DHCP clients by inserting additional metadata as DHCP packets traverse relay agents. This process ensures accurate IP assignment and aids in network management.

Workflow

The DHCP packet flow through a network when DHCP Option 82 is enabled is as given:

1. The DHCP client broadcasts a DHCP request on the local network.
2. The first DHCP relay agent intercepts the broadcast and inserts the DHCP Option 82 field, which includes suboptions such as Circuit ID and Remote ID.
3. If a second relay agent resides closer to the DHCP server, the second relay agent encapsulates the DHCP Option 82 field coming from the first relay agent and forwards it to the DHCP server. This process preserves the information from both relay agents.
4. The relay agent sets the gateway IP address (giaddr) field in the DHCP packet to its own IP address. The giaddr field enables the DHCP server to identify the subnet from which the request originated.
5. The relay agent unicasts the DHCP packet to the DHCP server.

6. The DHCP server uses the DHCP Option 82 information to determine the appropriate IP address pool and configuration parameters for the client.
7. The DHCP server sends the reply to the relay agent using the giaddr address.
8. The relay agent removes the DHCP Option 82 information from the reply before forwarding the DHCP reply to the client.

Hexadecimal pattern and suboption decoding

The DHCP Option 82 hexadecimal pattern can be decoded for Circuit ID and Remote ID separately as shown in the examples.

Hexadecimal pattern sample

The Option 82 hexadecimal pattern `010600040005010a020701057065747261` consists of two primary suboptions:

- `01 06 00040005010a`: Circuit ID (Suboption 1).
- `02 07 01057065747261`: Remote ID (Suboption 2).

Field	Meaning
01	Circuit-ID suboption.
06	Circuit-ID length.
00040005010a	Circuit-ID value.
02	Remote-ID suboption.
07	Remote-ID length.
01057065747261	Remote-ID value.

Circuit ID Decode

The Circuit ID `00040005010a` identifies client's point of attachment specifying the port and VLAN of the connected client. By default, it follows the **vlan-mod-port** format.

```
Router# show cdp neighbors
         Device ID        Local Intrfce     Holdtme    Capability  Platform  Port ID
         device1         Gig 1/10            175           R S I  ESR-6300- Gig 0/1/0

Router# show vlan brief
         VLAN          Name                                Status    Ports
         5    TURBINE_INTERNAL                 active    Gi1/10
```

Note Decoding the Circuit ID `00040005010a`:

- `00`: Default format (vlan-mod-port).
- `04`: Length (4 bytes).
- `0005`: VLAN ID (Hex 0005 = Decimal 5).
- `010a`: Module 1, Port 10 (Hex 0a = Decimal 10).

Remote ID Decode

The Remote ID `01057065747261` identifies the originating device (MAC address or Hostname). It can be configured to use the device hostname.

```
Router(config)# ip dhcp snooping information option format remote-id hostname
```

Note Decoding the Remote ID `01057065747261`:

- `01`: Hostname format.
- `05`: Length (5 bytes).
- `7065747261`: Hexadecimal ASCII for router.

Configure DHCP Option 82 on server

To configure DHCP Option 82 on Cisco IR router server, you need to follow these tasks:

- Enable DHCP Option 82 for DHCP address allocation on the server.
- Define the DHCP class and relay agent information patterns.
- Define the DHCP address pool on the server.
- Enable relay chaining on DHCP relay agents.
- Verify DHCP Option 82
- Debug DHCP Option 82 logs.

Follow these steps to configure DHCP Option 82 on the server.

Procedure

Step 1 Use the **configure terminal** command to enter global configuration mode.

Example:

```
Router# configure terminal
```

Step 2 Use the **ip dhcp class** *class_name* command to enable DHCP address allocation using DHCP class.

Example:

```
Router(config)# ip dhcp class class_name
```

Note

- You can use Step 2 to enable DHCP Option 82 for DHCP address allocation on the server.
- You can disable this functionality, without deleting the DHCP class configuration.

Step 3 Use the **relay agent information** command to enter relay agent information option configuration mode and define the DHCP class and relay agent information patterns.

Example:

```
Router(dhcp-class)# relay agent information
```

Note
The `relay agent information` command validates only the presence of Option 82 in the DHCP packet. It does not evaluate the Circuit ID or Remote ID. As a result, any DHCP packet containing Option 82, regardless of its source port or relay agent will match this DHCP class and be assigned an IP address.

Step 4 (Optional) Use the **relay-information hex** *pattern* [*] [**mask** *mask*] command to optionally specify a hexadecimal value for the full relay information option.

Example:

```
Router(dhcp-class-relayinfo)# relay-information hex 010600040005010a020701057065747261
```

Example:

```
Router(dhcp-class-relayinfo)# relay-information hex 01030a0b0c02050000000123 mask
00000000000000000ffffffffffffffff
```

Example:

```
Router(dhcp-class-relayinfo)# relay-information hex 010600040005010a*
```

Note

- The Option 82 hexadecimal pattern includes two sub options: one for Circuit ID and the other for Remote ID.
- Each suboption can be decoded to represent the suboption type, length, VLAN, module port, client interface and MAC address.
- You can match the hexadecimal pattern either fully or partially for specific scenarios of IP assignment.
- You can use the mask option to mask either the Circuit ID or Remote ID partially.
- You can use `*` as a wildcard pattern at the end of the hexadecimal pattern to match only the beginning of the relay information value.
- If you omit this step, no pattern is configured and it is considered a match to any relay agent information option value, but the relay information option must be present in the DHCP packet.

- Repeat Step 4 for each DHCP class you need to configure.

What to do next

For more information, see Configuring DHCP Address Allocation Using Option 82.

Configure DHCP address pool

Follow these steps to configure and define DHCP address pool for DHCP address allocation.

Procedure

Step 1 Use the **configure terminal** command to enter global configuration mode.

Example:

```
Router# configure terminal
```

Step 2 Use the **ip dhcp pool** *name* command to configure a DHCP address pool on a DHCP server and enter DHCP pool configuration mode.

Example:

```
Router(config)# ip dhcp pool ABC
```

Step 3 Use the **network** *network-number* [*mask* |*prefix-length*] command to configure the subnet number and mask for a DHCP address pool on the DHCP server.

Example:

```
Router(dhcp-pool)# network 10.0.20.0 255.255.0.0
```

Step 4 Use the **class** *name* command to associate a class with a pool and enters DHCP pool class configuration mode.

Example:

```
Router(dhcp-pool)# class CLASS1
```

Note
This command will also create a DHCP class if the DHCP class is not yet defined.

Step 5 Use the **address range** *start-ip end-ip* command optionally to set an address range for a DHCP class in a DHCP server address pool.

Example:

```
Router(dhcp-pool-class)# address range 10.0.20.1 10.0.20.100
```

Note

- Repeat Step 5 for each DHCP class you need to associate to the DHCP pool.

- If this command is not configured for a class, the default value is the entire subnet of the pool.

Configure DHCP relay agent

Use this task to configure the client-facing interface as a relay agent port on the router.

Procedure

Step 1 Use the **configure terminal** command to enter the global configuration mode.

Example:

```
Router# configure terminal
```

Step 2 Use the **ip dhcp snooping** command to enable the DHCP snooping feature globally

Example:

```
Router(config)# ip dhcp snooping
```

Step 3 Use the **ip dhcp snooping vlan** *vlan-id* command to enable DHCP snooping for VLAN.

Example:

```
Router(config)# ip dhcp snooping vlan 5
```

Step 4 Use these steps to configure SVI:

a) Use the **interface** *vlan-id* command to select the SVI corresponding to the client VLAN.

Example:

```
Router(config)# interface Vlan5
```

b) Use the **ip address** *ip-address subnet-mask* command to assign an IP address and subnet mask to the SVI.

Example:

```
Router(config-if)# ip address 10.166.1.1 255.255.0.0
```

c) Use the **ip helper-address** *ip-address* command to specify the IP address of the DHCP server.

Example:

```
Router(config-if)# ip helper-address 110.1.1.1
```

Step 5 Use the **interface** command to select the specific interface connected to the client.

Example:

```
Router(config)# interface GigabitEthernet1/10
```

Step 6 Use the **description** *client* command to provide a label for the client connection.

Example:

```
Router(config-if)# description CLIENT-1
```

Step 7 Use the **switchport mode access** command to set the interface to access mode.

Example:

```
Router(config-if)# switchport mode access
```

Step 8 Use the **switchport access vlan** command to assign the interface to the client VLAN.

Example:

```
Router(config-if)# switchport access vlan 5
```

Step 9 (Optional) Use these commands to define custom Circuit ID and Remote ID values, overriding the defaults:

a) (Optional) Use the **ip dhcp snooping vlan** *vlan-id* **information option format-type circuit-id string** *circuit-id-string* command to override the default Circuit ID (**vlan mod port**) with a user-defined string.

Example:

```
Router(config-if)# ip dhcp snooping vlan 5 information option format-type circuit-id
string abc
```

b) (Optional) Use the **ip dhcp snooping information option format remote-id string** *remote-id-string* command to override the default Remote ID (MAC address) with a user-defined string.

Example:

```
Router(config)# ip dhcp snooping information option format remote-id string abc
```

c) Use the **exit** command to return to global configuration mode.

Example:

```
Router(config-if)# exit
```

What to do next

For more information, see Configuring the Cisco IOS XE DHCP Relay Agent.

Configure relay chaining on relay agents

Use this task to configure relay chaining when multiple relay agents are used in the network. Enabling encapsulation on the secondary relay agent ensures that the original DHCP Option 82 data is preserved as the request is forwarded to the DHCP server.

Before you begin

Note

- Ensure DHCP snooping is enabled on both relay agents for their respective client VLANs.
- Relay chaining requires that DHCP snooping be enabled on both relay agents for their respective client VLANs. Ensure that interfaces are correctly configured between the relay agents to allow the passage of DHCP traffic.

Procedure

Step 1 Use the **configure terminal** command to enter the global configuration mode.

Example:

```
Router# configure terminal
```

Step 2 Use the **ip dhcp relay information option vpn** command to enable the inclusion of VPN information in the relay agent inforamtion.

Example:

```
Router(config)# ip dhcp relay information option vpn
```

Step 3 Use the **ip dhcp relay information policy encapsulate** command to enable the encapsulation policy on the secondary relay agent.

Example:

```
Router(config)# ip dhcp relay information policy encapsulate
```

Note
In a double relay agent scenario, this configuration must be applied on the relay agent closest to the server.

Step 4 Use the **exit** command to return to privileged EXEC mode.

Example:

```
Router(config)# exit
```

An example of an encapsulated hexadecimal pattern format received by server in relay chaining is as given:

```
091f000000091a0a18bea80a0152120106000401f401050208000668d972ad3540020c020a00003c3c3c010e000000
```

Verify DHCP Option 82

Use these steps to verify DHCP Option 82 configuration.

Procedure

Step 1 Use these commands on the DHCP server side:

a) Use the **more system:running-config** command to display the running configuration.

Example:

```
Router# show running-config
```

Note
This command will display the configured DHCP classes and DHCP pools.

b) Use the **show ip dhcp binding** command to display the client's MAC IP address.

Example:

```
Router# show ip dhcp binding
```

Step 2 Use these commands on the DHCP relay agent side:

a) Use the **show ip dhcp snooping binding** command to verify the relay agent's snooping bindings.

Example:

```
Router# show ip dhcp snooping binding
```

b) Use the **show ip dhcp snooping** command to validate the Circuit ID and Remote ID.

Example:

```
Router# show ip dhcp snooping
```

Debug DHCP Option 82 logs

Use this task to troubleshoot DHCP Option 82 configurations and derive the required hexadecimal patterns for class matching by enabling debug logs on both the DHCP server and the relay agent.

Procedure

Step 1 Use the **debug ip dhcp server** command on the DHCP server to monitor server-side events, classes, and packets.

Example:

```
Router# debug ip dhcp server event
         Router# debug ip dhcp server class
         Router# debug ip dhcp server packet
```

Step 2 Use the **debug ip dhcp snooping** command on the relay agent to monitor snooping events, agent processing, and packets.

Example:

```
Router# debug ip dhcp snooping agent
         Router# debug ip dhcp snooping event
         Router# debug ip dhcp snooping packet
```

Note
You can analyze the generated logs to identify the `relay-information` hexadecimal pattern being received by the server.

Example:

```
*May 13 2026 08:25:19.629: DHCPD: Searching for a match to 'relay-information
010600040005010a020701057065747261' in class class_turbine
          *May 13 2026 08:25:19.629: DHCPD: input pattern 'relay-information
010600040005010a020701057065747261' matches class CLASS1
```