

Cisco Catalyst IR1101 Rugged Series Router Software Configuration Guide

First Published: 2018-05-18

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

© 2018–2026 Cisco Systems, Inc. All rights reserved.

Full Cisco Trademarks with Software License

THE SPECIFICATIONS AND INFORMATION REGARDING THE PRODUCTS IN THIS MANUAL ARE SUBJECT TO CHANGE WITHOUT NOTICE. ALL STATEMENTS, INFORMATION, AND RECOMMENDATIONS IN THIS MANUAL ARE BELIEVED TO BE ACCURATE BUT ARE PRESENTED WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED. USERS MUST TAKE FULL RESPONSIBILITY FOR THEIR APPLICATION OF ANY PRODUCTS.

THE SOFTWARE LICENSE AND LIMITED WARRANTY FOR THE ACCOMPANYING PRODUCT ARE SET FORTH IN THE INFORMATION PACKET THAT SHIPPED WITH THE PRODUCT AND ARE INCORPORATED HEREIN BY THIS REFERENCE. IF YOU ARE UNABLE TO LOCATE THE SOFTWARE LICENSE OR LIMITED WARRANTY, CONTACT YOUR CISCO REPRESENTATIVE FOR A COPY.

The Cisco implementation of TCP header compression is an adaptation of a program developed by the University of California, Berkeley (UCB) as part of UCB's public domain version of the UNIX operating system. All rights reserved. Copyright © 1981, Regents of the University of California.

NOTWITHSTANDING ANY OTHER WARRANTY HEREIN, ALL DOCUMENT FILES AND SOFTWARE OF THESE SUPPLIERS ARE PROVIDED “AS IS" WITH ALL FAULTS. CISCO AND THE ABOVE-NAMED SUPPLIERS DISCLAIM ALL WARRANTIES, EXPRESSED OR IMPLIED, INCLUDING, WITHOUT LIMITATION, THOSE OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT OR ARISING FROM A COURSE OF DEALING, USAGE, OR TRADE PRACTICE.

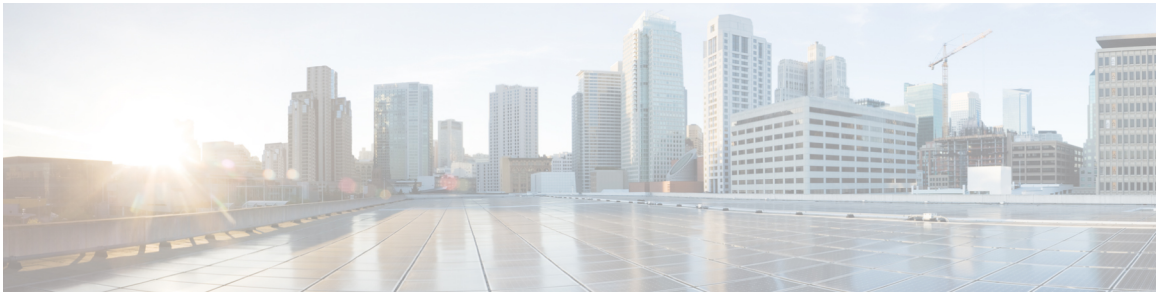
IN NO EVENT SHALL CISCO OR ITS SUPPLIERS BE LIABLE FOR ANY INDIRECT, SPECIAL, CONSEQUENTIAL, OR INCIDENTAL DAMAGES, INCLUDING, WITHOUT LIMITATION, LOST PROFITS OR LOSS OR DAMAGE TO DATA ARISING OUT OF THE USE OR INABILITY TO USE THIS MANUAL, EVEN IF CISCO OR ITS SUPPLIERS HAVE BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

Any Internet Protocol (IP) addresses and phone numbers used in this document are not intended to be actual addresses and phone numbers. Any examples, command display output, network topology diagrams, and other figures included in the document are shown for illustrative purposes only. Any use of actual IP addresses or phone numbers in illustrative content is unintentional and coincidental.

All printed copies and duplicate soft copies of this document are considered uncontrolled. See the current online version for the latest version.

Cisco has more than 200 offices worldwide. Addresses and phone numbers are listed on the Cisco website at www.cisco.com/go/offices.

Cisco and the Cisco logo are trademarks or registered trademarks of Cisco and/or its affiliates in the U.S. and other countries. To view a list of Cisco trademarks, go to this URL: https://www.cisco.com/c/en/us/about/legal/trademarks.html. Third-party trademarks mentioned are the property of their respective owners. The use of the word partner does not imply a partnership relationship between Cisco and any other company. (1721R)



CONTENTS

Full Cisco Trademarks with Software License iii

CHAPTER 1

Product Overview 1

Introduction 1

Accessing the CLI Using a Router Console 4

Using the Console Interface 7

Accessing the CLI from a Remote Console 7

Preparing to Connect to the Router Console Using Telnet 7

Using Telnet to Access a Console Interface 8

CLI Session Management 9

Information About CLI Session Management 9

Changing the CLI Session Timeout 9

Locking a CLI Session 9

CHAPTER 2

New Features 11

New Features for Cisco IOS XE 26.2.1 11

New Features for Cisco IOS XE 26.1.2 12

New Features for Cisco IOS XE 26.1.1 12

New Features for Cisco IOS XE 17.18.2 12

New Features for Cisco IOS XE 17.18.1a 12

New Features for Cisco IOS XE 17.17.1 13

New Features for Cisco IOS XE 17.16.1a 13

New Features for Cisco IOS XE 17.15.1a 13

New Features for Cisco IOS XE 17.14.1a 13

New Features for Cisco IOS XE 17.13.1 13

New Features for Cisco IOS XE 17.12.1a 14
New Features for Cisco IOS XE 17.11.1a 14
New Features for Cisco IOS XE 17.10.1a 14
New Features for Cisco IOS XE 17.9.1 14
New Features for Cisco IOS XE 17.8.1 15
Cellular serviceability enhancements 15
New Features for Cisco IOS XE 17.7.1 16
New Features for Cisco IOS-XE 17.6.1 16
New Features for Cisco IOS-XE 17.5.1 16
VXLAN 16
ERSPAN Support Cellular Interface as Source Interface 17
New Features for Cisco IOS-XE 17.4.1 18
New Features for Cisco IOS-XE 17.3.1 18
New Features for Cisco IOS-XE 17.2.1 18
New Features for Cisco IOS-XE 17.1.1 19

CHAPTER 3 **Basic Router CLI Configuration** 21
IR1101 Interface Naming 21
Basic Configuration 22
Configuring Global Parameters 26
Configuring the Gigabit Ethernet Interface 27
Support for sub-interface on GigabitEthernet0/0/0 28
Configuring a Loopback Interface 28
Enabling Cisco Discovery Protocol 29
Configuring Command-Line Access 30
Configuring Static Routes 31
Configuring Dynamic Routes 33
Configuring Routing Information Protocol 33
Configuring Enhanced Interior Gateway Routing Protocol 34
Configuring the Serial Interface 34
Specifying an Asynchronous Serial Interface 35
Specifying Asynchronous Serial Encapsulation 35
Configuring the Serial Port 35

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 55
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 72
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 Software Installation Methods 103
Install mode support 103
Bundle mode versus install mode 104
Installing software using install commands 105
Restrictions for installing the software using install commands 105
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 111
Configuration examples 112
One step installation 112
Complete three-step software installation 114
Show the installed packages 116
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 122
Consolidated Packages 122
Network-Essentials 122
Network-Advantage 122
Related Documentation 123
How to Install the Software for Cisco IOS XE 123

IOS XE downgrade warning 124
Enable the secure data wipe capability 125
ROMMON Images 126
File Systems 126
Option to Enable or Disable USB Access 127
Autogenerated File Directories and Files 128
Flash Storage 129
LED Indicators 129
Custom Controlled LED 130
Initial Bootup Security Improvements 130
Related Documentation 130

CHAPTER 12 **Software Maintenance Upgrade** 131
Software Maintenance Upgrade (SMU) Overview 131
SMU Work-flow and Basic Requirements 132
SMU Example 132
Installing a Patch Image 133
Uninstalling the Patch Image 134
Uninstalling the Patch Image Using Rollback 135
Uninstalling the Patch Image Using Deactivate, Commit, and Remove 136

CHAPTER 13 **Smart Licensing Using Policy** 139
Smart licensing using policy on Cisco routers 139

CHAPTER 14 **Ethernet Switch Ports** 141
Configuring VLANs 141
VLAN Trunking Protocol (VTP) 142
Configuring IEEE 802.1X Port-Based Authentication 142
Configuring Spanning Tree Protocol 143
Configuring MAC Address Table Manipulation 144
L2 Sticky Secure MAC Addresses 145
Configuring Switch Port Analyzer 145
IGMP Snooping for IPv4 146
Multi-Homing Configuration in BGP EVPN VXLAN Fabric 146

Configuring Multi-Homing in a BGP EVPN VXLAN Fabric 147
Configure Ethernet Segment and Redundancy in the Ethernet Segment 148
Configure Multi-Homing VLAN and Associate EVPN Ethernet-Segment 148
Configure SVI Service Instance 150
Configure EVPN-Instance 150
Apply the Configuration on the Bridge-Domain 151
Configuration Example 152
Verify the Multi-Homing Configuration 159
Misconfiguration of Multihoming VLANs 167

CHAPTER 15 **IP Device Tracking** 173
IP Device Tracking on routers 173
Guidelines and limitations 174
Options to enable SISF-based device tracking 174
Create and attach a custom device tracking policy 174
Apply the default device tracking policy to an interface target 176
Configuration and verification examples 177

CHAPTER 16 **Cellular pluggable interface modules** 179

CHAPTER 17 **P-LTE 450 Support** 187
Support for P-LTE-450 187
Support for P-LTE-450 Pluggable Interface Module 187
VRF for the P-LTE-450 Module 189
Configure VRF for the P-LTE-450 module 189
P-LTE-450 Support on Autonomous Mode 190
P-LTE-450 Support Over SDWAN/vManage 191
Enhancing P-LTE-450 Network Management Through SNMP MIB Objects 192
Upgrade the P-LTE-450 PIM 193

CHAPTER 18 **Support for P-5GS6-RC-GL PIM** 195
Support for P-5GS6-RC-GL PIM 195

CHAPTER 19 **Support for MACsec and IPsec Encryption** 197

Support for MACsec and IPsec Encryption 197

Software Supported MACsec 197

CHAPTER 20 **Support for the X25 over TCP** 201

Support for the X25 over TCP (XOT) 201

CHAPTER 21 **Dying Gasp Support on Ethernet or GigEthernet and SFP Port on IR1101** 203

Overview 203

Dying Gasp Support on IR1101 Routers 203

Dying Gasp Configuration Command 203

Configure Dying Gasp 204

Configuration Example for Dying Gasp 204

Verify Dying Gasp Packet Generation 205

CHAPTER 22 **2nd L3 WAN Port for IR1101** 207

2nd L3 WAN Port for IR1101 207

CHAPTER 23 **Managing USB Access Control** 209

Option to Enable or Disable USB Access 209

CHAPTER 24 **Cisco Network Plug and Play Agent** 211

Prerequisites for Cisco Network Plug and Play Agent 211

Restrictions for Cisco Network Plug and Play Agent 212

Information About Cisco Network Plug and Play Agent 212

Cisco Network Plug and Play Deployment Solution 212

Cisco Network Plug and Play Features 213

Cisco Network Plug and Play Agent Services and Capabilities 214

Software Maintenance Upgrade 216

Cisco Network Plug and Play Agent 217

Cisco Network Plug and Play Server 217

Cisco Network Plug and Play Agent Deployment 218

Cisco Network Plug and Play Agent Network Topology 219
Cisco Network Plug and Play Agent Initialization 219
Cisco Network Plug and Play Agent Deployment Solutions 220
Cisco Network Plug and Play Agent Discovery Process 220
Plug-n-Play Agent Deployment Protocol 225
Plug-n-Play Agent Security 226
Security Methods for the PnP Discovery Process 227
Self-Signed Certificate Based Authentication 227
Mobile Device Based Secured Installation 227
CA-Signed Certificate based Authentication 228
DHCP Option based Discovery over an IPv4 Network 228
DHCP Option based Discovery over an IPv6 Network 229
DNS-based Discovery 230
DNS-based Discovery over an IPv6 Network 231
Cisco Cloud Redirection over an IPv4 and IPv6 Network 231
Cisco Network PnP Discovery Over 4G Interface 232
Cisco Network PnP Discovery over a Management Interface 233
Cisco PnP over an EtherChannel 233
Security Methods for Post-PnP Discovery Process 233
Certificate Install Service 233
SUDI-based PnP Application Level Authentication 234
SUDI-based Client Certificate Validation 234
SUDI-based Serial Number 234
SUDI-Based Device Authentication 234
How to Configure Cisco Network Plug and Play Agent 236
Configuring Cisco Network Plug and Play Agent Profile 236
Configuring Network Plug and Play Agent Device 237
Configuring Cisco Network Plug and Play Reconnect Factor 237
Configuring Cisco Network Plug and Play HTTP Transport Profile 238
Configuring Cisco Network Plug and Play HTTPS Transport Profile 240
Configuring Backup Cisco Network Plug and Play Device 241
Configuring Backup Cisco Network Plug and Play Reconnect Factor 242
Configuring Backup Cisco Network Plug and Play HTTP Transport Profile 243
Configuring Backup Cisco Network Plug and Play HTTPS Transport Profile 244

Configuring Cisco Network Plug and Play Agent Tag 245
Troubleshooting and Debugging 246
Glossary 247
Additional References for Open Plug-n-Play Agent 248

CHAPTER 25 **Cellular Boot Time Improvements** 249
Cellular boot time improvements 249

CHAPTER 26 **CLI Output for the FN980 5G Modem** 251
Change in CLI output for the FN980 5G modem 251

CHAPTER 27 **Supervisory Control and Data Acquisition** 253
SCADA Overview 253
Role of the IR1101 254
Key Terms 254
Protocol Translation Application 255
Prerequisites 255
Guidelines and Limitations 256
Default Settings 256
Configuring Protocol Translation 257
Enabling the IR1101 Serial Port and SCADA Encapsulation 257
EXAMPLE 257
Configuring T101 and T104 Protocol Stacks 258
Prerequisites 258
Configuring the T101 Protocol Stack 258
EXAMPLE 260
Configuring the T104 Protocol Stack 260
EXAMPLE 262
SCADA enhancement for TNB 263
Configuration Example 264
Yang Data Model Support for Scada 265
Scada Yang Models 265
Cisco-IOS-XE-scada-gw 266
Cisco-IOS-XE-scada-gw-oper 267

Configuring the DNP3 Protocol Stacks 267
Configuring DNP3 Serial 267
EXAMPLE 268
Configuring DNP3 IP 269
EXAMPLE 270
DNP3 Enhancement 271
Configuration Example 271
Starting and Stopping the Protocol Translation Engine 271
EXAMPLE 272
Verifying Configuration 272
Debug Commands 273

CHAPTER 28 **Raw Socket Transport** 275
Information About Raw Socket Transport 275
TCP Transport 276
UDP Transport 276
Serial Data Processing 277
VRF-Aware Raw Socket 277
Prerequisites 278
Guidelines and Limitations 278
Raw socket feature enhancement 278
Default Settings 278
Configuring Raw Socket Transport 279
Enabling Raw Socket Transport on the Serial Interface 279
EXAMPLE: Enable Serial Port 279
Configuring Common Raw Socket Line Options 279
EXAMPLE: Common Raw Socket Line Options 280
Configuring Raw Socket TCP 280
Configuring the Raw Socket TCP Server 281
Configuring the Raw Socket TCP Client 282
Configuring a Raw Socket UDP Peer-to-Peer Connection 283
EXAMPLE: Raw Socket UDP Connection 284
Rawsocket Keepalive Configuration CLI 284
Yang Data Model Support for Raw Socket Transport 285

Verifying Configuration 286
Configuration Examples 286
Raw Socket TCP 286
Raw Socket UDP 287
Raw Socket VRF 288

CHAPTER 29 **IRM-1101 Expansion Module 291**
IRM-1100 Expansion Module Overview 291
mSATA Overview 293
Using the mSATA SSD 293
Displaying the Wear Leveling Data for the mSATA SSD 294
MIB support for mSATA Wear Ratio and Usage 295
Example: Actual OID and output of SNMP get/walk on OID 295
Feature Details 295
Feature Assumptions 295
Yang model for mSATA 295
Support for mSATA and IO Support for IRM-1100-SPMI in CM Side 296
Digital IO 297
Configuration Commands 297
Configuration Examples 298
New Cellular Pluggable Modules 300
IRM-1100 Expansion Module on the Compute Side 301
SFP Support 301
Support 1G SFPs 304

CHAPTER 30 **IRM-1100-4A2T Expansion Module 305**
IRM-1100-4A2T Overview 305
Guidelines and Limitations 307
Deployment Scenarios 308
Scenario One 308
Scenario Two 309
Scenario Three 309
Scenario Four 310
Inventory Details based on Deployment 311

Gigabit Ethernet Switch Ports 312
LEDs 312
Async Ports 314
Serial RJ45 Pin-Outs 314
DCE Interface Configuration Steps 315
GPIO Configuration Pins 316
Configuration Examples for Additional Async Interfaces 318
Scada Protocol Translations 319
Serial Relay 321
Using the WebUI to Configure Async Ports 322

CHAPTER 31 IRM-1100-4S8I expansion module 325
Expansion module for IR1101 routers 325
Guidelines and limitations 326
Support for jumbo frames 326
Configure the MTU size on a router interface 326
Digital IOs 327
Configure and manage alarms on the router 328
Configure additional parameters on the router 329
Configuration and verification examples 330
Support for L2/L3 SFP interfaces 331

CHAPTER 32 IOx Application Hosting 335
Information About Application Hosting 335
Need for Application Hosting 335
IOx Overview 335
Cisco Application Hosting Overview 336
IOXMAN 336
GPS access to IOx Apps 336
Guest Shell as IOx Container APP 337
Application Hosting on the IR1101 Router 338
VirtualPortGroup 340
vNIC 341
How to Configure Application Hosting 342

Enabling IOx 342
Configuring a VirtualPortGroup to a Layer 3 Data Port 343
Native docker support 345
Digital IO for IOx container applications 346
Signed application support 347
Installing and Uninstalling Apps 348
Overriding the App Resource Configuration 349
Verifying the Application Hosting Configuration 350
Digital IO Enhancement 352
Configuration Examples for Application Hosting 352
Example: Enabling IOx 353
Example: Configuring a VirtualPortGroup to a Layer 3 Data Port 353
Example: Installing and Uninstalling Apps 353
Example: Overriding the App Resource Configuration 353
IOx Access to USB Storage 353
Cisco ThousandEyes Enterprise Agent 354

CHAPTER 33 **Cisco SD-WAN Support** 357
Cisco SD-WAN Overview 357
SD-WAN 358
Related Documentation 359

CHAPTER 34 **Serial Relay Service** 361
Serial Relay Service Overview 361
Data Paths 361
Configuration Commands 363

CHAPTER 35 **ROM Monitor** 365
ROM Monitor Overview 365
Access ROM Monitor Mode 366
Checking the Current ROMMON Version 366
Commonly Used ROM Monitor Commands 367
Rommon Command Examples 368
Changing the ROM Monitor Prompt 368

Displaying the Configuration Register Setting 368
Environment Variable Settings 369
Frequently Used Environmental Variables 369
Displaying Environment Variable Settings 369
Entering Environment Variable Settings 369
Saving Environment Variable Settings 370
Exiting ROM Monitor Mode 370
ROMMON Configuration Example 370
Upgrading the ROMmon for a Router 371
Async serial port for console 371

CHAPTER 36 **Digital Subscriber Line** 373
Overview 373
DSL Feature Specifications 375
Dying-Gasp SMS Notification for EM74XX Modems 376
Configuration Steps 377
Configuration Example 377
Installing the DSL SFP 377
DSL SFP Annex J support 380
DSL SFP Annex M support 381
Support for DSL Annex B 381
LED Indications on the SFP 381
DSL SFP Firmware Upgrade 382
Support DSL SFP Firmware Signing and Signature Validation 383
Command Line Interface 383
ADSL2/2+ Overview 384
Configuring ADSL2/2+ 384
ADSL2/2+ Controller Configuration Commands 385
ADSL2/2+ PVC Sub Mode 386
ADSL2+ Example 389
ADSL2 Annex A Example 390
ADSL2 Annex L Example 391
VDSL2 Overview 393
Configuring VDSL2 393

VDSL2 Controller Configuration Commands 394
VDSL Example 395
Yang Model for DSL 396
Support Four ADSL MIB Objects 397
Command Line Interface 397
Support ADSL MIB Objects 397
Support VDSL MIB Objects 397
DSL Troubleshooting 398
Frequently Asked Questions 404
Controller Status Messages 407
L1 Training Logs 408

CHAPTER 37 **Out Of Band Management** 411
Out Of Band Management (OOB) Overview 411
OOB Topology 411
Feature Caveats 412
OOB Configuration 412

CHAPTER 38 **HDLC Support for SCATS** 415
HDLC Support for SCATS Overview 415
Configure IOx Application 416
Enable IOx 417
Configure a VirtualPortGroup to a Layer 3 Data Port 417
Configure Serial Port for IOx Communication 418
Deploy SCATs Application 419
Deploy SCATs Application via Local Manager 419
Deploy SCATs Application Using the IOS-XE CLI 425
Troubleshooting 427

CHAPTER 39 **Ethernet VPN Virtual Extensible LAN Over Generic Routing Encapsulation** 429
Overview 429
Configuration Examples 430
Configuration Steps 432
Troubleshooting 434

Additional Resources 436

CHAPTER 40 **Cyber Vision Center** 437

Cyber Vision Support 437

Deployment of Cyber Vision Center (CVC) on IOS-XE platform 437

Example Configuration for ERSPAN over L3 configuration along with Virtual Port Groups 438

CLI Installation 439

LMGUI Installation 439

Register the Router Details 439

Capture Live Traffic 443

Installing CVC Sensor using LM GUI 444

Binding eth0 447

Binding eth1 450

Activate the Application 451

Starting the Application 453

CHAPTER 41 **Troubleshooting** 455

Troubleshooting Overview 455

Understanding Diagnostic Mode 455

Before Contacting Cisco or Your Reseller 456

show interfaces Troubleshooting Command 456

Change the Configuration Register 457

Configuring the Configuration Register for Autoboot 458

Reset the Router 459

Recovering a Lost Password 460

Reset the Password and Save Your Changes 461

Password Recovery Disable 461

Reset the Configuration Register Value 462

Configuring a Console Port Transport Map 463

Viewing Console Port, SSH, and Telnet Handling Configurations 464

Using the factory reset Commands 466

Support for Security-Enhanced Linux (SELinux) 466

Command Examples 467

Syslog Message Reference 468

CHAPTER 42 System Messages 471
Information About Process Management 471
How to Find Error Message Details 471

CHAPTER 43 Environmental Monitoring 477
Environmental Monitoring 477
Environmental Monitoring and Reporting Functions 477
Environmental Monitoring Functions 478
Environmental Reporting Functions 479
SNMP Polling of Temperature OID 485
Additional References 486
Technical Assistance 486

CHAPTER 44 WAN Monitoring 487
Information About WANMon 487
Built-in Recovery Actions 487
Prerequisites 488
Guidelines and Limitations 488
Configuring WANMon 488
Verifying WANMon Configuration 490
Configuration Examples 491
WANMon Cellular Interface Configuration Example 491
Multiple WAN Link Monitoring Example 491

CHAPTER 45 Process Health Monitoring 493
Monitoring Control Plane Resources 493
Avoiding Problems Through Regular Monitoring 493
Cisco IOS Process Resources 493
Overall Control Plane Resources 497
Monitoring Hardware Using Alarms 498
Router Design and Monitoring Hardware 499
BootFlash Disk Monitoring 499
Approaches for Monitoring Hardware Alarms 499

Viewing the Console or Syslog for Alarm Messages 499
Report Alarms Through SNMP 499
Yang Support for IO Ports 500
SNMP MIB for Digital I/O 501
SNMP MIB supports the show power CLI 501

CHAPTER 46 gRPC Network Operations 503
gRPC Network Operations Interface updates 503
GNMI Broker update 503

CHAPTER 47 Support for YANG 505
Support for YANG Data Models (Call-home) 505
Yang Data Model Support for Scada 505
Support for Model Driven support for GNMI Telemetry Dial-In 505

CHAPTER 48 Cellular Network Slicing 507
Cellular network slicing 507
Restrictions for cellular network slicing 508
Configure cellular network slicing 508
Example for configuring cellular network slicing 509
Verify cellular network slicing configuration 509

CHAPTER 49 H-VPLS N-PE Redundancy for MPLS Access 511
H-VPLS 511
Prerequisites for H-VPLS 511
Restrictions for H-VPLS 511
H-VPLS N-PE redundancy with MPLS access based on pseudowire redundancy 512
Configuration of H-VPLS N-PE Redundancy for MPLS Access 512
Specify the devices in the layer 2 VPN VFI 512
Specify the N-PE devices that form the layer 2 VPN cross connection with the U-PE 513
Configuration Example for H-VPLS N-PE Redundancy for MPLS Access 515
Example of configuring H-VPLS N-PE redundancy for MPLS access 515
Pseudowire types for H-VPLS 516

CHAPTER 50 **Support for 20,000 Routers** 519

CHAPTER 51 **Segment Routing over IPv6** 525

Segment Routing over IPv6 525

Prerequisites 526

Configure SRv6 526

Configure SRv6 under IS-IS 527

Verify SRv6 528

CHAPTER 52 **Quantum-Safe Encryption using Postquantum Preshared Keys** 531

Quantum-Safe Encryption with Postquantum Preshared Keys 531

Postquantum Preshared Key restrictions 534

Supported platforms 534

Configure manual Postquantum Preshared Keys 534

Configure dynamic Postquantum Preshared Keys 535

Configuration examples for Quantum-Safe Encryption Using Postquantum Preshared Keys 536

Postquantum Preshared Key configuration examples 537

Verify the Postquantum Preshared Key configuration 539

CHAPTER 53 **DHCP Option 82** 541

DHCP Option 82 541

Prerequisites 542

Key considerations 543

Limitations 543

DHCP Option 82 packet flow 543

Hexadecimal pattern and suboption decoding 544

Configure DHCP Option 82 on server 545

Configure DHCP address pool 547

Configure DHCP relay agent 548

Configure relay chaining on relay agents 549

Verify DHCP Option 82 550

Debug DHCP Option 82 logs 551

CHAPTER 54 **IKEv2 and IPsec PFS Post Quantum Cryptography** 553

IKEv2 and IPsec PFS Post Quantum Cryptography 553

Benefits of PQC 554

Prerequisites 554

Limitations 554

Phased migration in Hub-and-Spoke deployments 555

ML-KEM algorithm and parameter sets 555

PFS negotiation behavior 556

Configure an IKEv2 proposal for PQC 556

Configure Perfect Forward Secrecy with PQC 558

Verify the PQC hybrid key exchange 559

Troubleshooting PQC hybrid key exchange 559

CHAPTER 55 **Post Quantum Cryptography for SSHv2 sessions** 561

Implementation of Post Quantum Cryptography 561

Overview 562

Factors to consider for deploying PQC 562

Protocol requirements 562

Restrictions for PQC 562

PQC negotiation scenarios 563

Configure PQC for SSHv2 sessions 563

Verify PQC for SSHv2 sessions 564

Key elements of PQC 564

Supported platforms 566

CHAPTER 56 **Segment Routing over MPLS** 567

Segment Routing over MPLS 567

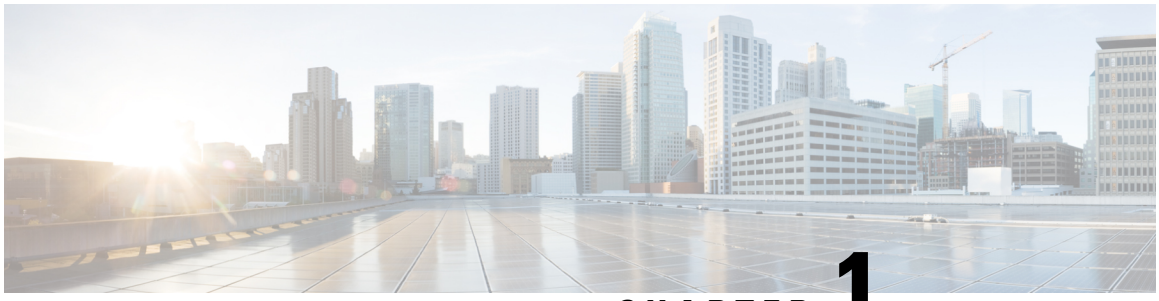
Prerequisites 568

Configure Segment Routing over MPLS 568

Configure Segment Routing MPLS on OSPF network 569

Configure Prefix-SID for OSPF 570

Verify SR-MPLS configuration 570



CHAPTER 1

Product Overview

This chapter contains the following sections:

- Introduction, on page 1
- Accessing the CLI Using a Router Console, on page 4
- Accessing the CLI from a Remote Console , on page 7
- CLI Session Management, on page 9

Introduction

The Cisco Catalyst IR1101 Rugged Series Router is a next generation modular industrial router which has a Base module with additional Pluggable Modules that can be added. The Pluggable Module provides the flexibility of adding different interfaces to the IR1101 platform.

Note The documentation set for this product strives to use bias-free language. For purposes of this documentation set, bias-free is defined as language that does not imply discrimination based on age, disability, gender, racial identity, ethnic identity, sexual orientation, socioeconomic status, and intersectionality. Exceptions may be present in the documentation due to language that is hardcoded in the user interfaces of the product software, language used based on RFP documentation, or language that is used by a referenced third-party product.

The IR1101 also has two Expansion Modules that add key capabilities such as dual LTE Pluggables, mSATA SSD FRU, SFP, additional ethernet and async ports,and Digital GPIO connections.

The IR1101 is the first IoT platform to run the Cisco IOS-XE operating system. IOS-XE is a Linux based OS that comes with many enhancements and more features compared to the classic IOS version.

This secion of the guide also includes:

IR1101 Base Router

The following figure shows the front panel of the IR1101 and highlights some of its capabilities:

Figure 1: IR1101 Front Panel

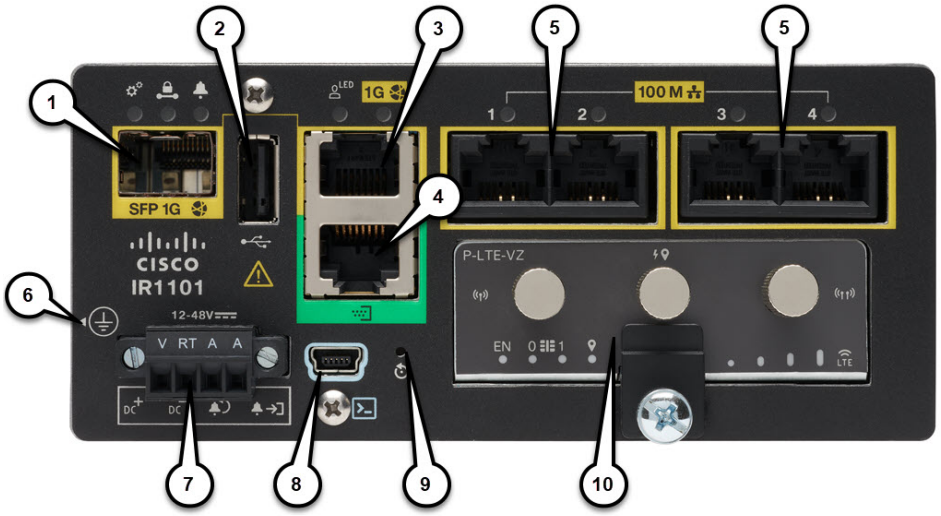


Table 1: Front Panel Descriptions

Item	Description
1	SFP GigE WAN Port (Combo port of #3 below)
2	Type A USB 2.0 Host Port
3	RJ45 GigE WAN Port (Combo port of #1 above)
4	Asynchronous Serial Port (DTE only)
5	RJ45 Fast Ethernet LAN Ports
6	Grounding Point (On side of device)
7	DC Power and Alarm Input
8	Type B Mini-USB Console Port
9	Reset Button
10	Pluggable Module Slot (ex. 4G/LTE module)

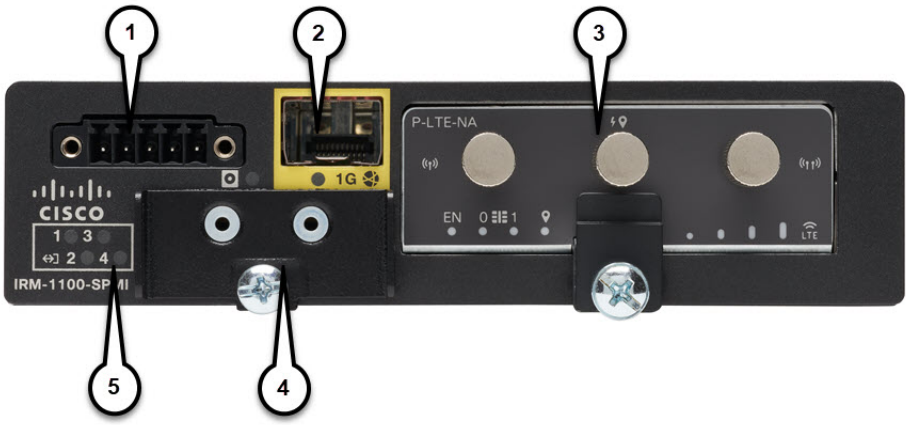
IRM-1100 Expansion Modules

The Expansion Module comes in two types:

- IRM-1100-SPMI
- IRM-1100-SP

The following figure shows the front panel of the IRM-1100-SPMI and highlights some of its capabilities:

Figure 2: IR-1100-SPMI Expansion Module Details



Item	Description
1	4 GPIO + 1 Return (Digital I/O) **Note** Functionality is available on Cisco IOS-XE release 16.12.1 and above.
2	SFP Connector
3	Pluggable Module
4	mSATA SSD Slot
5	Digital I/O LEDs

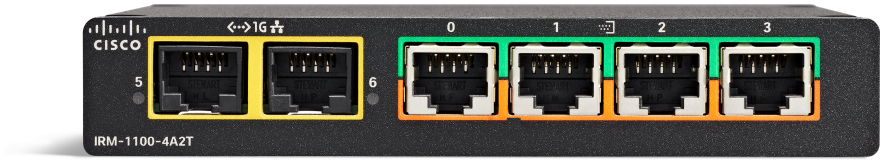
The IR-1100-SP Expansion Module is the same as the IR-1100-SPMI module, without the Digital I/O and mSATA components.

More information can be found in IRM-1101 Expansion Module, on page 291.

Complete details on the IR1101 can be found in the product data sheet.

IRM-1100-4A2T Expansion Module

The IRM-1100-4A2T is an expansion module that can be attached to the IR1101. It offers an additional four asynchronous serial ports and two Ethernet interfaces to the IR1101. The following graphic shows the IRM-1100-4A2T.



The IRM-1100-4A2T Ethernet interfaces are Layer 2 RJ45 10/100/1000 Mbps ports.

The IRM-1100-4A2T serial ports are RJ45 combo ports (RS232/RS485/RS422).

The IR1101 has two sides that expansion modules mount to. The top is called the Expansion side, and the bottom is called the Compute side. If the additional module is connected to the top, then it is referenced as the Expansion Module (EM) side. If the additional module is connected on the bottom, then it is referenced as the Compute Module (CM) side. Functionality differs depending on which side the expansion module is attached to, and how many and type of expansion modules are in use.

The IRM-1100-4A2T can be managed from the following tools:

- Cisco DNA Center
- WebUI

More information can be found in IRM-1100-4A2T Expansion Module, on page 305.

Accessing the CLI Using a Router Console

Cisco IR1101 routers have console port with only USB support. The console cable (Cisco P/N CAB-CONSOLE-USB, 6ft long) is not included and must be ordered.

The console port is a USB 2.0 mini USB Type B connector which is located on the front panel of the chassis. The default baud rate is 9600.

If your laptop or PC warns you that you do not have the proper drivers to communicate with the router, you can obtain them from your computers manufacturer, or go here: https://www.silabs.com/products/development-tools/software/usb-to-uart-bridge-vcp-drivers

On a device fresh from the factory, you are greeted with a System Configuration Dialog where you respond to basic configuration questions. If the router was ordered for the use of Cisco PnP connect services, in the case of centralized provisioning, the router skips the initial dialog. The following is an 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 '[]'.
```

```
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>

  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.
  Configure SNMP Network Management? [no]: <return>

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
FastEthernet0/0/1      unassigned      YES unset  down                  down
FastEthernet0/0/2      unassigned      YES unset  down                  down
FastEthernet0/0/3      unassigned      YES unset  down                  down
FastEthernet0/0/4      unassigned      YES unset  up                    up
Async0/2/0             unassigned      YES unset  up                    down
Vlan1                  unassigned      YES unset  up                    up
```



Note Names and IP addresses in this next section are shown as examples.

```
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 FastEthernet0/0/1
!
interface FastEthernet0/0/2
!
interface FastEthernet0/0/3
!
interface FastEthernet0/0/4
!
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


[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.



Press RETURN to get started! <return>


*Jul 27 21:35:24.369: %CRYPTO_ENGINE-5-KEY_ADDITION: A key named TP-self-signed-3211716068
 has been generated or imported by crypto-engine
*Jul 27 21:35:24.372: %SSH-5-ENABLED: SSH 1.99 has been enabled
*Jul 27 21:35:24.448: %PKI-4-NOCONFIGAUTOSAVE: Configuration was modified.  Issue "write
memory" to save new IOS PKI configuration
*Jul 27 21:35:24.532: %CRYPTO_ENGINE-5-KEY_ADDITION: A key named
TP-self-signed-3211716068.server has been generated or imported by crypto-engine
hostname>
```

The device now has a basic configuration that you can build upon.

Using the Console Interface

Procedure

Step 1 Enter the following command:

```
Router > enable
```

Step 2 (Go to Step 3 if the enable password has not been configured.) At the password prompt, enter your system password:

```
Password: enablepass
```

When your password is accepted, the privileged EXEC mode prompt is displayed.

```
Router#
```

You now have access to the CLI in privileged EXEC mode and you can enter the necessary commands to complete your desired tasks.

Step 3 To exit the console session, enter the **quit** command:

```
Router# quit
```

Accessing the CLI from a Remote Console

The remote console of the IR1101 can be accessed through Telnet or the more secure SSH. Details on telnet access follow in this chapter. For details on SSH access see Secure Shell, on page 61.

The following topics describe the procedure to access the CLI from a remote console:

Preparing to Connect to the Router Console Using Telnet

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.

Configuring the diagnostic and wait banners is optional, but recommended. The banners are especially useful as indicators to users about the status of their Telnet or SSH attempts.

To access the router remotely using Telnet from a TCP/IP network, configure the router to support virtual terminal lines using the **line vty** global configuration command. Configure the virtual terminal lines to require users to log in and specify a password.

See the Cisco IOS Terminal Services Command Reference document for more information about the line **vty global** configuration command.

To prevent disabling login on a line, specify a password with the **password** command when you configure the **login** command.

If you are using authentication, authorization, and accounting (AAA), configure the **login authentication** command. To prevent disabling login on a line for AAA authentication when you configure a list with the login authentication command, you must also configure that list using the **aaa authentication login** global configuration command.

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.

In addition, before you make a Telnet connection to the router, you must have a valid hostname for the router or have an IP address configured on the router. For more information about the requirements for connecting to the router using Telnet, information about customizing your Telnet services, and using Telnet key sequences, see the Cisco IOS Configuration Fundamentals Configuration Guide.

Using Telnet to Access a Console Interface

Procedure

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

- **connect host** *[port] [keyword]*
- **telnet host** *[port] [keyword]*

Here, *host* is the router hostname or IP address, *port* is a decimal port number (23 is the default), and *keyword* is a supported keyword. For more information about these commands, see the Cisco IOS Terminal Services Command Reference document.

Note
If you are using an access server, specify a valid port number, such as **telnet 172.20.52.40 2004**, in addition to the hostname or IP address.

The following 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:

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

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

Step 3 From user EXEC mode, enter the **enable** command:

```
Router> enable
```

Step 4 At the password prompt, enter your system password:

```
Password: enablepass
```

Step 5 When the **enable** password is accepted, the privileged EXEC mode prompt is displayed:

```
Router#
```

Step 6 You now have access to the CLI in privileged EXEC mode and you can enter the necessary commands to complete your desired tasks.

Step 7 To exit the Telnet session, use the **exit** or **logout** command.

```
Router# logout
```

CLI Session Management

An inactivity timeout is configurable and can be enforced. Session locking provides protection from two users overwriting changes that the other has made. To prevent an internal process from using all the available capacity, some spare capacity is reserved for CLI session access. For example, this allows a user to remotely access a router.

Information About CLI Session Management

An inactivity timeout is configurable and can be enforced. Session locking provides protection from two users overwriting changes that each other has made. To prevent an internal process from using all the available capacity, some spare capacity is reserved for CLI session access. For example, this allows a user to remotely access the router.

Changing the CLI Session Timeout

Procedure

Step 1 **configure terminal**

Enters global configuration mode

Step 2 **line console 0**

Step 3 **session-timeout** *minutes*

The value of *minutes* sets the amount of time that the CLI waits before timing out. Setting the CLI session timeout increases the security of a CLI session. Specify a value of 0 for *minutes* to disable session timeout.

Step 4 **show line console 0**

Verifies the value to which the session timeout has been set, which is shown as the value for `" Idle Session "`.

Locking a CLI Session

Before you begin

To configure a temporary password on a CLI session, use the **lock** command in EXEC mode. Before you can use the **lock** command, you need to configure the line using the **lockable** command. In this example the line is configured as **lockable**, and then the **lock** command is used and a temporary password is assigned.

Procedure

Step 1 `Router# configure terminal`

Enters global configuration mode.

Step 2 Enter the line upon which you want to be able to use the **lock** command.

`Router(config)# line console 0`

Step 3 `Router(config)# lockable`

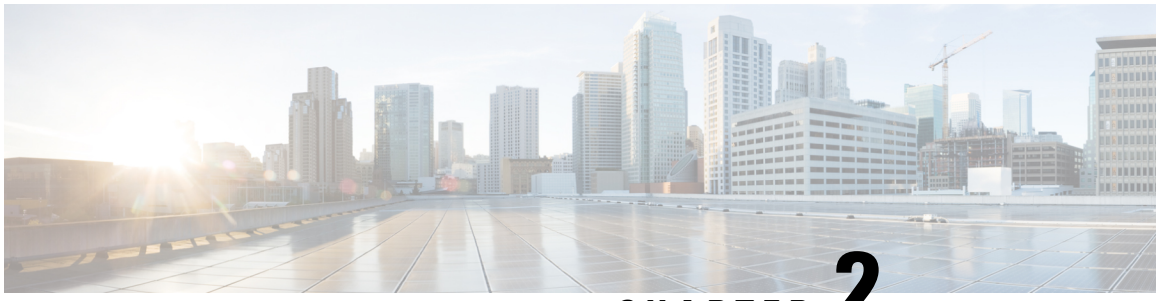
Enables the line to be locked.

Step 4 `Router(config)# exit`

Step 5 `Router# lock`

The system prompts you for a password, which you must enter twice.

```
Password: <password>
Again: <password>
Locked
```



CHAPTER 2

New Features

This document outlines the new features introduced in the current release.

- New Features for Cisco IOS XE 26.2.1, on page 11
- New Features for Cisco IOS XE 26.1.2, on page 12
- New Features for Cisco IOS XE 26.1.1, on page 12
- New Features for Cisco IOS XE 17.18.2, on page 12
- New Features for Cisco IOS XE 17.18.1a, on page 12
- New Features for Cisco IOS XE 17.17.1, on page 13
- New Features for Cisco IOS XE 17.16.1a, on page 13
- New Features for Cisco IOS XE 17.15.1a, on page 13
- New Features for Cisco IOS XE 17.14.1a, on page 13
- New Features for Cisco IOS XE 17.13.1, on page 13
- New Features for Cisco IOS XE 17.12.1a, on page 14
- New Features for Cisco IOS XE 17.11.1a, on page 14
- New Features for Cisco IOS XE 17.10.1a, on page 14
- New Features for Cisco IOS XE 17.9.1, on page 14
- New Features for Cisco IOS XE 17.8.1, on page 15
- New Features for Cisco IOS XE 17.7.1, on page 16
- New Features for Cisco IOS-XE 17.6.1, on page 16
- New Features for Cisco IOS-XE 17.5.1, on page 16
- New Features for Cisco IOS-XE 17.4.1, on page 18
- New Features for Cisco IOS-XE 17.3.1, on page 18
- New Features for Cisco IOS-XE 17.2.1, on page 18
- New Features for Cisco IOS-XE 17.1.1, on page 19

New Features for Cisco IOS XE 26.2.1

New features in this release are:

- Segment Routing over MPLS
- IKEv2 and IPsec PFS Post Quantum Cryptography
- Post Quantum Cryptography for SSHv2 sessions
- DHCP Option 82

New Features for Cisco IOS XE 26.1.2

New features in this release are:

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

New Features for Cisco IOS XE 26.1.1

New features in this release are:

- Segment Routing over IPv6
- GPS NMEA unique identifier
- Support for 20,000 Router Instances

New Features for Cisco IOS XE 17.18.2

New features in this release are:

- Cellular Network Slicing
- HVPLS N-PE Redundancy for MPLS Access
- Cisco ThousandEyes Enterprise Agent, on page 354
- Common HSEC License

New Features for Cisco IOS XE 17.18.1a

New feature in this release are listed below:

- IRM-1100-4S8I Expansion Module
- Support for jumbo frames
- IP Device Tracking
- VRF for the P-LTE-450 module
- Inline Tagging on switch interfaces
- IRM-1100-4S8I Expansion Module
- Quantum-Safe Encryption using Postquantum Preshared Keys

New Features for Cisco IOS XE 17.17.1

New feature 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:

- Monitoring Radio Signal Parameters and GPS Coordinates of Cellular Telemetry using Syslog Messages
- Enhancing P-LTE-450 Network Management Through SNMP MIB Objects
- mSATA LED Toggling
- Raw Socket Multi Host Capability

New Features for Cisco IOS XE 17.15.1a

New features in this release are listed below:

- Dying Gasp Support on Ethernet or GigEthernet and SFP Port on IR1101
- 2nd L3 WAN Port for IR1101

New Features for Cisco IOS XE 17.14.1a

New features in this release are listed below:

- Support for P-LTE-450 Pluggable Interface Module.
- Support for Class B End Devices in the LoRaWAN Network
- Support for MACsec and IPsec Encryption

New Features for Cisco IOS XE 17.13.1

This chapter contains the following sections:

- IOx Access to USB Storage
- P-LTE-450 Support on Autonomous Mode
- P-LTE-450 Support Over SDWAN/vManage
- Additional Modem Support for Cellular Pluggable Modules

- SD-WAN Remote Access (SD-WAN RA)
- 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:

- Support for P-LTE-450
- HDLC Support for SCATS Overview
- Uncapped License Implementation

New Features for Cisco IOS XE 17.11.1a

New features in this release are listed below:

- Async Serial Port for Console
- Change to Smart Licensing Packaging
- Galileo Support on the LTE Pluggable Modules

New Features for Cisco IOS XE 17.10.1a

New features in this release are listed below:

- Software Supported MACsec
- High Security (HSEC) License
- 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:

- Cellular Boot Time Improvements
- IOS XE Downgrade Warning
- SNMP Polling of Temperature OID
- GPS Mode Enabled By Default
- Install Mode Support

- Cisco WebUI Access Point Name (APN)

New Features for Cisco IOS XE 17.8.1

- Support for DSL Annex B, on page 381
- Support for mSATA and IO Support for IRM-1100-SPMI in CM Side, on page 296
- gRPC Network Operations Interface updates, on page 503
- GNMI Broker update, on page 503
- Raw socket feature enhancement, on page 278
- SCADA enhancement for TNB, on page 263

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

- IRM-1100 Expansion Module on the Compute Side, on page 301
- Support ADSL MIB Objects, on page 397
- Support VDSL MIB Objects, on page 397
- Support 1G SFPs, on page 304

New Features for Cisco IOS-XE 17.6.1

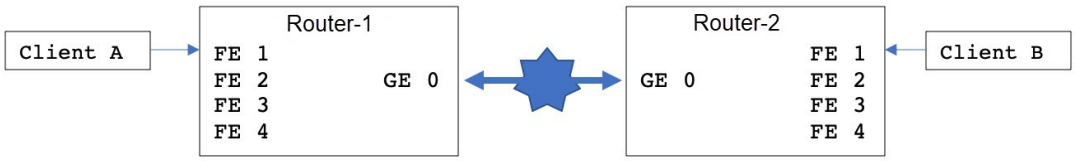
- Custom Controlled LED, on page 130
- Support DSL SFP Firmware Signing and Signature Validation, on page 383
- DSL SFP Annex M support, on page 381
- Support Four ADSL MIB Objects, on page 397
- Digital IO Enhancement, on page 352

New Features for Cisco IOS-XE 17.5.1

- DSL SFP Annex J support, on page 380
- Dying-Gasp SMS Notification for EM74XX Modems, on page 376
- SNMP MIB for Digital I/O, on page 501
- GPS access to IOx Apps, on page 336
- Yang model for mSATA, on page 295
- Guest Shell as IOx Container APP, on page 337
- SNMP MIB supports the show power CLI, on page 501
- Yang Model for DSL, on page 396
- DNP3 Enhancement, on page 271

VXLAN

VXLAN is a MAC in IP/UDP (MAC-in-UDP) encapsulation technique with a 24-bit segment identifier in the form of a VXLAN ID. The larger VXLAN ID allows LAN segments to scale to 16 million in a cloud network. In addition, the IP/UDP encapsulation allows each LAN segment to be extended across existing Layer 3 networks, making use of Layer 3 Equal-Cost Multi-Path (ECMP).



The configuration for the two devices is shown in the following table:

Router-1	Router-2
bridge-domain 1 member vni 6001 member Vlan100 service-instance 1 ! interface Loopback1 ip address 200.200.200.200 255.255.255.255 ! interface GigabitEthernet0/0/0 ip address 192.168.1.2 255.255.255.0 media-type rj45 ! interface FastEthernet0/0/1 switchport access vlan 100 ! interface Vlan100 no ip address service instance 1 ethernet encapsulation dot1q 100 //untag ! interface nve1 no ip address source-interface Loopback1 member vni 6001 ingress-replication 100.100.100.100 ! ip forward-protocol nd ip pim rp-address 200.200.200.200 ip http server ip http secure-server ip route 0.0.0.0 0.0.0.0 192.168.1.3 !	bridge-domain 1 member vni 6001 member Vlan100 service-instance 1 ! interface Loopback1 ip address 100.100.100.100 255.255.255.255 ! interface GigabitEthernet0/0/0 ip address 192.168.1.3 255.255.255.0 media-type rj45 ! interface FastEthernet0/0/1 switchport access vlan 100 ! interface Vlan100 no ip address service instance 1 ethernet encapsulation dot1q 100 //untag ! interface nve1 no ip address source-interface Loopback1 member vni 6001 ingress-replication 200.200.200.200 ! ip forward-protocol nd ip pim rp-address 100.100.100.100 no ip http server ip http secure-server ip route 0.0.0.0 0.0.0.0 192.168.1.2 !

ERSPAN Support Cellular Interface as Source Interface

Encapsulated Remote Switched Port Analyzer (ERSPAN) allows traffic from Cellular interfaces to be monitored. ERSPAN sends monitored traffic to a network analyzer.

The following is a sample configuration:

```
Router(config)#monitor session 1 type erspan-source
Router(config-mon-erspan-src)#no shut
Router(config-mon-erspan-src)#source interface Cellular0/1/0
Router(config-mon-erspan-src)#destination
Router(config-mon-erspan-src-dst)#erspan-id 1
Router(config-mon-erspan-src-dst)#mtu 146
Router(config-mon-erspan-src-dst)#ip address 169.254.1.2
```

```
Router(config-mon-erspan-src-dst)#origin ip address 169.254.1.1
Router#show monitor session erspan-source
Session 1
---------
Type : ERSPAN Source Session
Status : Admin Enabled
Source Ports :
Both : Ce0/1/0
Destination IP Address : 169.254.1.2
MTU : 1464
Destination ERSPAN ID : 1
Origin IP Address : 169.254.1.1
```

For detailed information on configuring ERSPAN, see Configuring ERSPAN.

New Features for Cisco IOS-XE 17.4.1

- Cyber Vision Support, on page 437
- Installing CVC Sensor using LM GUI, on page 444
- DSL using a SFP
- Out Of Band Management

New Features for Cisco IOS-XE 17.3.1

- Yang Support for IO Ports, on page 500
- Support for Security-Enhanced Linux (SELinux), on page 466
- Initial Bootup Security Improvements, on page 130

Support Added for the P-LTEAP18-GL Modem PID

- The P-LTEAP18-GL PID uses the Telit modem LM960 modem. Details about all of the IR1101 modems are documented in the Cisco Catalyst IR1101 Rugged Series Router Hardware Installation Guide

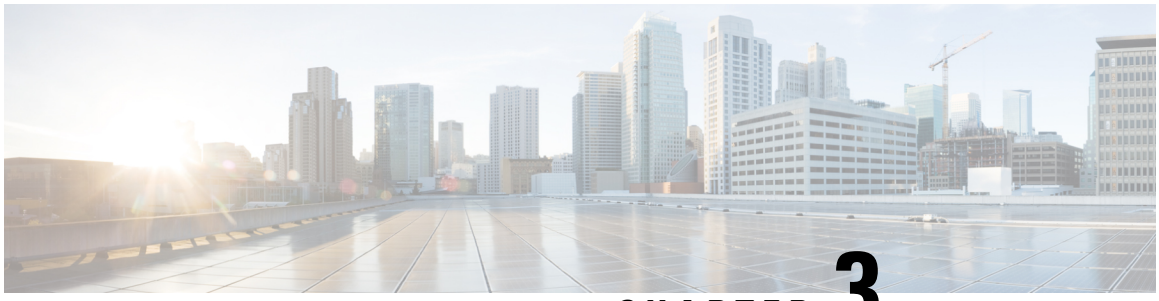
New Features for Cisco IOS-XE 17.2.1

- Native docker support, on page 345
- Yang Data Model Support for Raw Socket Transport, on page 285
- Digital IO for IOx container applications, on page 346
- L2 Sticky Secure MAC Addresses, on page 145
- Signed application support, on page 347

New Features for Cisco IOS-XE 17.1.1

New features in this release are listed below:

- Support for the X25 over TCP (XOT)
- Support for YANG Data Models (Call-home)
- Yang Data Model Support for Scada
- Support for Model Driven support for GNMI Telemetry Dial-In
- Option to Enable or Disable USB Access
- Day 0 Web User Interface



CHAPTER 3

Basic Router CLI Configuration

This chapter contains the following sections:

- IR1101 Interface Naming, on page 21
- Basic Configuration, on page 22
- Configuring Global Parameters, on page 26
- Configuring the Gigabit Ethernet Interface, on page 27
- Support for sub-interface on GigabitEthernet0/0/0, on page 28
- Configuring a Loopback Interface, on page 28
- Enabling Cisco Discovery Protocol, on page 29
- Configuring Command-Line Access, on page 30
- Configuring Static Routes, on page 31
- Configuring Dynamic Routes, on page 33
- Configuring the Serial Interface, on page 34

IR1101 Interface Naming

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

Hardware Interface	Naming Convention
Gigabit Ethernet combo port	gigabitethernet 0/0/0
Fast Ethernet ports	fastethernet0/0/1-0/0/4
Cellular Interface	cellular 0/1/0 and cellular 0/1/1
Asynchronous Serial Interface	async 0/2/0
USB	usbflash0:
mSATA	msata
IR1101 Base Unit Alarm input	alarm contact 0

Note The IRM expansion modules do not apply to IR1001-G2 Rugged Secure Routers.

Interface names for the different expansion modules are found in the following chapters:

- IRM-1100-4A2T Expansion Module, on page 305
- IRM-1101 Expansion Module, on page 291

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 : 8079 bytes
!
! Last configuration change at 17:33:19 GMT Tue Jun 25 2019
!
version 16.12
service timestamps debug datetime msec localtime show-timezone
service timestamps log datetime msec localtime show-timezone
service internal
service call-home
platform qfp utilization monitor load 80
no platform punt-keepalive disable-kernel-core
!
hostname IR1101
!
boot-start-marker
boot-end-marker
!
!
no aaa new-model
clock timezone GMT 0 0
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
  no destination transport-method email
!
!
ip name-server 171.70.168.183 198.224.173.135 8.8.8.8
no ip domain lookup
ip domain name cisco.com
!
login on-success log
ipv6 unicast-routing
!
```

```
chat-script lte "" "AT!CALL" TIMEOUT 20 "OK"
chat-script hspa-R7 "" "AT!SCACT=1,1" TIMEOUT 60 "OK"
!
crypto pki trustpoint SLA-TrustPoint
 enrollment pkcs12
 revocation-check crl
!
crypto pki trustpoint TP-self-signed-756885843
 enrollment selfsigned
 subject-name cn=IOS-Self-Signed-Certificate-756885843
 revocation-check none
 rsakeypair TP-self-signed-756885843
!
!
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-756885843
 certificate self-signed 01
  3082032E 30820216 A0030201 02020101 300D0609 2A864886 F70D0101 05050030
  30312E30 2C060355 04031325 494F532D 53656C66 2D536967 6E65642D 43657274
  69666963 6174652D 37353638 38353834 33301E17 0D313930 35333130 30303530
  385A170D 33303031 30313030 30303030 5A303031 2E302C06 03550403 1325494F
  532D5365 6C662D53 69676E65 642D4365 72746966 69636174 652D3735 36383835
  38343330 82012230 0D06092A 864886F7 0D010101 05000382 010F0030 82010A02
  82010100 D2F61742 3B651909 95856431 9BC2CCB7 D4B04861 DD6E0924 4C3E6A51
  8BF2ABD9 5C3A597D 2EE0112C ECA615AA D0297F9E 071B6B5D 9B831332 021E61F4
  2352EEC9 EE70742E 46EFBAFC A03744D8 A22E4DA3 AAF919CC 0A7929A7 3BDB3B17
  C04DA5B9 028DD3EC 992493A6 EA864ED6 354CB3F4 094D3EBF 5307CAA3 192B5759
  E458712D 841A43CD 709D4D9E 72A9DE3E F935A688 59B6F278 65B59EE0 6B72469E
  7B97582A 64E511A6 D81735FF 117CE399 4C2A2973 F5FD407D BCEB62A6 FD7C6B08
  882E0749 ACE5BD44 32634790 3607ADEA 9F319343 4CA76B0D B1DE6A1C AD144548
  E38119E2 8B34F7AC 090C0450 03166B42 8C7C9EA7 5132687F E1F7BF6E B065CD4E
  889F02BB 02030100 01A35330 51300F06 03551D13 0101FF04 05300301 01FF301F
  0603551D 23041830 16801405 77954127 36509205 7025CF4E 84B5D4A2 A3D53730
  1D060355 1D0E0416 04140577 95412736 50920570 25CF4E84 B5D4A2A3 D537300D
  06092A86 4886F70D 01010505 00038201 01004147 49C6A0A9 56F5BD4D 4892AEE0
  22955E06 AF192FA6 868D5556 959ACF05 398F3907 DFE3148B 0E2CFC12 20BEEA05
```

```
  DC23E8D7 A47DB4AE D6CB6665 BCAE7F39 24D010F0 DB8F0E70 5E7C3F73 25AB1783
  1346D540 47BB7E89 2BB1BE4D 16990318 A4612CC5 C7CC9376 7DF1A1F4 C09C0051
  4D950D99 3CC0C65B 0A98859A 3B81E324 BAB34EDF 64CA8C38 184DC796 47DDD9DD
  F71F8D5E D3B7A962 3D0FDE44 012AC034 D0E7F75A DB1BF12A CF23E2F5 6A4FDA14
  A588DCDA 8272CE33 36ABC57A BFF52980 5FFC7C34 4D4307BB AC0C0F18 AA783B9D
  27C61E89 0EC1C6AA 6AB3F73B EF8450FD 782DFC63 038F6A27 456CA32B D3FEDB97
  C8064523 EBB93FF5 8B98B546 44F853E9 0E04
   quit
!
license udi pid IR1101-K9 sn FCW222700KS
diagnostic bootup level minimal
!
spanning-tree extend system-id
memory free low-watermark processor 50357
file prompt quiet
!
!
username cisco privilege 15 password 0 cisco
username lab password 0 lab123
!
redundancy
!
!
controller Cellular 0/1/0
 no lte firmware auto-sim
 lte modem link-recovery disable
!
controller Cellular 0/3/0
!
vlan internal allocation policy ascending
!
!
interface GigabitEthernet0/0/0
 no ip address
 shutdown
!
interface FastEthernet0/0/1
 switchport access vlan 192
 switchport mode access
!
interface FastEthernet0/0/2
 switchport access vlan 172
 switchport mode access
!
interface FastEthernet0/0/3
 switchport access vlan 172
!
interface FastEthernet0/0/4
 switchport mode access
!
interface GigabitEthernet0/0/5
!
interface Cellular0/1/0
 ip address negotiated
 load-interval 30
 dialer in-band
 dialer idle-timeout 0
 dialer watch-group 1
 ipv6 enable
 pulse-time 1
 ip virtual-reassembly
!
interface Cellular0/1/1
 no ip address
```

```
 shutdown
!
interface Cellular0/3/0
 ip address negotiated
 dialer in-band
 dialer idle-timeout 0
 dialer watch-group 2
 ipv6 enable
 pulse-time 1
 ip virtual-reassembly
!
interface Cellular0/3/1
 no ip address
 shutdown
!
interface Vlan1
 ip address 192.168.10.15 255.255.255.0
!
interface Vlan172
 ip address 172.27.167.121 255.255.255.128
!
interface Vlan175
 ip address 175.1.1.1 255.255.255.0
!
interface Async0/2/0
 no ip address
 encapsulation scada
!
ip default-gateway 172.27.167.1
ip forward-protocol nd
!
ip http server
ip http authentication local
ip http secure-server
ip route 0.0.0.0 0.0.0.0 172.27.167.1
ip route 0.0.0.0 0.0.0.0 Cellular0/1/0
ip route 0.0.0.0 0.0.0.0 Cellular0/3/0 253
ip route 8.8.4.0 255.255.255.0 Cellular0/3/0
ip route 171.70.0.0 255.255.0.0 172.27.167.1
ip route 192.1.1.0 255.255.255.0 Cellular0/1/0
ip route 192.168.193.0 255.255.255.0 192.168.10.1
!
!
ip access-list standard 1
 10 permit any
dialer watch-list 1 ip 5.6.7.8 255.255.255.255
dialer watch-list 1 delay route-check initial 60
dialer watch-list 1 delay connect 1
dialer watch-list 2 ip 5.6.7.8 255.255.255.255
dialer watch-list 2 delay route-check initial 60
dialer watch-list 2 delay connect 1
dialer-list 1 protocol ip permit
dialer-list 1 protocol ipv6 permit
ipv6 route ::/0 Cellular0/1/0
!
!
snmp-server community public RO
snmp-server community private RW
snmp-server host 171.70.127.43 version 2c public
snmp-server host 172.27.167.220 version 2c public
snmp-server manager
!
control-plane
!
```

```
line con 0
 exec-timeout 0 0
 stopbits 1
 speed 115200
line 0/0/0
line 0/2/0
line vty 0 4
 exec-timeout 0 0
 password cisco
 login
 transport input none
!
!
end

IR1101#
```

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* **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 IR1101 is Layer 3 (L3). It is possible to configure the interface as a Layer 2 (L2) interface. The Gigabit Ethernet Interface on the IR1101 is a combo port, which means it is a RJ45+SFP connector.

The IRM-1100-SPMI Expansion Module also has an SFP port. The Gigabit Ethernet Interface (GI0/0/5) on the IRM-1100-SPMI is Layer 2 (L2) only. This means you can assign this port to any vlan (switchport acc vlan #) and use the SVI interface. You cannot assign an ip address directly under this port.

The correct connector must be selected, refer to the Cisco Catalyst IR1101 Rugged Series Router 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	**interface GigabitEthernet** *slot/bay/port* **Example:** `Router(config)# interface GigabitEthernet 0/0/0`	Enters the configuration mode for an interface on the router.
Step 2	**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 3	**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 4	**ipv6 unicast-routing** **Example:** `Router (config)# ipv6 unicast-routing`	Enables forwarding of IPv6 unicast data packets.
Step 5	**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 6	**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 release 16.11.1 and beyond 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	**interface** *type number* **Example:** `Router(config)# interface Loopback 0`	Enters configuration mode on the loopback interface.
Step 2	(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 3	(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 4	**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.

Step 7	**password** *password* **Example:**	Specifies a unique password for the virtual terminal line.

	Command or Action	Purpose
	`Router(config-line)# password aldf2ad1`	
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:**	Specifies a static route for the IP packets. (If you are configuring an IPv6 address, use the **ipv6 route** command described below.)

	Command or Action	Purpose
	`Router(config)# ip route 192.10.2.3 255.255.0.0 10.10.10.2`	
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 **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, FastEthernet0
```

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-17/series.html#Configuration.

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.

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.

	Command or Action	Purpose
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/routers/ios/config/17-x/ip-routing/b-ip-routing.html

Configuring the Serial Interface

This section describes configuring serial interface management.

The IR1101 supports asynchronous serial interface protocols used for SCADA, Raw Socket, or reverse Telnet. It has a single serial interface, designated async 0/2/0. The serial interface is DTE only.

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 asynchronous serial interfaces support the following serial encapsulation methods:

- Raw-TCP
- Raw-UDP
- SCADA
- Encapsulation Relay

Command or Action	Purpose
Router(config-if)# encapsulation {raw-tcp | raw-udp | scada}	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.

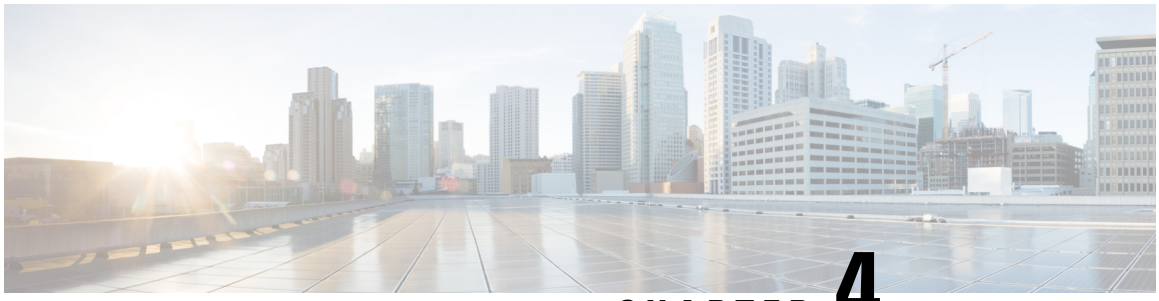
Configuring the Serial Port

To configure the serial port perform the steps in the following example:

```
Router#sh run int async 0/2/0
Building configuration...
Current configuration : 62 bytes
!
interface Async0/2/0
no ip address
encapsulation raw-tcp
end
Router#show line
   Tty Line Typ     Tx/Rx    A Modem  Roty AccO AccI  Uses  Noise Overruns  Int
*     0    0 CTY              -    -      -    -    -     0      0    0/0      -
  0/2/0   50 TTY   9600/9600  -    -      -    -    -     0      0    0/0      -
     74   74 VTY              -    -      -    -    -     3      0    0/0      -
     75   75 VTY              -    -      -    -    -     0      0    0/0      -
     76   76 VTY              -    -      -    -    -     0      0    0/0      -
     77   77 VTY              -    -      -    -    -     0      0    0/0      -
     78   78 VTY              -    -      -    -    -     0      0    0/0      -
```

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

1-49, 51-73, 79-726



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 55
- 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 2: 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 0/1/0


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

Show commands example:

Note The example shows the current GNSS constellation as Galileo.

Example:

```
config#

show cellular 0/1/0 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

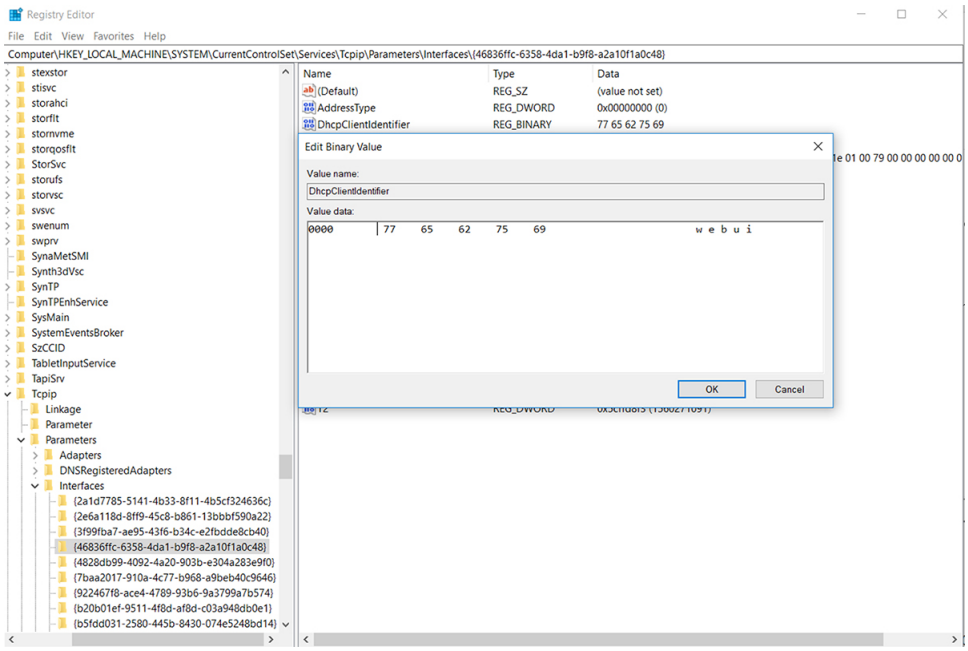
Set up the DHCP Client Identifier on the client to get the IP address from the router, and to be able to authenticate with Day 0 login credentials.

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 3: Setting up DHCP Client Identifier on Windows

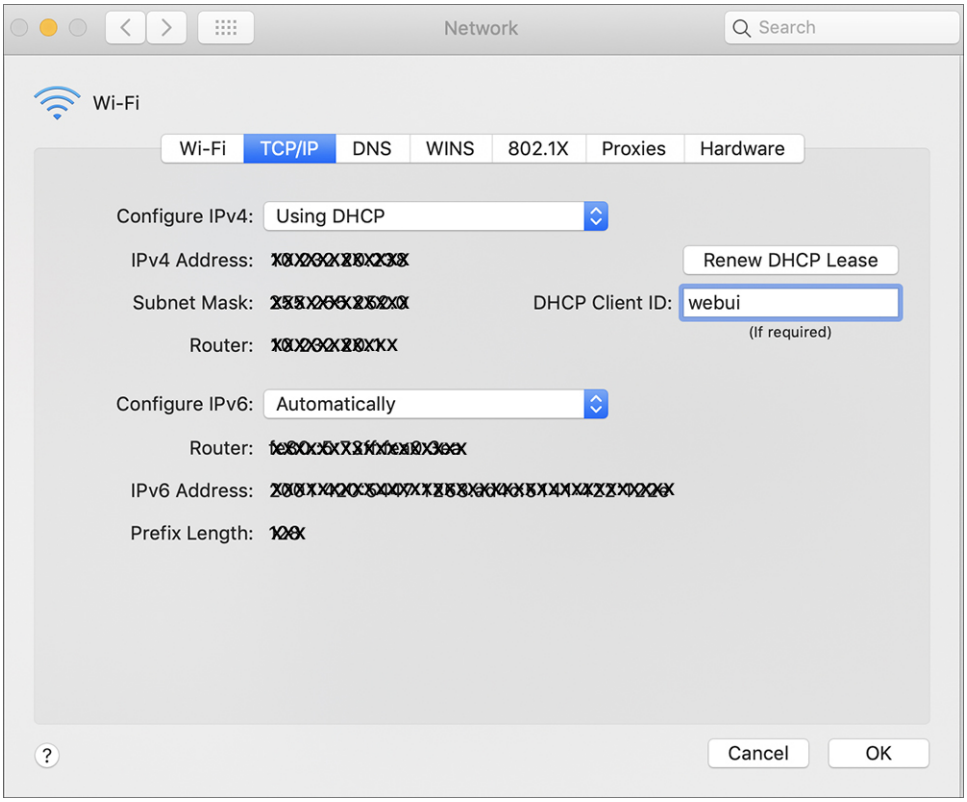


5. Restart the PC for the configuration to take effect.

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 4: Setting up DHCP Client Identifier on MAC



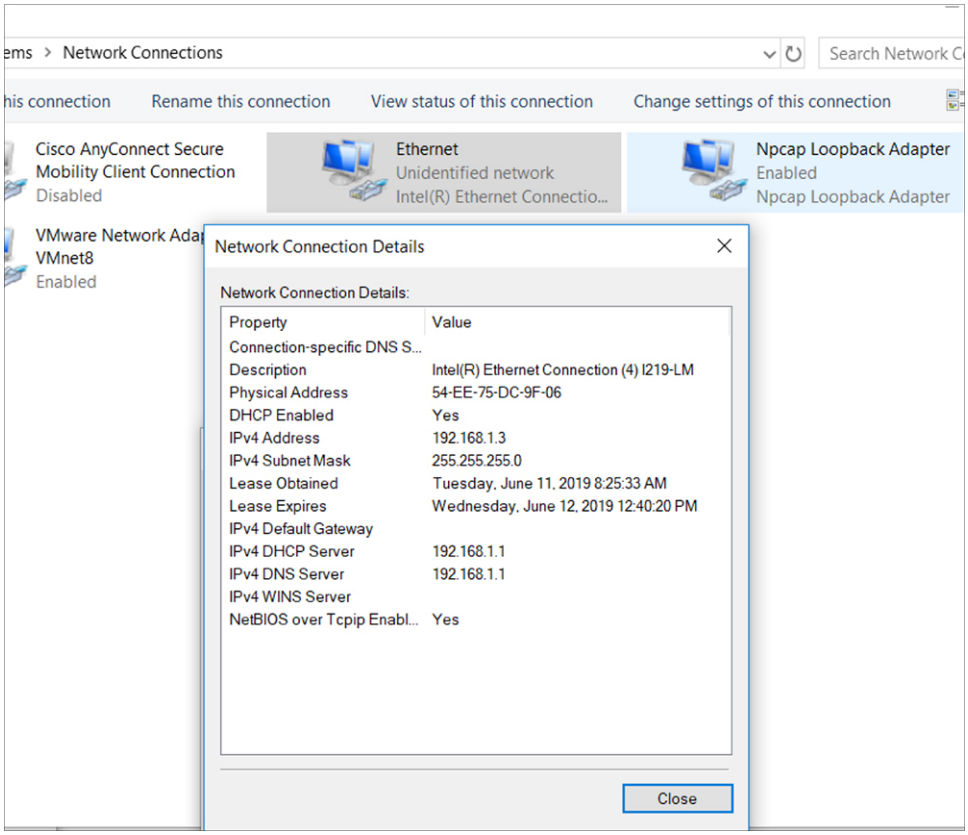
2. Click **OK** to save the changes.

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 5: 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**.

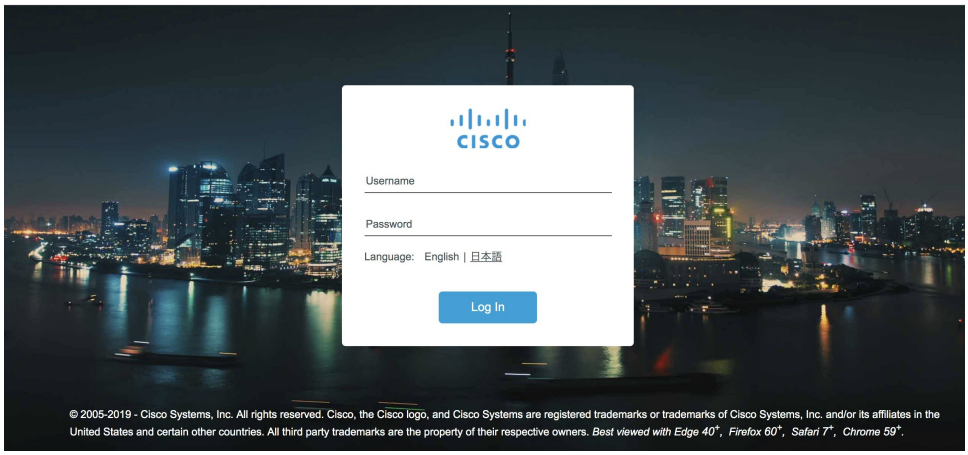
Configuring Basic Mode WebUI through the Browser

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

Procedure

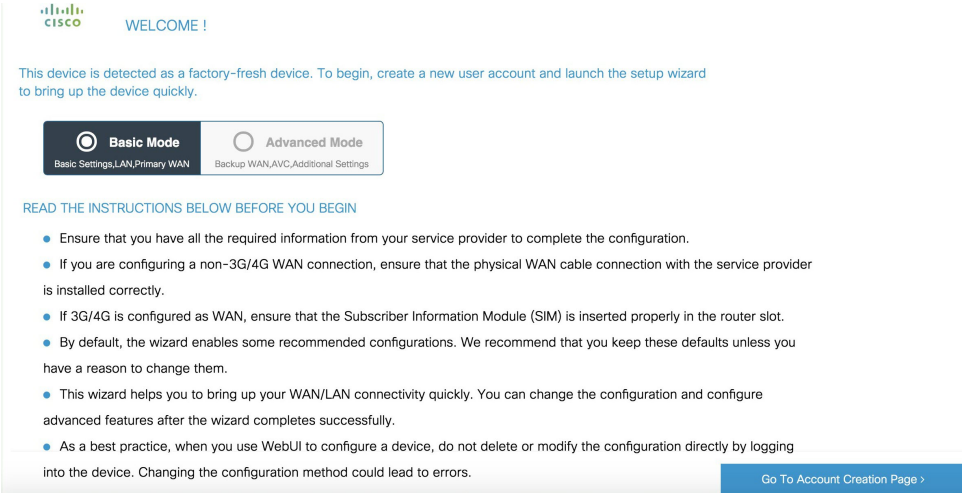
Step 1 Open your browser and enter 192.168.1.1in the address bar. The Login Screen appears. Enter the Username **webui** and the Password **cisco**. Then click **Log In**.

Figure 6: Login Screen



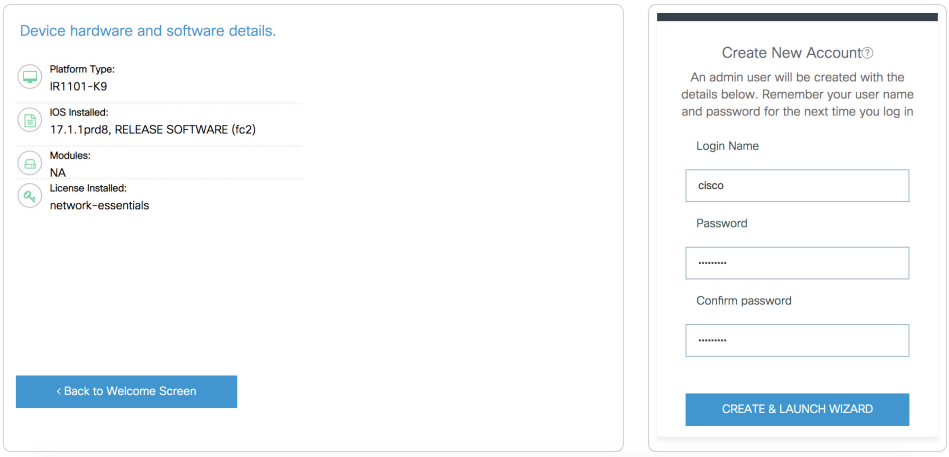
Step 2 The Welcome Screen appears. Select Advanced Mode or Basic Mode. Basic Mode allows for configuring Basic settings, LAN, and a Primary WAN. Advanced Mode allows you to configure an additional Backup WAN, AVC, as well as additional settings. For the purposes of this section, Basic Mode is used. Select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

Figure 7: Welcome Screen



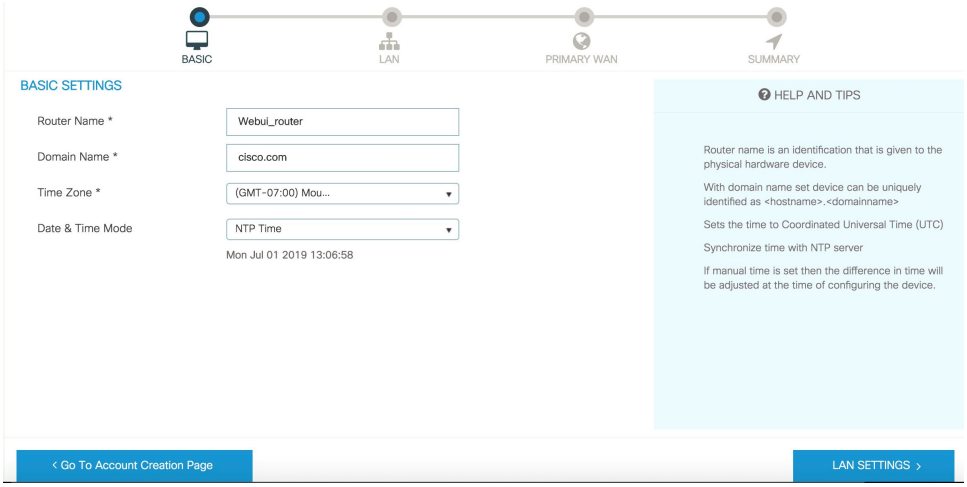
Step 3 Click **Go To Account Creation Page**. The Create New Account Screen appears. Create a new Login Name and Password to access the WebUI.

Figure 8: Create New Account Screen



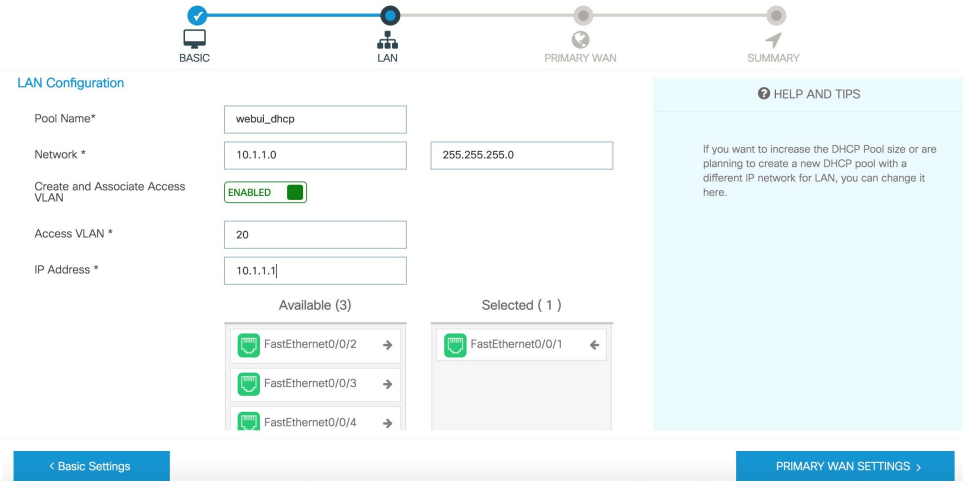
Step 4 Click **CREATE & LAUNCH WIZARD**. The Basic Settings Screen appears. Provide a Router Name (hostname), Domain Name, Time Zone and Date & Time Mode.

Figure 9: BASIC SETTINGS Screen



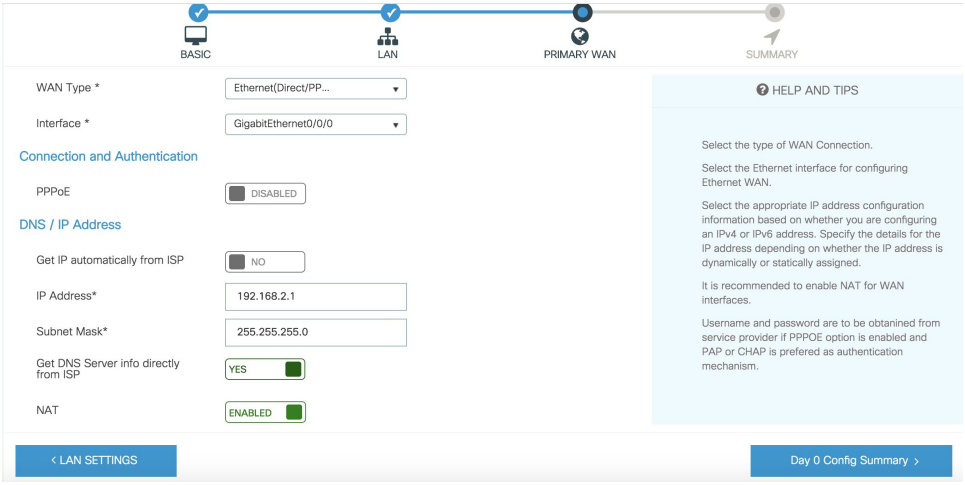
Step 5 Click **LAN SETTINGS**. The LAN Configuration Screen appears. Enter the webui_dhcp Pool Name, VLAN interface IP address, and select the interface that is connected to your laptop from the list of available interfaces.

Figure 10: LAN Configuration Screen



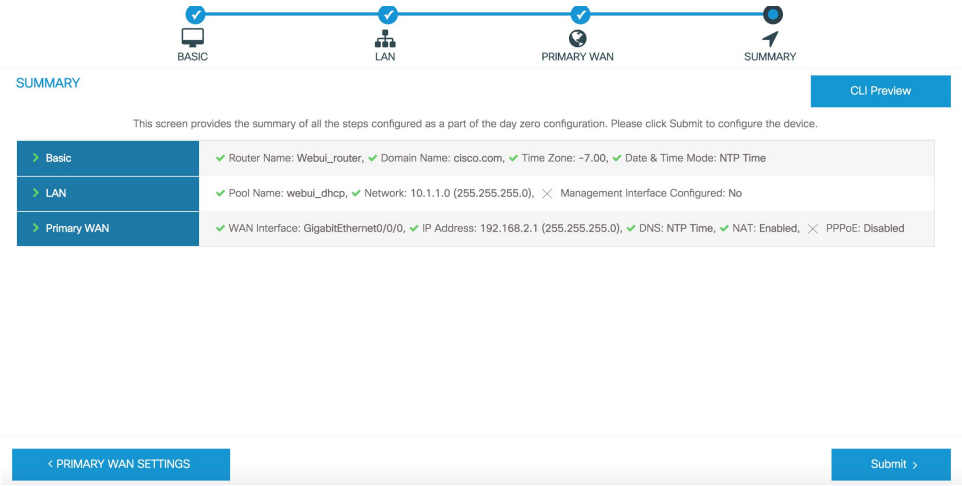
Step 6 Click **PRIMARY WAN SETTINGS**. The PRIMARY WAN SETTINGS Screen appears. Configure the WAN interface by selecting the WAN Type and Interface from the available options. Next enter your DNS IP address information and select Enable/Disable NAT.

Figure 11: Primary WAN Interface Screen



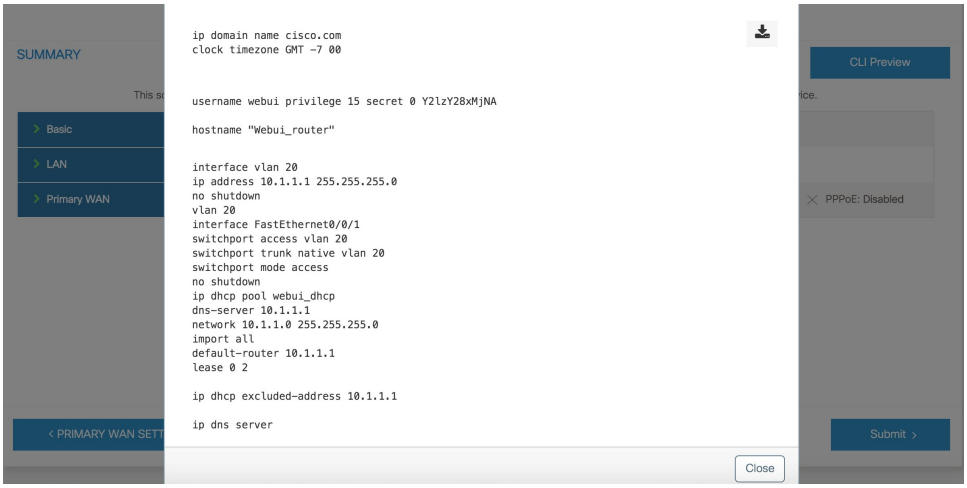
Step 7 Click **Day 0 Config Summary**. The Review Summary Screen appears. Verify your entries before applying the configuration.

Figure 12: Summary Screen



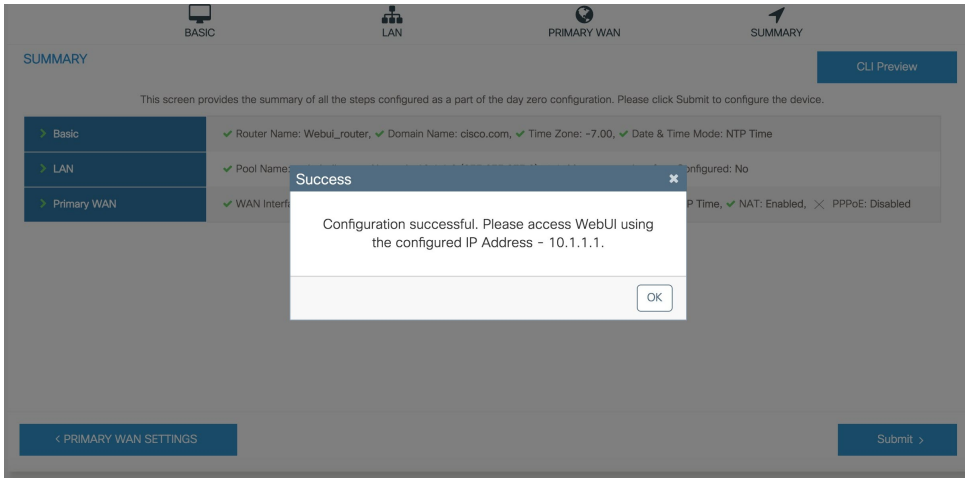
Step 8 (Optional) You can click on **CLI Preview** to see the Configuration that is being applied to the router. Close the CLI Preview and if you are ready, Click **Submit**.

Figure 13: CLI Preview Screen



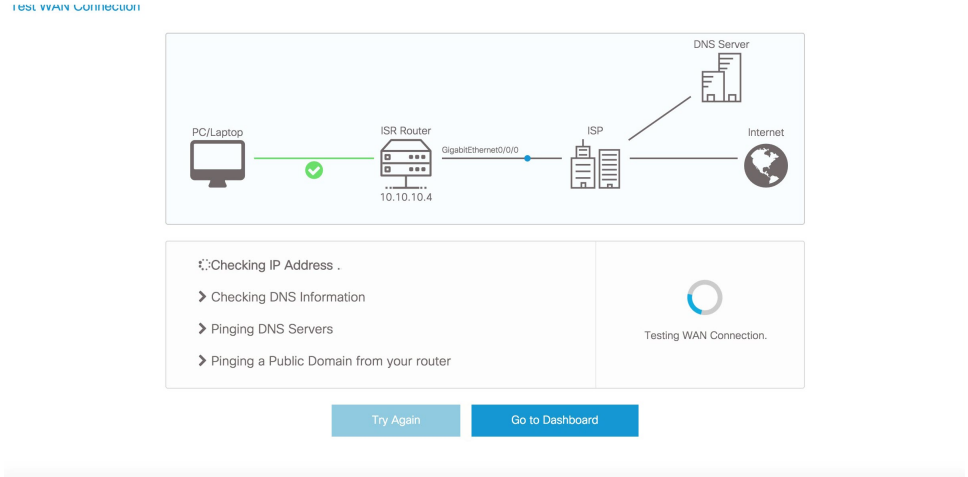
Step 9 After clicking on **Submit**, a dialog box will appear which informs you that the configuration has been applied successfully. The new WebUI ip address is also presented.

Figure 14: 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 15: 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.

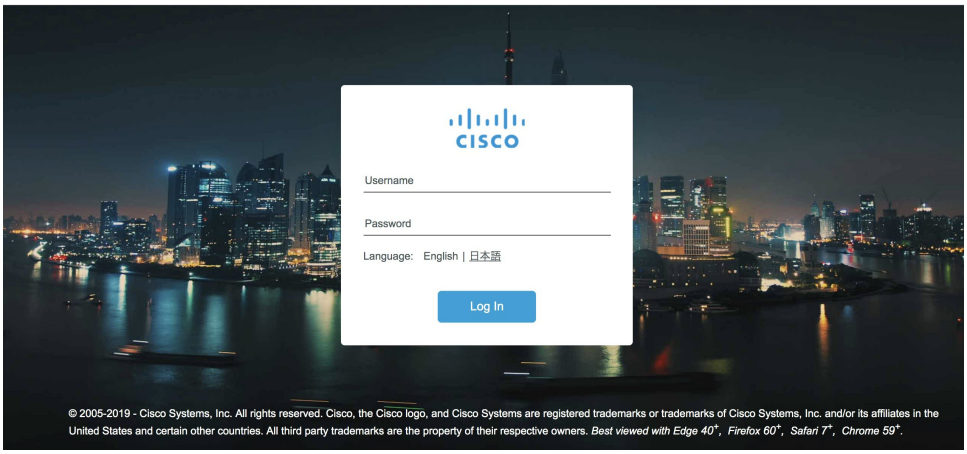
Make sure your laptop is configured to obtain an IP address through DHCP, or assign an IP address *n.n.n.n* matching the default subnet.

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

Procedure

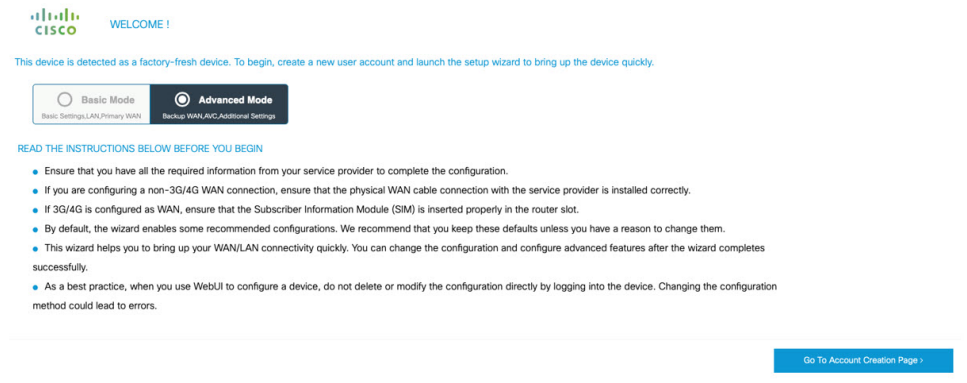
Step 1 Open your browser and enter 192.168.1.1 in the address bar. The Login Screen appears. Enter the Username **webui** and the Password **cisco**. Then click **Log In**.

Figure 16: Login Screen



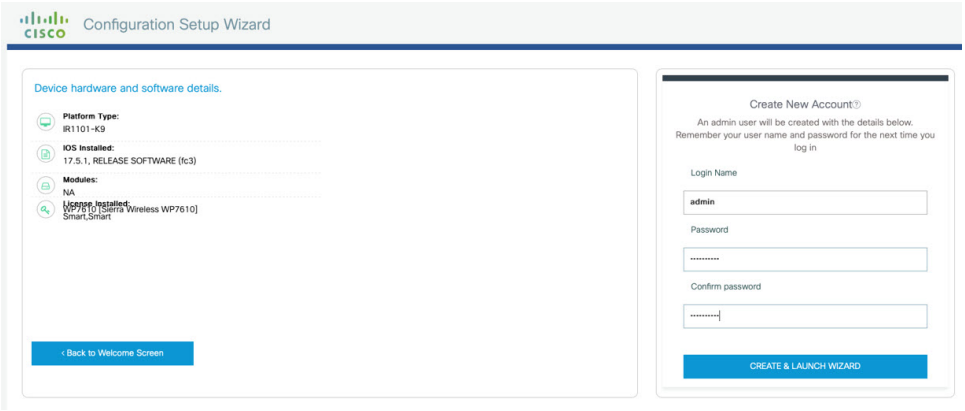
Step 2 The WELCOME screen appears. Select Advanced Mode or Basic Mode. Basic Mode allows for configuring Basic settings, LAN, and a Primary WAN. Advanced Mode allows you to configure an additional Backup WAN, AVC, as well as additional settings. For the purposes of this section, Advanced Mode is used.

Figure 17: WELCOME Screen



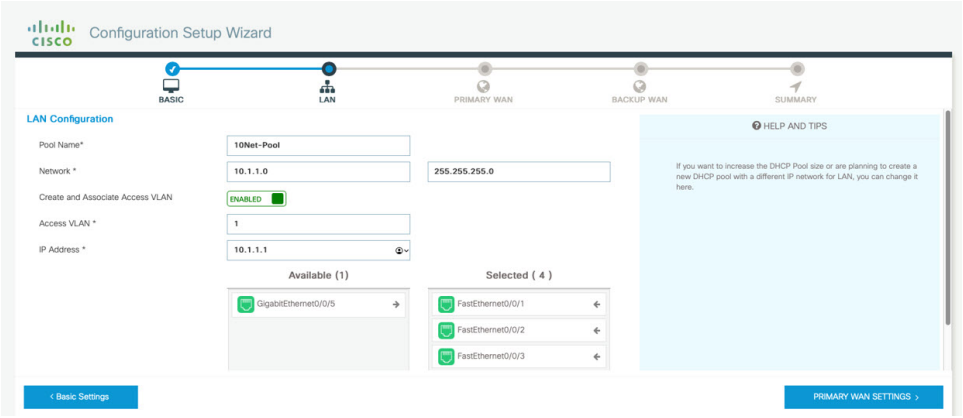
Step 3 Select **Advanced Mode**, then click **Go To Account Creation Page**. The Create New Account screen appears. Create a new Login Name and Password to access the WebUI.

Figure 18: Create New Account Screen



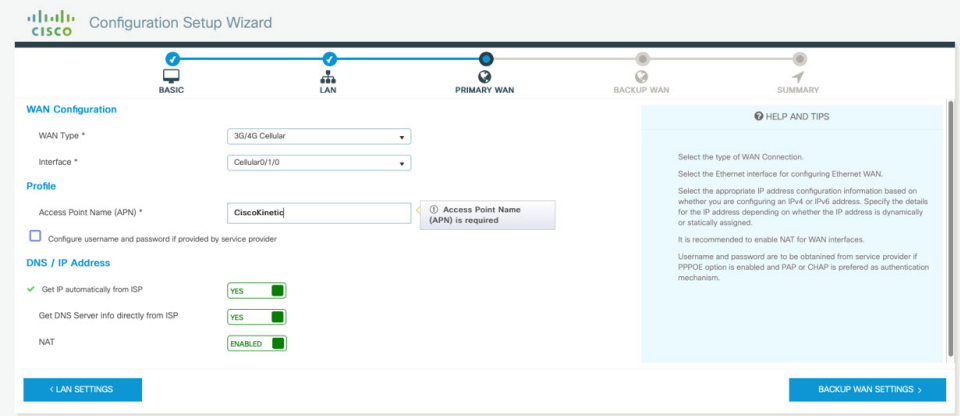
Step 4 Click **CREATE & LAUNCH WIZARD** The LAN Configuration screen appears. Provide a Pool Name, Network IP Address, Subnet, Access VLAN, and Device IP Address. A list of available interfaces is shown to select from.

Figure 19: LAN Configuration Screen



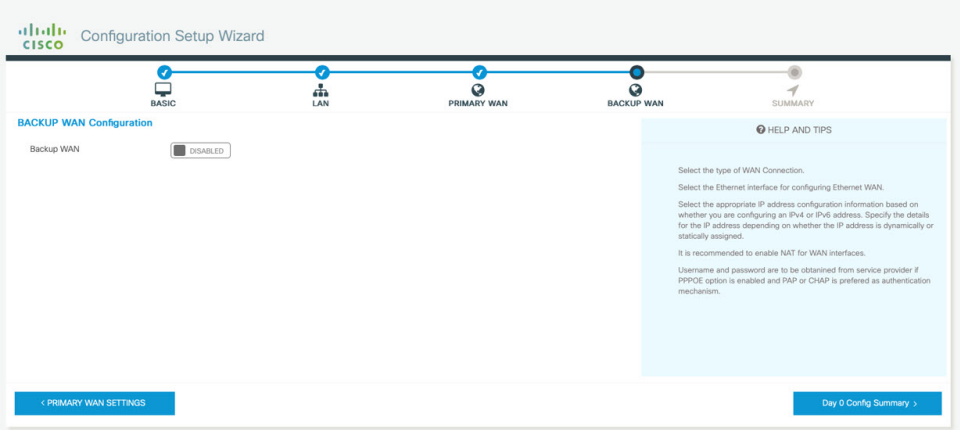
Step 5 Click **PRIMARY WAN SETTINGS**. The WAN Configuration screen appears. Select the WAN Type and Interface from the pull-downs. Provide an APN (Access Point Name) from your LTE Service Provider, and then select the DNS and IP Address settings for your network.

Figure 20: WAN Configuration Screen



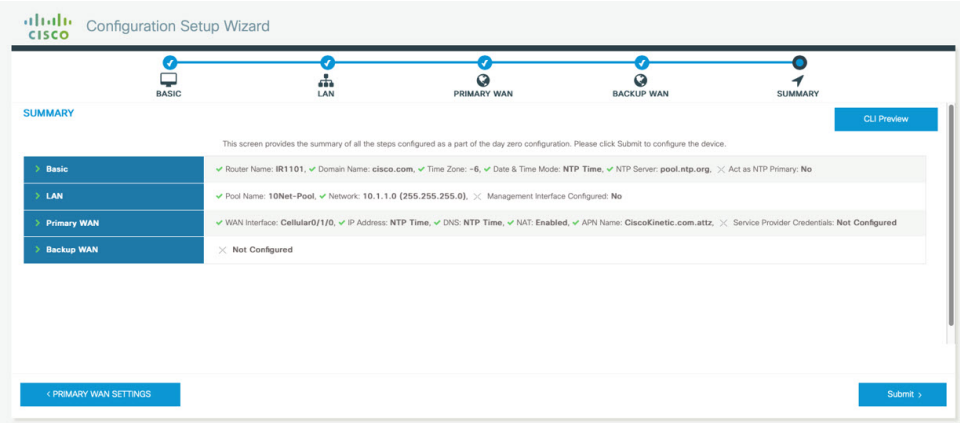
Step 6 Click **BACKUP WAN SETTINGS**. The BACKUP WAN Configuration screen appears. Select the button to Enable or Disable a backup WAN.

Figure 21: BACKUP WAN Configuration



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

Figure 22: Summary Screen

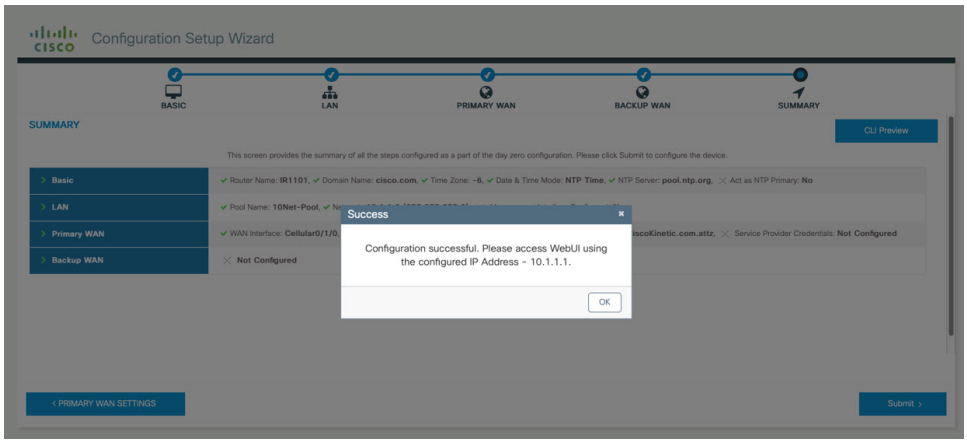


Step 8 (Optional) You can click on **CLI Preview** to see the Configuration that is being applied to the router. Close the CLI Preview, and if you are ready, click **Submit**.

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 23: 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
```

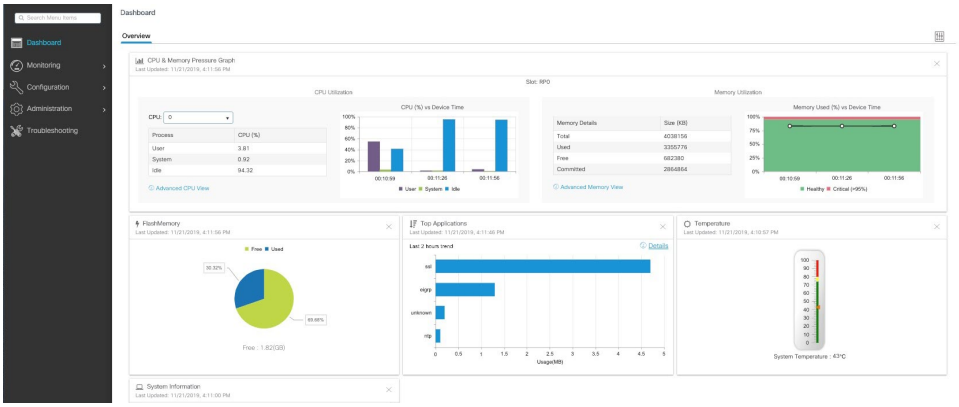
WebUI Dashboard

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 24: 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

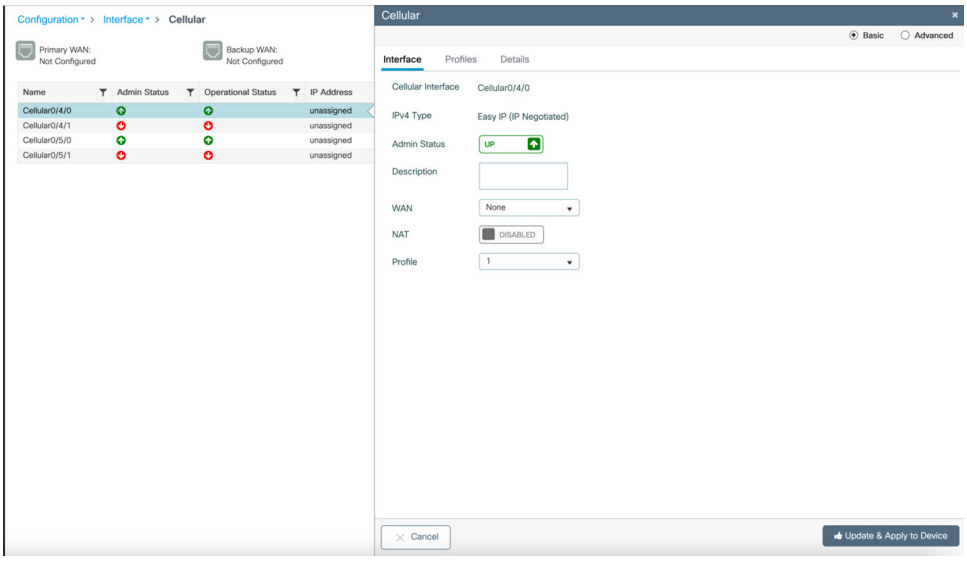
Starting with Cisco IOS XE Release 17.9.1, you can add, edit, or delete the APN from the Cisco WebUI interface. These steps provide an overview of how to perform this function.

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

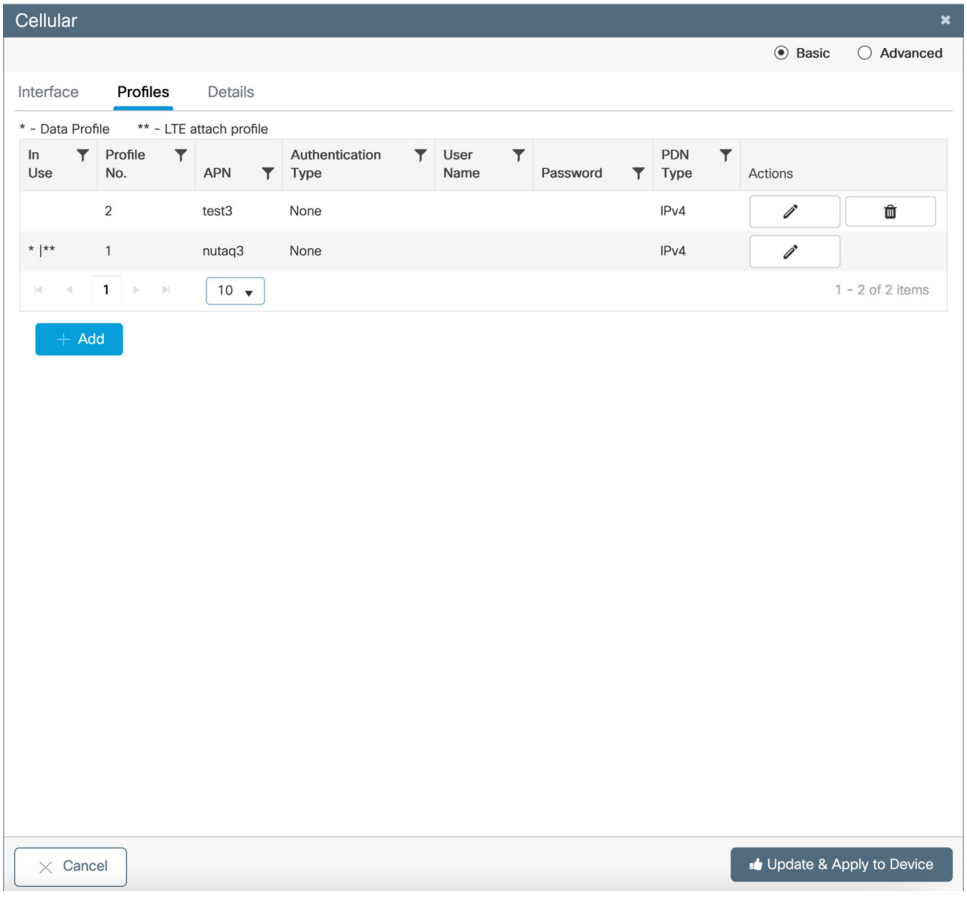
Procedure

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.

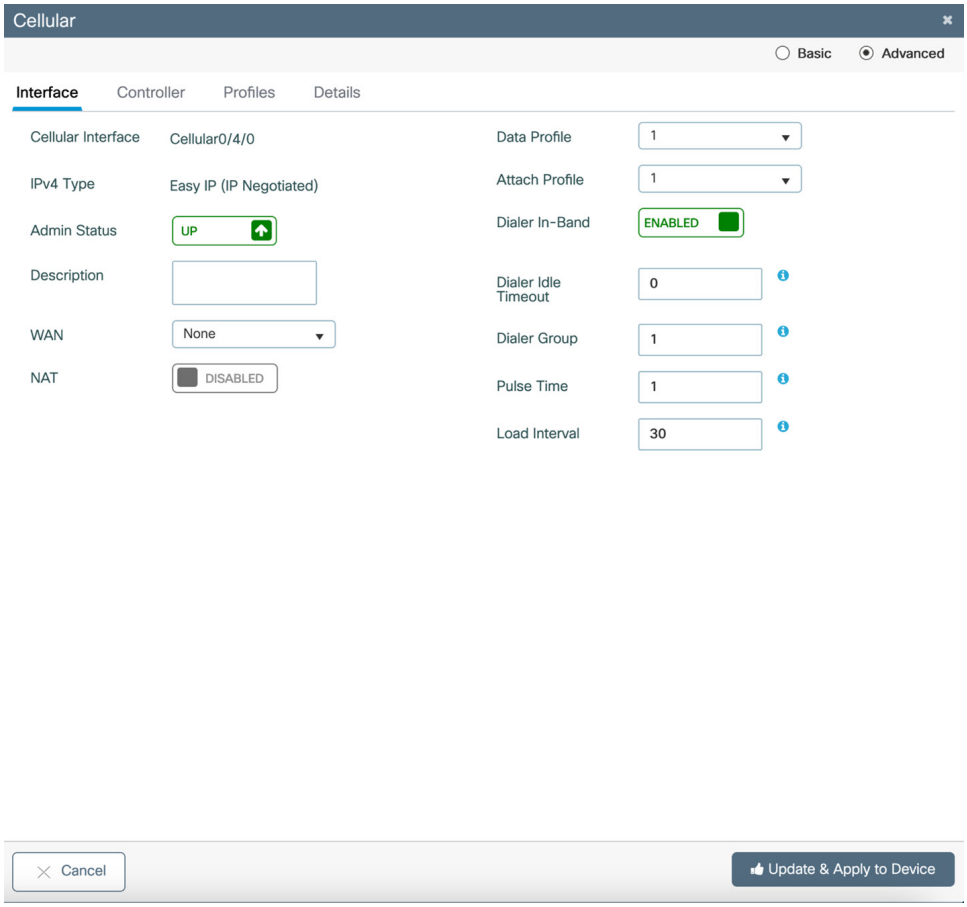


Step 4 From the **Profiles** tab, add, delete, or edit the APN as needed.

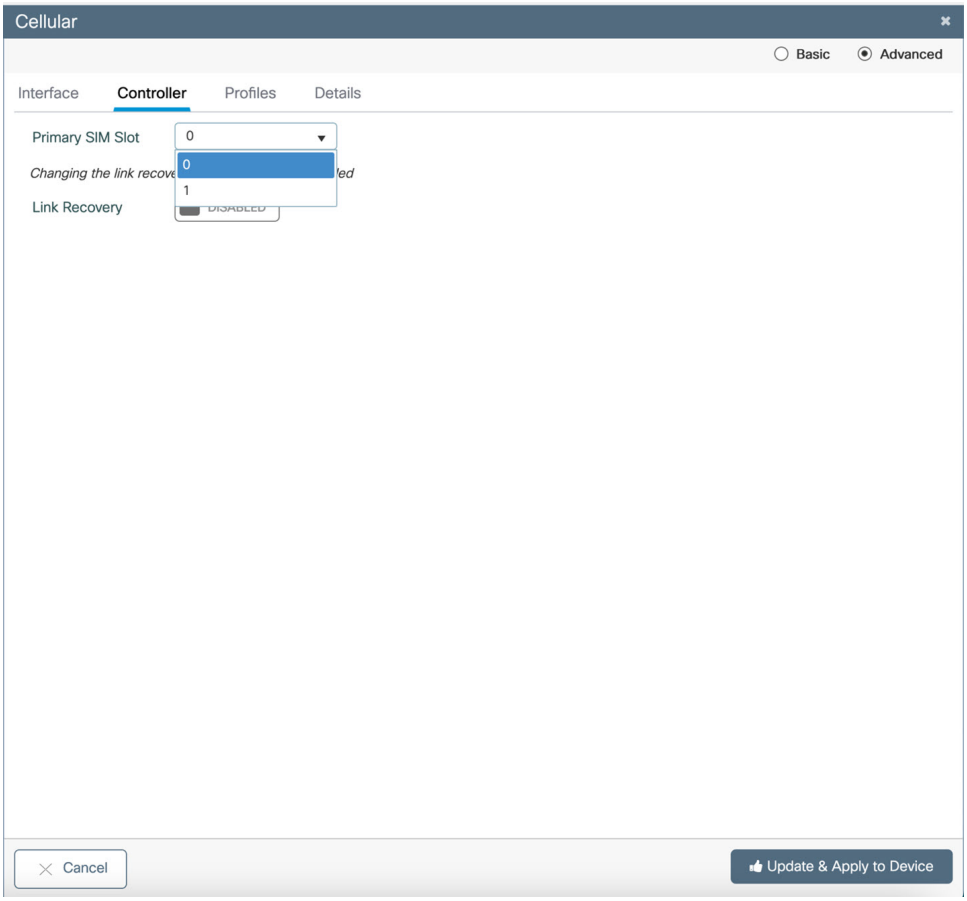
Step 5 Click **Update & Apply to Device** at the bottom of the window.

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.

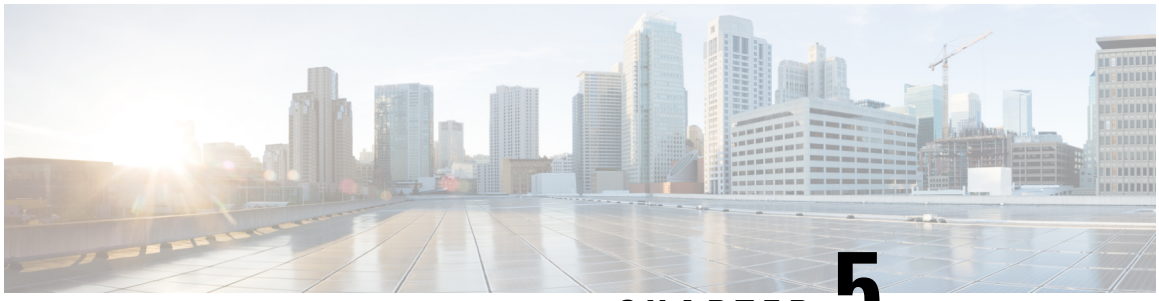


Step 7 Click the **Controller** tab at the top of the window.



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 3: 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:

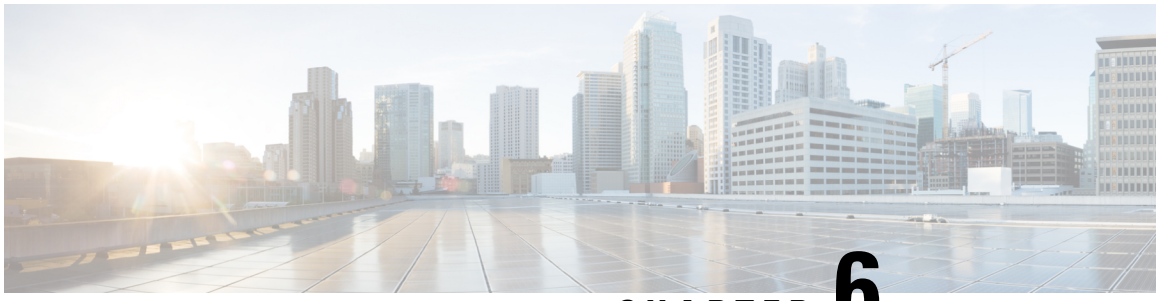
```
Router# show ssh rekey
            SSHv2 server sessions
            Connection  TimeTillRekey  DataLeftTillRekey TimeRekeys  VolumeRekeys
            0           0:00           99997              0           0

            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
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.

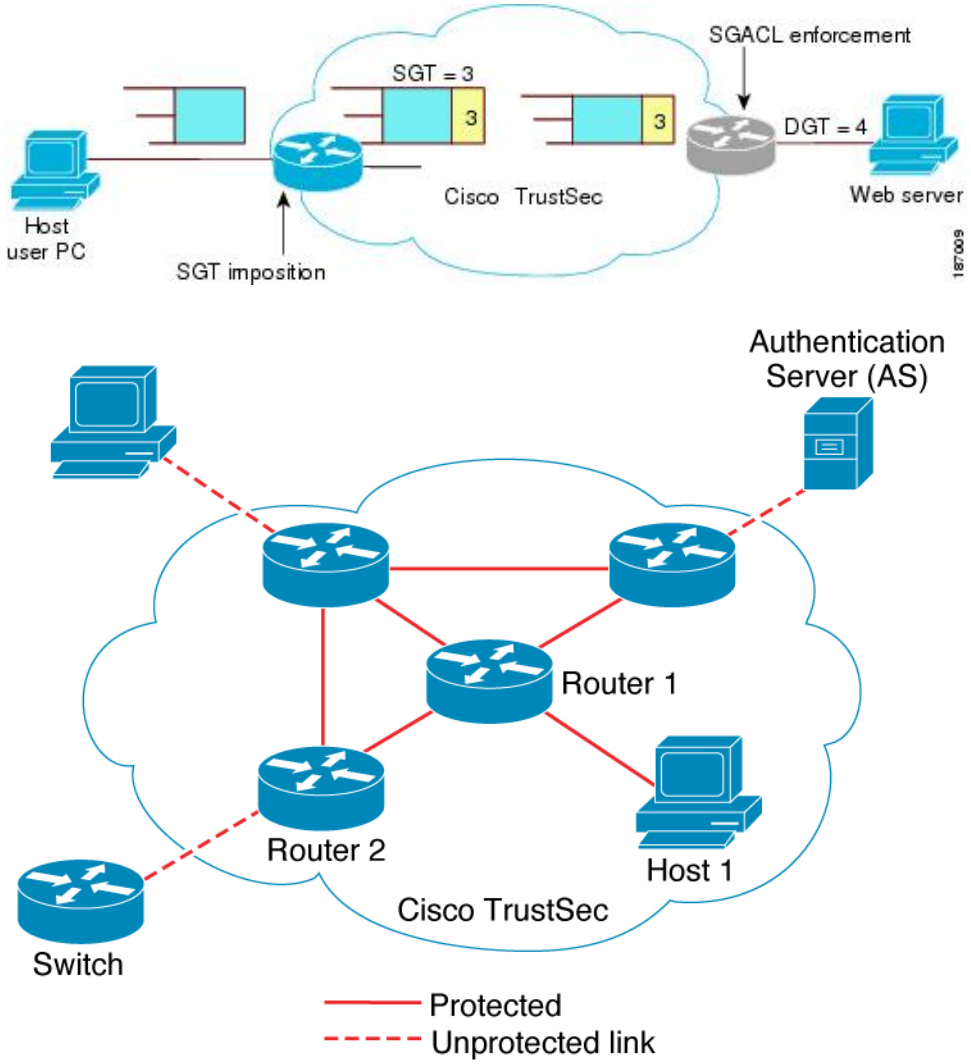
Cisco TrustSec enforces role-based access control using Security Group Tags (SGTs) and Destination Group Tags (DGTs). An SGT is assigned to traffic at the ingress point or can be propagated from a trusted peer device, identifying the security group of the source. At the egress point, the Destination 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 25: 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 does not support Cisco Ethertype CMD 0x8909 frame format).

	Command or Action	Purpose
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

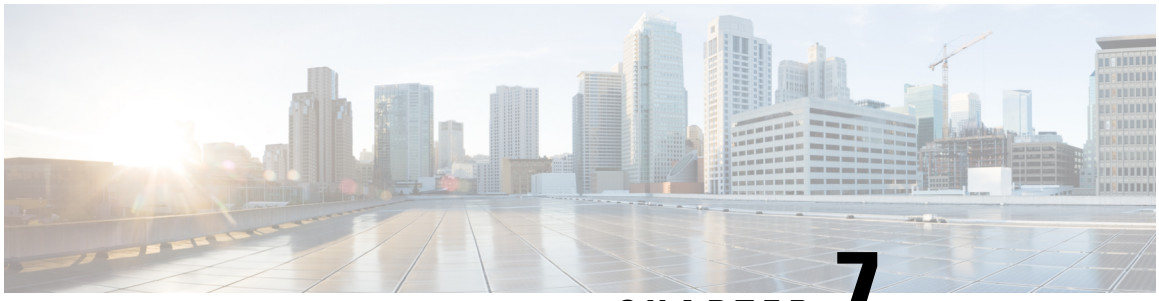
This is an example showing how to enable an interface connecting to a non-trustsec capable endpoint.

```
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.

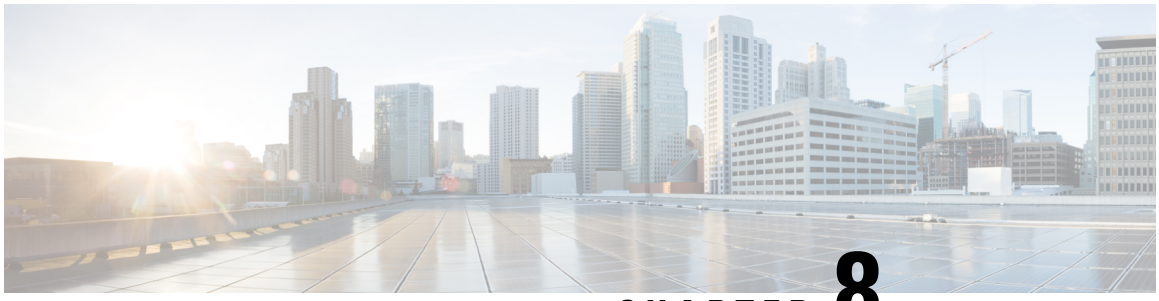
Example:

```
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]

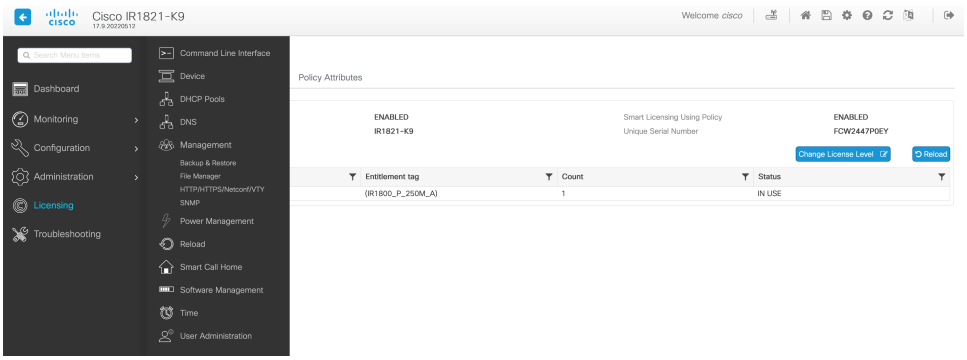
Router# reload
Proceed with reload? [confirm] <enter>
Dec 04 17:42:54.445 R0/0: %PMAN-5-EXITACTION: Process manager is exiting: process exit with
 reload

Initializing Hardware ...
```

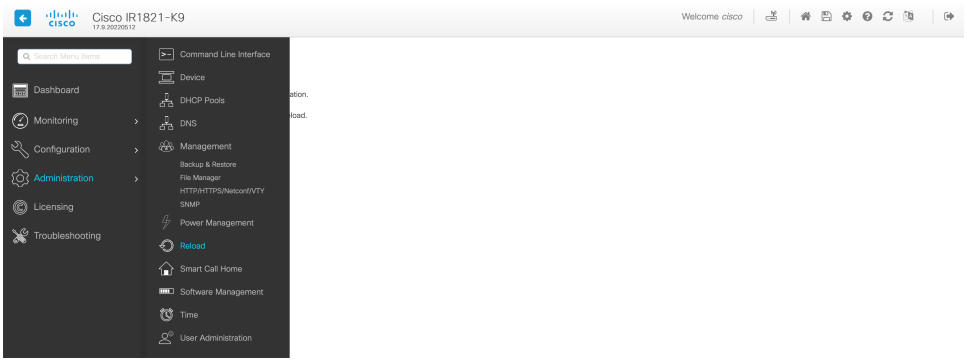
Upgrading the Router Image through the WebUI

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.

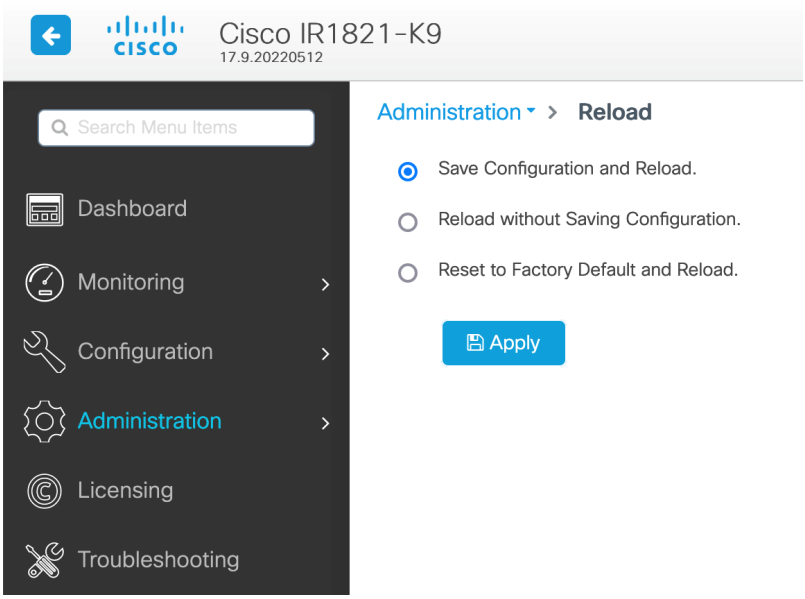
After you launch the WebUI, go to the **Administration** tab.



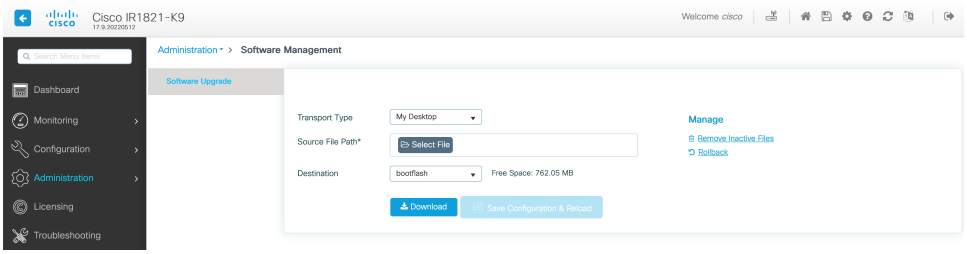
Reload the router by selecting **Administration > Reload**.



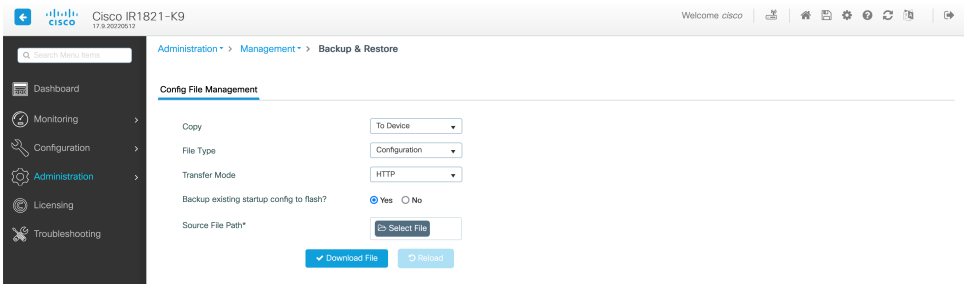
Select your option from the choices, then click **Apply**.



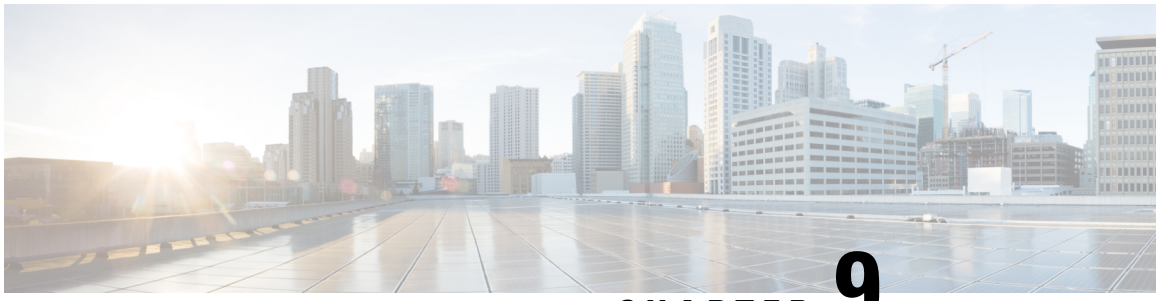
Select **Software Management** under the **Administration** tab. Browse to the location of the new IOS XE image file on your PC.



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 4: 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 5: 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 6: 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 7: 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

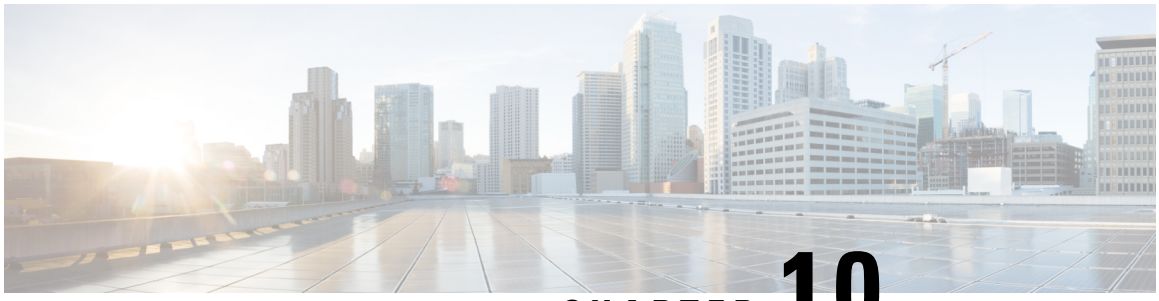
Software release notes provide essential information about the current release. This reference lists the types of information available in release notes and explains how to access cumulative feature information from previous releases.

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 Software Installation Methods

This chapter contains the following sections:

- Install mode support, on page 103
- Bundle mode versus install mode, on page 104
- Installing software using install commands, on page 105
- Restrictions for installing the software using install commands, on page 105
- 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

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 8: 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.

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.

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 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.

Bundle Mode	Install Mode
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.

Installing software using install mode commands

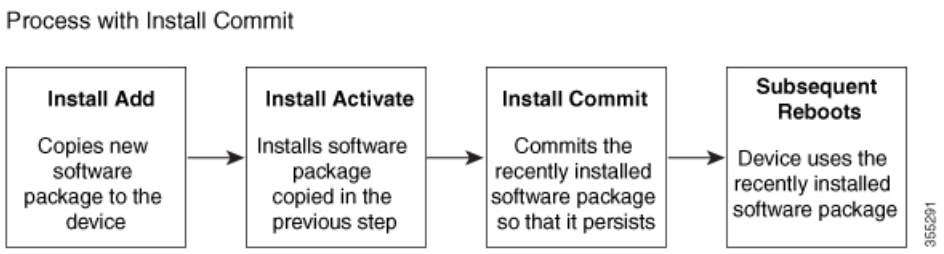
Summary

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 = IR1100
DEVICE_MANAGED_MODE = autonomous
LICENSE_SUITE =
RET_2_RTS =
THRPUT = 250
BOOT = flash:packages.conf,12;
LICENSE_BOOT_LEVEL = network-advantage,all:IR1100;
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:ir1100-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:ir1101-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/ir1101-mono-universalk9.SSA.pkg
/flash/ir1101-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.

IR1101-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:ir1101-universalk9.17.09.01.SPA.bin
```

```
  Package: ir1101-universalk9.17.09.01.SPA.bin
    Size: 674114352
```

```
  Timestamp:
Canonical path: /flash1/ir1101-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: IR1101
  Platform: IR1101
  User: mcpre
  PackageName: universalk9
  Build: 17.09.01
  CardTypes:

Package is bootable from media and tftp.
Package contents:

Package: ir1101-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: IR1101
    Platform: IR1101
    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/ir1101-mono-universalk9_iot.17.09.01.SPA.pkg File is in use, will not
delete.
    [R0]: /flash/ir1101-universalk9.17.09.01.SPA.conf File is in use, will not delete.
    [R0]: /flash/ir1101-rpboot.17.09.01.SPA.pkg File is in use, will not delete.

The following files will be deleted:
    [R0]: /flash/ir1101-universalk9.17.09.01.SPA.bin
    [R0]: /flash/ir1101-mono-universalk9_iot.SSA.pkg
    [R0]: /flash/ir1101-universalk9.SSA.conf
    [R0]: /flash/ir1101-rpboot.SSA.pkg

Do you want to remove the above files? [y/n]y
```

```
Deleting file /flash/ir1101-universalk9.17.09.01.SPA.bin ... done.
Deleting file /flash/ir1101-mono-universalk9_iot.SSA.pkg ... done.
Deleting file /flash/ir1101-universalk9.SSA.conf ... done.
Deleting file /flash/ir1101-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

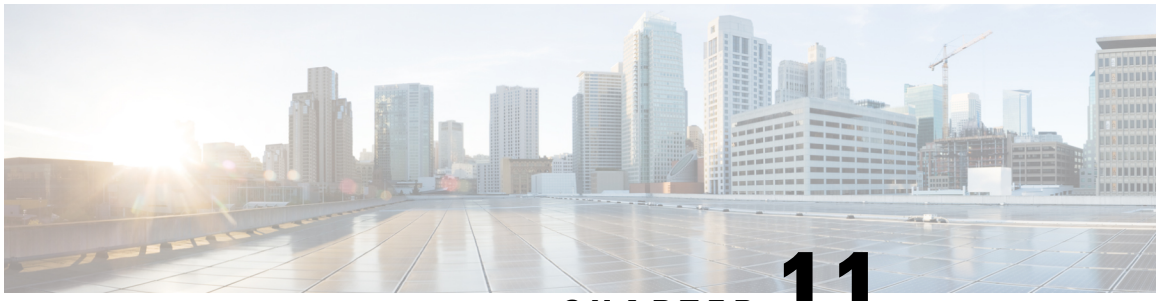
Use these show commands to view installation summary, logs, and software versions.

- **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 124
- Enable the secure data wipe capability, on page 125
- ROMMON Images, on page 126
- File Systems, on page 126
- Option to Enable or Disable USB Access, on page 127
- Autogenerated File Directories and Files, on page 128
- Flash Storage, on page 129
- LED Indicators, on page 129
- Initial Bootup Security Improvements, on page 130
- Related Documentation, on page 130

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.

These are the two main methods to install the software:

- Managing and Configuring a Router to Run Using Consolidated Packages —This method allows for individual upgrade of subpackages and generally has reduced boot times compared to the method below. Use this method if you want to individually upgrade a module's software.
- Managing and Configuring a Router to Run Using Individual Packages —This simple method is similar to a typical Cisco router image installation and management that is supported across Cisco routers.

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 IR1101 uses Smart Licensing, which is discussed in detail in the next chapter.

The IR1101 does not support the Right to Use licenses, and supports only the Specific License Reservation (SLR)

Consolidated Packages

To obtain software images for the router, go to: https://software.cisco.com/download/home/286319772/type

Note All of the IOS-XE feature set may not apply to the IR1101. 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 IR1101 router:

- Network-Essentials
- Network-Advantage

Note Details of the Network-Essentials and Network-Advantage contents can be found in the product data sheet located here:

https://www.cisco.com/c/en/us/products/collateral/routers/1101-industrial-integrated-services-router/datasheet-c78-741709.html

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.

Related Documentation

For further information on software licenses, see the Smart Licensing chapter.

How to Install the Software for Cisco IOS XE

To install the software, use one of the following methods from the Installing the Software, on page 121 chapter.

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 sys bootflash:ir1101-universalk9.16.10.01.SPA.bin
Router(config)#config-register 0x2102
Router(config)#exit
Router#
*Nov  7 00:07:06.784: %SYS-5-CONFIG_I: Configured from console by console
Router#
Router#show run | inc license
license udi pid IR1101-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:/ir1101-universalk9.16.10.01.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:/ir1101-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!
```

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.

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 IR1101 router.

Table 10: 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.

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 IR1101 supports ext2 and vfat file systems for USB flash drives.

The IR1101 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#
```

Files 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
```

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 11: 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 Cisco Catalyst IR1101 Rugged Series Router 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

Custom LED : Off

VPN LED : Off

ALARM LED : Off

GigabitEthernet0/0/0 LED : Off
FastEthernet0/0/1 LED : Off
FastEthernet0/0/2 LED : Off
FastEthernet0/0/3 LED : Off
FastEthernet0/0/4 LED : Off
GigabitEthernet0/0/5 LED : On

EM Module digital I/O 1 LED : Off
EM Module digital I/O 2 LED : Off
EM Module digital I/O 3 LED : Off
EM Module digital I/O 4 LED : Off

*System LTE Pluggable*
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 : On

*EM Module LTE Pluggable*
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 : On
Router#
```

Custom Controlled LED

The IR-1101 has a non-blinking tri-color custom LED, which can be controlled with the following executive privilege CLI.

```
router# set platform hardware custom-led <0-7>
```

The numbers 0-7 are as follows:

- 0: Off
- 1: Blue
- 2: Green
- 3: Red
- 4: Blue/Green
- 5: Blue/Red
- 6: Green Red
- 7: Blue/Green/Red

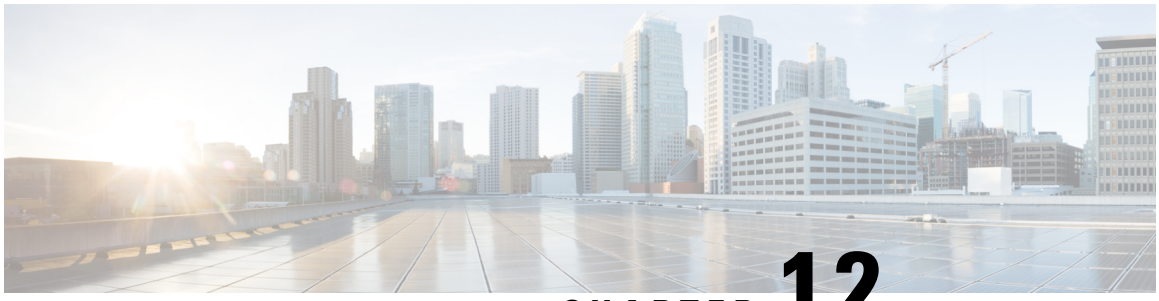
Initial Bootup Security Improvements

This section contains the following:

Related Documentation

For further information on software licenses, see the Smart Licensing Chapter.

For further information on obtaining and installing feature licenses, see Configuring the Cisco IOS Software Activation Feature.



CHAPTER 12

Software Maintenance Upgrade

This chapter contains the following sections:

- Software Maintenance Upgrade (SMU) Overview, on page 131
- SMU Work-flow and Basic Requirements, on page 132
- SMU Example, on page 132
- Installing a Patch Image, on page 133
- Uninstalling the Patch Image, on page 134

Software Maintenance Upgrade (SMU) Overview

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.

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 ir1101-image_name.release_version.CSCxxyyyyy.SPA.smu.bin

SMU Example

This section shows an example of a patch for the CDET CSCvk58743.

Command example:

```
Router# config t
Enter configuration commands, one per line. End with CNTL/Z.
Router(config)# interface g0/0/0
Router(config-if)# ipv6 nd cache expire 770
Router(config-if)# end
Router#
 *Sep 25 12:00:29.978: %SYS-5-CONFIG_I: Configured from console by console
```

As the following CDET states, the ND Cache expire timer did not appear in the command output of **show ipv6 neighbors g0/0/0**

- **CSCvk58743**

Summary: Show ipv6 interface does not display "ND Cache expire timer"

Component: ipv6

Defective Image: ir1101-universalk9.16.11.01.SPA.bin

Patch Image: ir1101-universalk9.16.11.01.CSCvk58743.SPA.smu.bin

The following is what the required configuration output should look like:

```
Interface GigabitEthernet0/0/0
 no switchport
```

```
 no ip address
 ipv6 address FE80::1 link-local
 ipv6 address 2001::1/64
 ipv6 nd na glean

ipv6 nd cache expire 770

 end
```

In the above output, the blue text configures the length of time before an IPv6 neighbor discovery cache entry expires. The range is from 1 to 65536 seconds.

Installing a Patch Image

Perform the following steps to install the patch image:

Procedure

Step 1 Add the image.

```
Router# install add file flash:ir1101-universalk9.16.11.01.CSCvk58743.SPA.smu.bin
install_add: START Mon Dec 17 21:11:23 UTC 2018
install_add: Adding SMU
*Dec 17 21:11:26.241: %INSTALL-5-INSTALL_START_INFO: R0/0: install_engine: Started install add
flash:ir1101-universalk9.16.11.01.CSCvk58743.SPA.smu.bin--- Starting SMU Add operation ---
Performing SMU_ADD on Active/Standby
  [R0] SMU_ADD package(s) on R0
  [R0] Finished SMU_ADD on R0
Checking status of SMU_ADD on [R0]
SMU_ADD: Passed on [R0]
Finished SMU Add operation
SUCCESS: install_add  Mon Dec 17 21:11:39 UTC 2018
```

Step 2 Activate the patch image.

```
Router# install activate file flash:ir1101-universalk9.16.11.01.CSCvk58743.SPA.smu.bin
install_activate: START Mon Dec 17 21:11:57 UTC 2018
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
*Dec 17 21:12:02.086: %SYS-2-PRIVCFG_ENCRYPT: Successfully encrypted private config
fileinstall_activate: Activating SMU
*Dec 17 21:12:05.339: %INSTALL-5-INSTALL_START_INFO: R0/0: install_engine: Started install activate
 flash:ir1101-universalk9.16.11.01.CSCvk58743.SPA.smu.bin
Executing pre scripts....
Executing pre scripts done.
--- Starting SMU Activate operation ---
Performing SMU_ACTIVATE on Active/Standby
 [R0] SMU_ACTIVATE package(s) on R0
  [R0] 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/ir1101-universalk9.16.11.01.CSCvk58743.SPA.smu.bin Mon Dec 17
```

```
21:12:26 UTC 2018
*Dec 17 21:12:25.463: %INSTALL-5-INSTALL_AUTO_ABORT_TIMER_PROGRESS: R0/0: rollback_timer: Install
auto abort timer will expire in 7200 seconds
*Dec 17 21:12:27.358: %INSTALL-5-INSTALL_COMPLETED_INFO: R0/0: install_engine: Completed install
activate SMU flash:ir1101-universalk9.16.11.01.CSCvk58743.SPA.smu.bin
```

Step 3 Commit the installation.

```
Router# install commit
install_commit: START Mon Dec 17 21:13:28 UTC 2018
install_commit: Committing SMU

*Dec 17 21:13:31.516: %INSTALL-5-INSTALL_START_INFO: R0/0: install_engine: Started install commit
Executing pre scripts....
Executing pre scripts 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/ir1101-universalk9.16.11.01.CSCvk58743.SPA.smu.bin Mon Dec 17 21:13:47
 UTC 2018
```

Step 4 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
--------------------------------------------------------------------------------
SMU   C    /flash1/ir1101-universalk9.16.11.01.CSCvk58743.SPA.smu.bin
IMG   C    16.11.1.0.4

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

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:ir1101-image_name.release_version.CSCxxyyyyy.SPA.smu.bin**
  - **install commit**

- **install remove file flash:ir1101-image_name.release_version.CSCxxyyyyy.SPA.smu.bin**

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
--------------------------------------------------------------------------------
SMU   C    /flash1/ir1101-universalk9.16.12.02.CSCvq74407.SPA.smu.bin
IMG   C    16.12.02.0.6
```

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 Fri Apr 24 22:58:25 UTC 2020

*Apr 24 22:58:28.375: %INSTALL-5-INSTALL_START_INFO: R0/0: install_engine: Started install
 rollbackinstall_rollback: Rolling back SMU
Executing pre scripts....
Executing pre sripts done.

--- Starting SMU Rollback operation ---
Performing SMU_ROLLBACK on Active/Standby
  [R0] SMU_ROLLBACK package(s) on R0
  [R0] Finished SMU_ROLLBACK on R0
Checking status of SMU_ROLLBACK on [R0]
SMU_ROLLBACK: Passed on [R0]
Finished SMU Rollback operation

SUCCESS: install_rollback /flash1/ir1101-universalk9.16.12.02.CSCvq74407.SPA.smu.bin Fri
Apr 24 22:58:54 UTC 2020
```

```
*Apr 24 22:58:55.368: %INSTALL-5-INSTALL_COMPLETED_INFO: R0/0: install_engine: Completed
install rollback
```

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    16.12.02.0.6
```



Note In the above command output, the patch has been removed and the device returns to the base image version prior to the upgrade.

Uninstalling the Patch Image Using Deactivate, Commit, and Remove

In the following sequence, there are two patches installed on the device. Only one 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/ir1101-universalk9.16.12.02.CSCvq74407.SPA.smu.bin
SMU   C    /flash1/ir1101-universalk9.16.12.02.CSCvt63576.SPA.smu.bin
IMG   C    16.12.02.0.6
```

Procedure

Step 1 Deactivate the patch.

```
Router# install deactivate file flash:ir1101-universalk9.16.12.02.CSCvt63576.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/ir1101-universalk9.16.12.02.CSCvt63576.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/ir1101-universalk9.16.12.02.CSCvq74407.SPA.smu.bin
SMU   D    /flash1/ir1101-universalk9.16.12.02.CSCvt63576.SPA.smu.bin
IMG   C    16.12.02.0.6
```

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/ir1101-universalk9.16.12.02.CSCvt63576.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/ir1101-universalk9.16.12.02.CSCvq74407.SPA.smu.bin
SMU   I    /flash1/ir1101-universalk9.16.12.02.CSCvt63576.SPA.smu.bin
IMG   C    16.12.02.0.6
```

Step 3 Remove the patch.

```
Router# install remove file flash:ir1101-universalk9.16.12.02.CSCvt63576.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:ir1101-universalk9.16.12.02.CSCvt63576.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

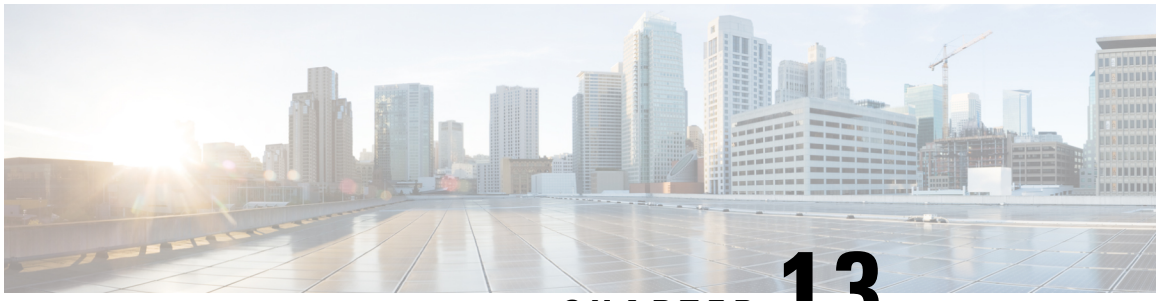
SUCCESS: install_remove /flash1/ir1101-universalk9.16.12.02.CSCvt63576.SPA.smu.bin Fri Apr 24 22:57:34
 UTC 2020

*Apr 24 22:57:34.902: %INSTALL-5-INSTALL_COMPLETED_INFO: R0/0: install_engine: Completed install
remove flash:ir1101-universalk9.16.12.02.CSCvt63576.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/ir1101-universalk9.16.12.02.CSCvq74407.SPA.smu.bin
IMG   C    16.12.02.0.6
```

Note in the above command output the patch for CDET CSCvt63576 has been removed, while the patch for CDET CSCvq74407 remains.



CHAPTER 13

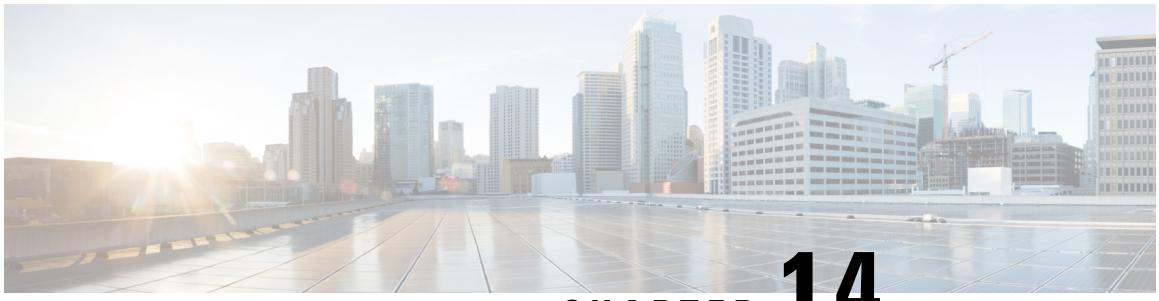
Smart Licensing Using Policy

This reference provides information regarding the implementation and management of Smart Licensing Using Policy.

- Smart licensing using policy on Cisco routers, on page 139

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

Ethernet Switch Ports

This chapter contains the following sections:

- Configuring VLANs, on page 141
- VLAN Trunking Protocol (VTP), on page 142
- Configuring IEEE 802.1X Port-Based Authentication, on page 142
- Configuring Spanning Tree Protocol, on page 143
- Configuring MAC Address Table Manipulation, on page 144
- L2 Sticky Secure MAC Addresses, on page 145
- Configuring Switch Port Analyzer, on page 145
- IGMP Snooping for IPv4, on page 146
- Multi-Homing Configuration in BGP EVPN VXLAN Fabric, on page 146

Configuring VLANs

A VLAN is a switched network that is logically segmented by function or application, without regard to the physical locations of the users. VLANs have the same attributes as physical LANs. However, you can group end-stations even if they are not physically located on the same LAN segment. Any device port can belong to a VLAN, 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 or a device supporting fallback bridging. In a device stack, VLANs can be formed with ports across the stack. Because a VLAN is considered a separate logical network, it contains its own bridge Management Information Base (MIB) information and can support its own implementation of spanning tree.

VLANs are often associated with IP subnetworks. For example, all the end stations in a particular IP subnet belong to the same VLAN. Interface VLAN membership on the device is assigned manually on an interface-by-interface basis. When you assign device interfaces to VLANs by using this method, it is known as interface-based, or static, VLAN membership.

The device can route traffic between VLANs by using device virtual interfaces (SVIs). An SVI must be explicitly configured and assigned an IP address to route traffic between VLANs.

Note IR1101 routers have reserved a set of VLANs 2350 to 2449 for internal platform use. It is not allowed to add the reserved VLANs. You must ensure that these VLANs are not used in the network.

Access Ports

An access port belongs to and carries the traffic of only one VLAN (unless it is configured as a voice VLAN port). Traffic is received and sent in native formats with no VLAN tagging. Traffic arriving on an access port is assumed to belong to the VLAN assigned to the port. If an access port receives a tagged packet IEEE 802.1Q tagged), the packet is dropped, and the source address is not learned.

Trunk Ports

A trunk port carries the traffic of multiple VLANs and by default is a member of all VLANs in the VLAN database. These trunk port types are supported:

- An IEEE 802.1Q trunk port supports simultaneous tagged and untagged traffic. An IEEE 802.1Q trunk port is assigned a default port VLAN ID (PVID), and all untagged traffic travels on the port default PVID. All untagged traffic and tagged traffic with a NULL VLAN ID are assumed to belong to the port default PVID. A packet with a VLAN ID equal to the outgoing port default PVID is sent untagged. All other traffic is sent with a VLAN tag.

Although by default, a trunk port is a member of every VLAN known to the VTP, you can limit VLAN membership by configuring an allowed list of VLANs for each trunk port. The list of allowed VLANs does not affect any other port but the associated trunk port. By default, all possible VLANs (VLAN ID 1 to 4094) are in the allowed list. A trunk port can become a member of a VLAN only if VTP knows of the VLAN and if the VLAN is in the enabled state. If VTP learns of a new, enabled VLAN and the VLAN is in the allowed list for a trunk port, the trunk port automatically becomes a member of that VLAN and traffic is forwarded to and from the trunk port for that VLAN. If VTP learns of a new, enabled VLAN that is not in the allowed list for a trunk port, the port does not become a member of the VLAN, and no traffic for the VLAN is forwarded to or from the port.

For more information on VLANs, see VLAN Configuration Guide, Cisco IOS XE Gibraltar 16.10.x.

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 in Configure VLAN Trunk Protocol (VTP).

Configuring IEEE 802.1X Port-Based Authentication

IEEE 802.1X port-based authentication is configured on a device to prevent unauthorized devices (supplicants) from gaining access to the network. The device can combine the function of a router, switch, and access point, depending on the fixed configuration or installed modules. The switch functions are provided by either built-in

switch ports or a plug-in module with switch ports. This feature supports both access ports and trunk ports. For more information on 802.1X port-based authentication, see the Configuring IEEE 802.1X Port-Based Authentication Guide.

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

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 FastEthernet 0/0/1
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 FastEthernet 0/0/1
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).

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 FastEthernet 0/0/1 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
```

L2 Sticky Secure MAC Addresses

This is a new feature for the IR1101, however, it been present in IOS-XE for some time.

You can configure an interface to convert the dynamic MAC addresses to sticky secure MAC addresses and to add them to the running configuration by enabling sticky learning. The interface converts all the dynamic secure MAC addresses, including those that were dynamically learned before sticky learning was enabled, to sticky secure MAC addresses. All sticky secure MAC addresses are added to the running configuration.

The sticky secure MAC addresses do not automatically become part of the configuration file, which is the startup configuration used each time the switch restarts. If you save the sticky secure MAC addresses in the configuration file, when the switch restarts, the interface does not need to relearn these addresses. If you do not save the sticky secure addresses, they are lost.

Configuring Switch Port Analyzer

The Cisco IR1101 supports local SPAN only, with a limit of one SPAN session. SPAN copies traffic received or sent on source ports to a destination port. You must dedicate the destination port for SPAN use, as it does not receive or forward standard traffic.

Only traffic that enters or leaves source ports can be monitored; traffic routed to a source cannot be monitored.

Procedure

Step 1 Configure a SPAN session to monitor bidirectional traffic from a source interface.

Example:

```
Router# configure terminal
Router(config)# monitor session 1 source FastEthernet 0/0/1
Router(config)# end
```

Step 2 Configure an interface as the destination for a SPAN session.

Example:

```
Router# configure terminal
Router(config)#
monitor session 1 destination FastEthernet 0/0/1
Router(config)# end
```

Step 3 Remove an interface as a SPAN source for a session.

Example:

```
Router# configure terminal
Router(config)# no monitor session 1 source FastEthernet 0/0/1
Router(config)# end
```

What to do next

For detailed information on SPAN configuration, see the Cisco SPAN configuration guide .

IGMP Snooping for IPv4

IGMP snooping allows switches to examine IGMP packets and make forwarding decisions based on their content. You can configure the switch to use IGMP snooping in subnets that receive IGMP queries from either IGMP or the IGMP snooping querier. IGMP snooping constrains IPv4 multicast traffic at Layer 2 by configuring Layer 2 LAN ports dynamically to forward IPv4 multicast traffic only to those ports that want to receive it.

Layer 2 switches can use IGMP snooping to constrain 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. For more information on this feature, see https://www.cisco.com/c/en/us/td/docs/routers/7600/ios/15S/configuration/guide/7600_15_0s_book/snooigmp.html.

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:

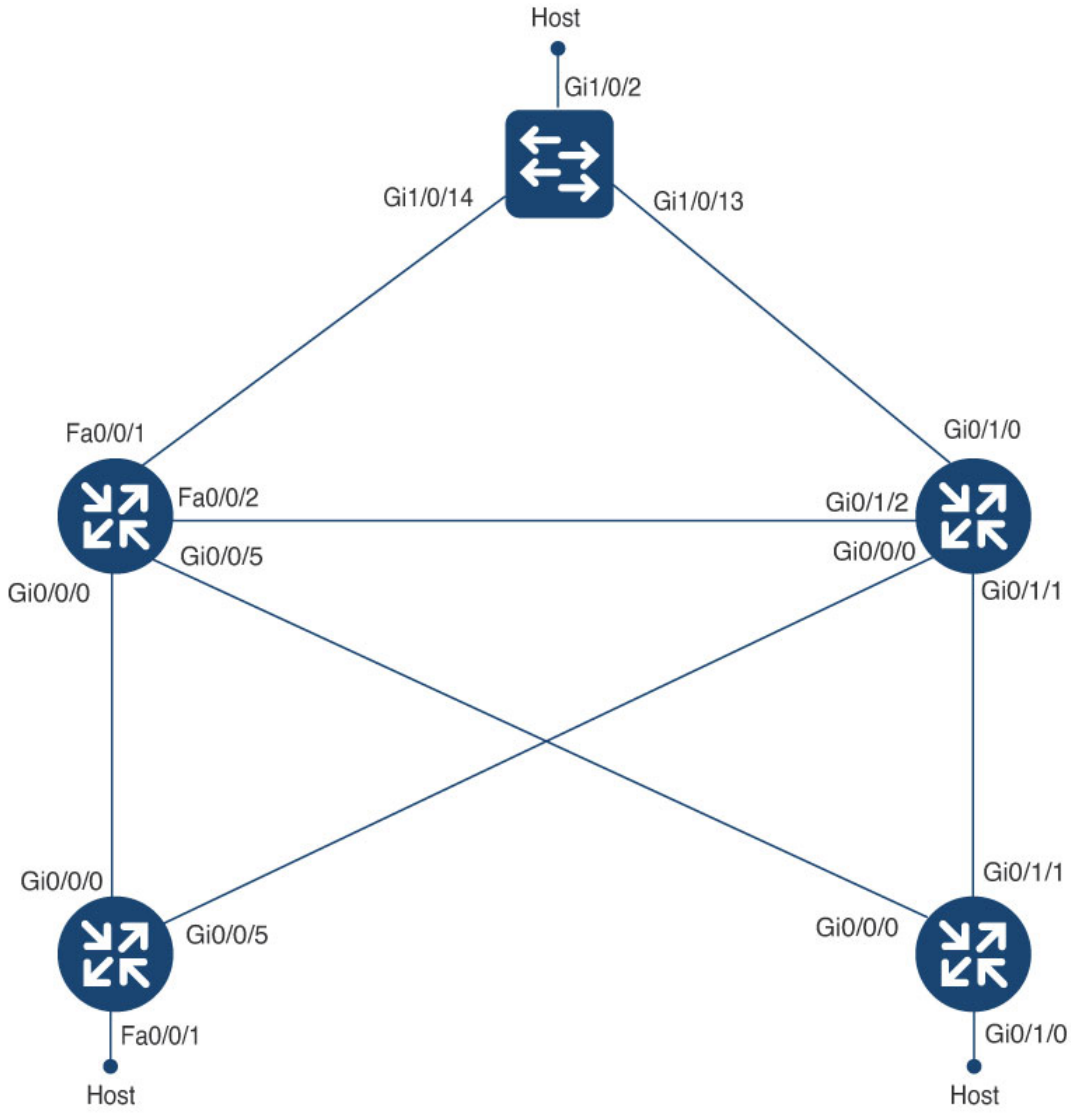
```
Router(config-bdomain)#member evpn-instance 1 vni 20011 ethernet-tag 20011
```

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 26: 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][0300120012001200001][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][0300120012000012000001][4294967295]/23
                     2.2.2.2                  0    100      0 ?
Route Distinguisher: 10.1.2.2:32770
 *>i  [1][10.1.2.2:32770][0300120012000012000001][50200]/23
                     2.2.2.2                  0    100      0 ?
Route Distinguisher: 100.109.165.27:8
 *>   [1][100.109.165.27:8][0300120012000012000001][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][0300120012000012000001][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][0300120012000012000001][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][0300120012000001][4294967295]/23
                      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 ?
Route Distinguisher: 2.2.2.1:4
 *>   [1][2.2.2.1:4][0300120012000001][4294967295]/23
                      0.0.0.0                            32768 ?
Route Distinguisher: 2.2.2.1:32770
 *>   [1][2.2.2.1:32770][0300120012000001][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][0300120012000001][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][0300120012000001][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][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][03001200120012000001][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][03001200120012000001][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][03001200120012000001][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][03001200120012000001][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.

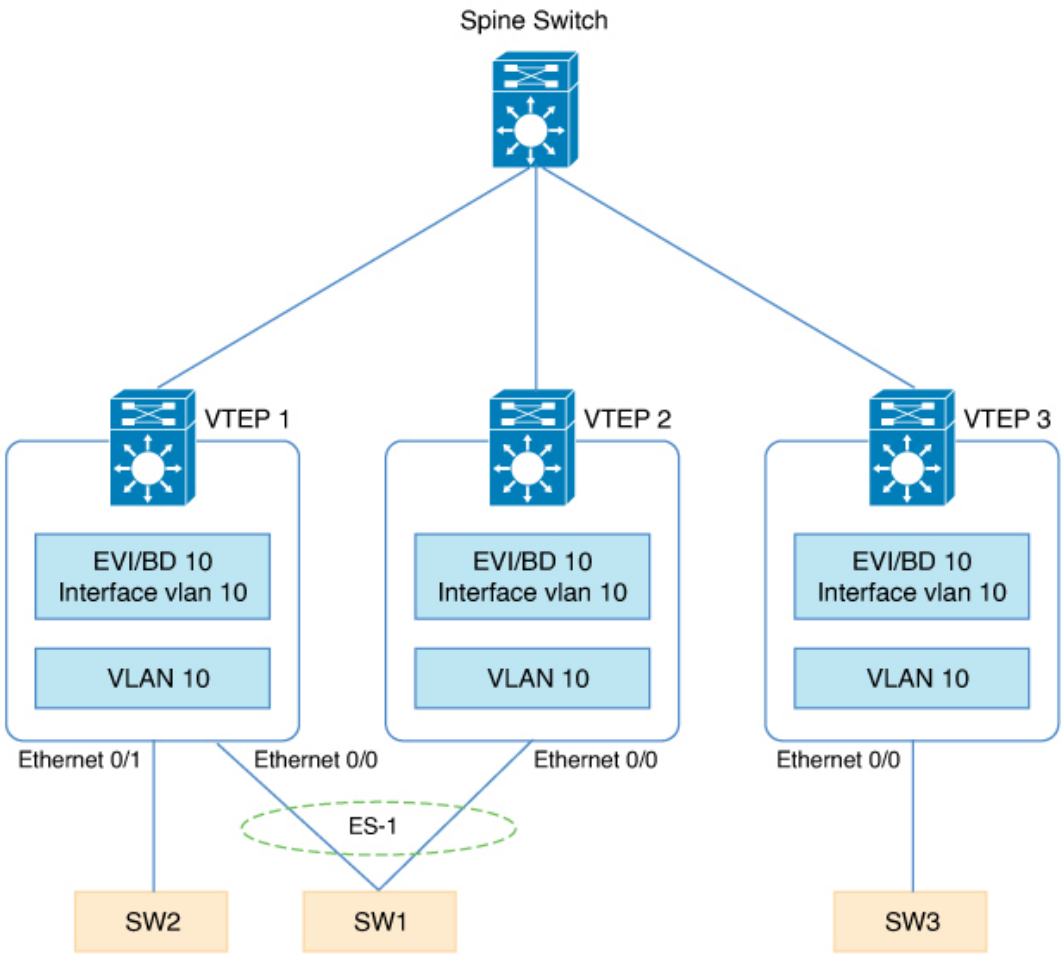
Note Ensure that only one switch port is configured per multihoming VLAN to prevent traffic loss.

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 27: 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.

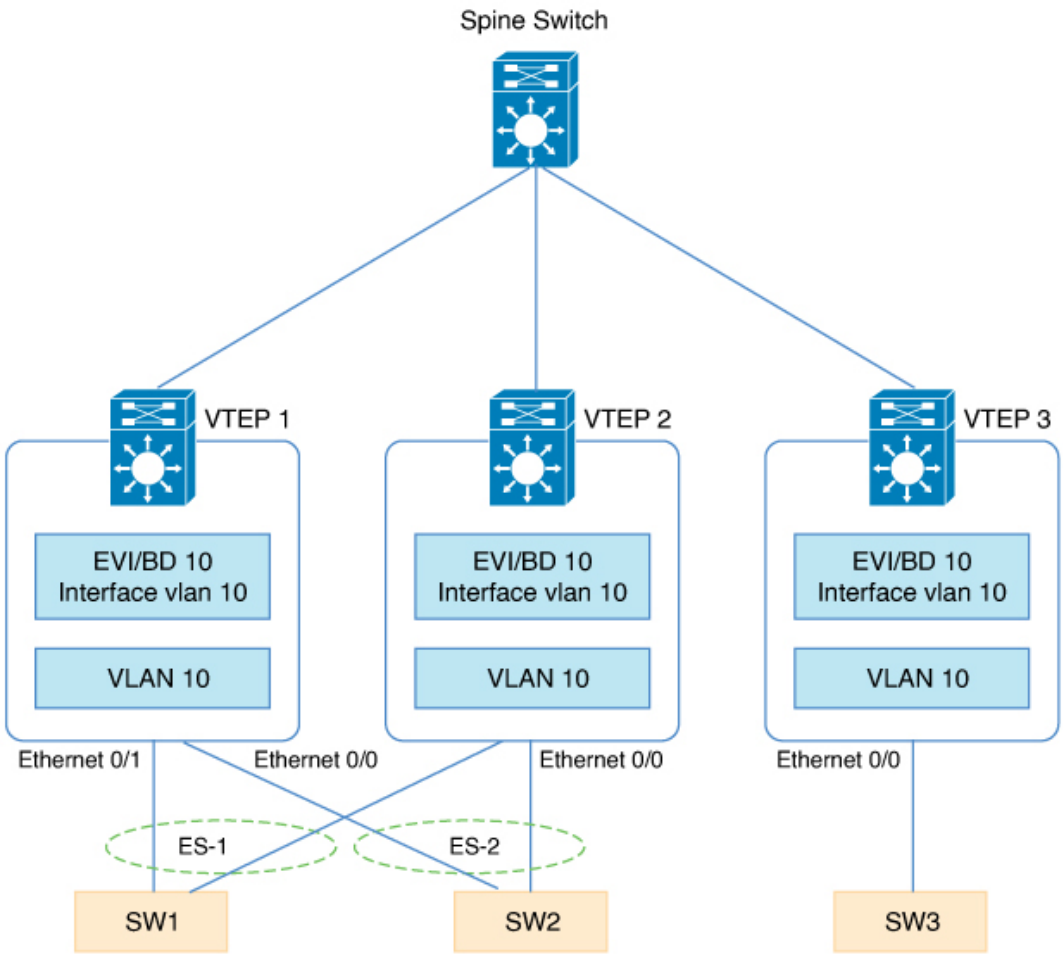
If a Port-channel is used for All-Active multihoming and if it is connected to a multihoming VLAN on a non-designated forwarder (DF) VTEP, the BUM) traffic to a single-homed client switch may be lost.

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 28: 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.

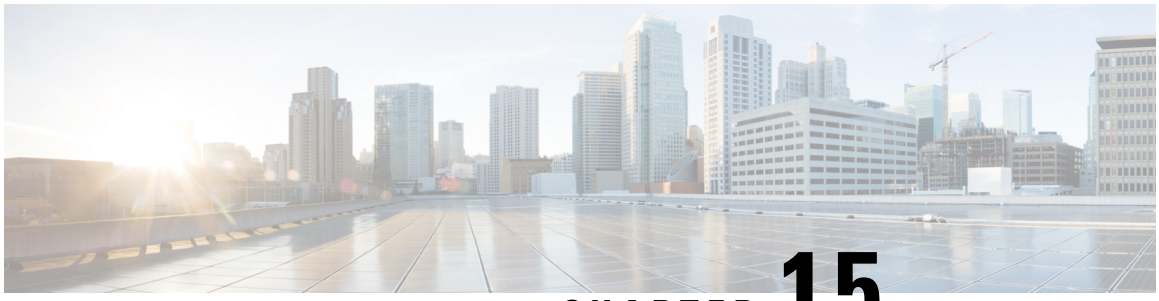
  SW3 receives the BUM traffic forward from EVPN BD 10

- 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 173
- Guidelines and limitations, on page 174
- Options to enable SISF-based device tracking, on page 174
- Configuration and verification examples, on page 177

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 12: 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

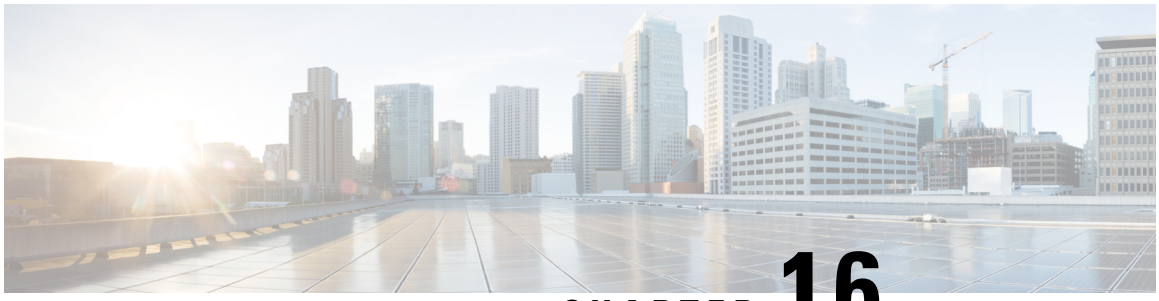
Broadcast/Multicast converted to unicast messages from Gi0/1/2:
Protocol        Protocol message
NDP
DHCPv6
ARP
DHCPv4
ACD&DAD

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

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 179
- Galileo support on the LTE pluggable modules, on page 179
- Radio signal parameters and GPS coordinates of cellular telemetry monitoring using syslog messages, on page 181
- GPS NMEA unique identifier , on page 182
- Configure cellular modem GPS NMEA unique identifier, on page 184

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 0/1/0


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

Show commands example:

Note The example shows the current GNSS constellation as Galileo.

Example:

```
config#

show cellular 0/1/0 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 0/1/0
```

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 0/1/0


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: Cellular0/1/0: 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: Cellular0/1/0: 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: Cellular0/1/0: 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: Cellular0/1/0: 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 13: 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 14: NMEA sentence type and HEX value

HEX value	NMEA sentence type
0x01	GGA
0x02	GSA
0x04	GSV
0x08	RMC
0x10	VTG
0x20	GNS

Configure cellular modem GPS NMEA unique identifier

Follow these steps to configure cellular modem GPS NMEA UID.

Procedure

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

Example:

```
Router# configure terminal
```

Step 2 Use the **controller cellular** *slot* command to enter the controller cellular configuration mode.

Example:

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

Step 3 Use the **lte gps 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)# lte gps nmea uid custom 1428
```

Example:

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

Example:

```
Router(config-controller)# lte gps nmea uid serial-number
```

Step 4 Use the **lte gps nmea ip udp** *source-address destination-address destination-port* **stream** *stream-id* command to specify the source and destination IP address.

Example:

```
Router(config-controller)# lte gps nmea ip udp 10.195.79.179 171.70.55.77 14013 stream 1
```

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)# lte gps nmea filter 28
```

This example shows the HEX value if you want to send sentence type as only GGA (0x01) and GSA (0x02):

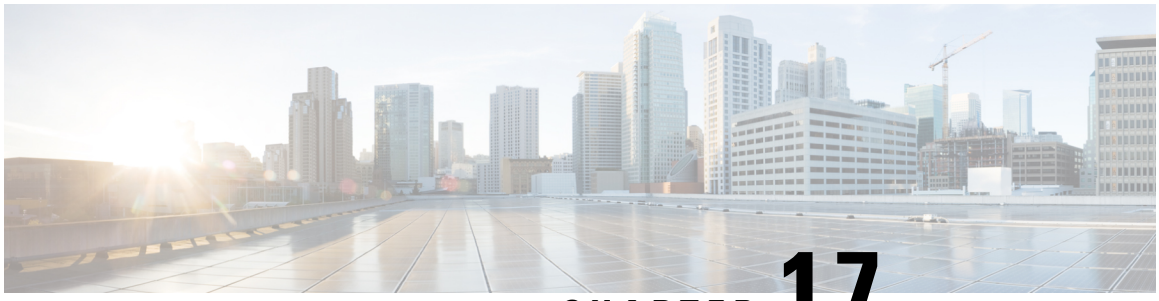
Example:

```
Router(config-controller)# lte gps nmea filter 03
```

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 17

P-LTE 450 Support

- Support for P-LTE-450, on page 187
- Support for P-LTE-450 Pluggable Interface Module, on page 187
- VRF for the P-LTE-450 Module, on page 189
- P-LTE-450 Support on Autonomous Mode, on page 190
- P-LTE-450 Support Over SDWAN/vManage, on page 191
- Enhancing P-LTE-450 Network Management Through SNMP MIB Objects, on page 192
- Upgrade the P-LTE-450 PIM, on page 193

Support for P-LTE-450

The P-LTE-450 is a 450MHz Category-4 LTE PIM, which addresses LTE use cases primarily targeting utility, public safety, and critical infrastructure maintained by public organizations in Europe and other world regions. The module supports only Band 31 and 72 for LTE 450MHz networks.

Note Throughout the user documentation, you will see the module referred to as P-LTE-450, which is the Cisco product name. The module is designed and manufactured by Intelliport, which refers to it as the IPS-701. Both names will be present in documentation.

Unlike regular LTE modules, there are some differences with regards to the P-LTE-450MHz on IOS-XE platform. Some of the key differences are:

- IP pass through will be on Gigabit Ethernet interfaces rather than cellular interface
- Troubleshooting commands are from web interface of third-party hardware

See the Cellular Pluggable Interface Module Configuration Guide for complete details.

For additional information, see the LTE 450MHz Alliance.

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

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 15: 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.

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
```

P-LTE-450 Support on Autonomous Mode

This release introduces two modes of setting the required credentials to communicate with the module. The username and password that should be used in these CLIs can be found on the sticker label that comes with the P-LTE-450 module.

Important You MUST set the username and password before performing any P-LTE-450 parameter configuration.

Configuration

The recommended configuration is through the Config mode:

```
interface GigabitEthernet 0/1/0
lte450 credential username username password password
```

Using the Exec mode:

```
hw-module subslot 0/1 lte450 set-info username username password password [encrypt]
```

Note Execution of this command will create a file called **bootflash:lte450.info** and should not be deleted.

P-LTE-450 Support Over SDWAN/vManage

The P-LTE-450 is a 450MHz Category-4 LTE PIM, which addresses LTE use cases primarily targeting utility, public safety, and critical infrastructure maintained by public organizations in Europe and other world regions. The module supports only Band 31 and 72 for LTE 450MHz networks.

Support for the P-LTE-450 was introduced in IOS XE 17.12.1a. This release introduces support for the P-LTE-450 over SDWAN /vManage.

Guidelines and Limitations

The following are the limitations of the P-LTE-450 with SDWAN/vManage:

- No PNP support on P-LTE-450 as a primary link.
- P-LTE-450 parameter configuration is only supported with CLI templates.
- P-LTE-450 credential configuration via vManage is not supported on this release. Will be supported in the vManage 20.16 release.

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

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
```

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
```

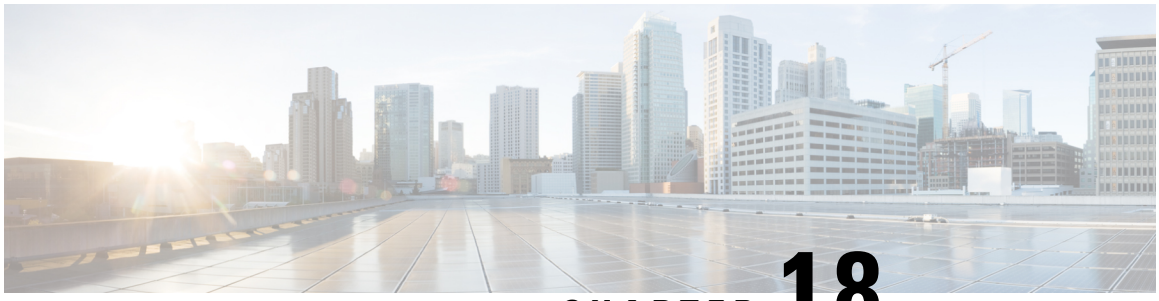
Step 2 Use the **upgrade hw-module subslot** <*slot*> / <*subslot*> **lte450 modem bootflash:** / <*filename*> command for executing firmware upgrade of modem.

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 18

Support for P-5GS6-RC-GL PIM

Table 16: Feature History Table

Feature name	Release information	Feature description
Support for P-5GS6-RC-GL PIM	Release 26.1.2	This feature enables support for the P-5GS6-RC-GL Pluggable Interface Module (PIM) on the Cisco Catalyst IR1101 router.

- Support for P-5GS6-RC-GL PIM, on page 195

Support for P-5GS6-RC-GL PIM

P-5GS6-RC-GL is a 3GPP Pluggable Interface Module (PIM) designed for reduced-throughput, moderate-bandwidth and cost-optimized 5G applications. It utilizes the Cisco EM8695 Reduced Capability (RedCap) cellular modem to provide 5G SA (Standalone) RedCap and 4G LTE connectivity.

The P-5GS6-RC-GL PIM provides an affordable, future-ready 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

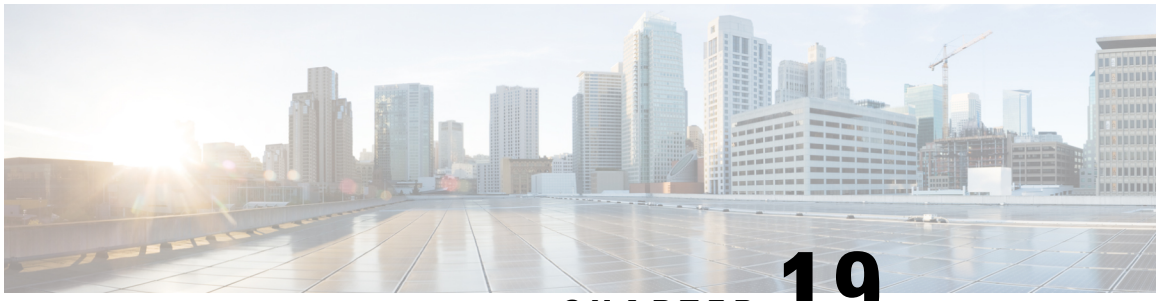
The benefits of using the P-5GS6-RC-GL PIM include:

- It delivers 4G LTE and 5G SA RedCap connectivity suited for entry level 5G deployments.
- It provides a reliable migration path for customers as cellular networks evolve.
- If you are an existing customer, it allows you to migrate to 5G SA RedCap technology.
- It addresses the need for affordable, long-term cellular connectivity.

Support for P-5GS6-RC-GL PIM

Note

- RedCap is a streamlined 5G NR device category designed for use cases that do not require the full performance of standard 5G. It offers reduced bandwidth, complexity, and power consumption, making it suitable for mid-tier Cisco IIoT applications.
- The P-5GS6-RC-GL PIM supports only 5G SA RedCap networks. It does not support 5G Non-Standalone (NSA) operation.
- The P-5GS6-RC-GL PIM supports additional LTE Band 106 along with other additional supported bands.
- For more information on the technical specifications of EM8695 modem, see 5G RedCap Module Available with IR1101 and 5G RedCap Module Available with IR1000



CHAPTER 19

Support for MACsec and IPsec Encryption

- Support for MACsec and IPsec Encryption, on page 197
- Software Supported MACsec, on page 197

Support for MACsec and IPsec Encryption

From Cisco IOS XE 17.14.1 on IR1101, you can configure MACsec on the LAN port and IPsec on the WAN port to secure sensitive traffic. This traffic moves from DA assets. The IR1101 router connects to the DA assets, allowing them to use MACsec encryption at Layer 2 and create IPsec tunnels through the LTE or WAN port.

Note Cisco supports only the **should secure** MACsec mode for IR1101, which allows unencrypted traffic even in a secured state.

Limitation:

The IR1101 does not support the **must secure** mode.

For more information about IPsec and MACsec configuration, see the following guides:

- Cisco Catalyst IR1101 Rugged Series Router Software Configuration Guide
- Security and VPN Configuration Guide, Cisco IOS XE 17.X

Software Supported MACsec

Overview

All existing Cisco IOS XE based router/switch use special transceiver to do MACsec encryption/decryption. This software MACsec uses CDAL infrastructure in QFP to do crypto operation. Comparing to the hardware choice, the way configuration/status/datapath is done is different thus creating some limitation on the functionality.

Release 17.10.1a only supports MACsec on L2 interfaces. The MACsec port must be put into access mode. As the encryption happens on the egress SVI interface, the vlan used for the port should be unique, meaning no other interface can use that vlan. This limitation is because the QFP does not have MAC table information.



Note Since MACsec is being done through software, performances are not line rate on L2 interfaces.



Note Cisco supports only the should secure MACsec mode for IR1101, which allows unencrypted traffic even in a secured state.

Limitation:

The IR1101 does not support the must secure mode.

For an egress packet, SVI only know the packet needs to go out on a vlan without info about any specific interface. It is up to the switch chip to decide which port to go. All the packets without MACsec tag can come in as usual. Outgoing L2 packet will also egress without encryption or modification.

Both the NE and NA license support GCM-AES-128. This feature is not available running the NPE image.

The MACsec protocol is defined in IEEE802.1AE.

Feature Limitations

- MACsec is not supported in controller mode in this release.
- There must be a unique vlan id for a MACsec interface.
- Only gcm-aes-128 is supported in this initial release.
- Both explicit and non-explicit SCI are supported on ingress side. The IR1101 sends out only explicit SCI packets as it is not an end system.
- The IR1101 does not support confidentiality offset.
- Integrity only is not supported in this first release.
- 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.
- IP Device Tracking (IPDT) is not supported in the host-to-switch MACsec configurations.

Related Documentation

Further information can be found at the following:

- MACsec and the MACsec Key Agreement (MKA) Protocol
- MACSEC and MKA Configuration Guide, Cisco IOS XE 17

Sample MKA Configuration

See the following example:

```
conf t
   aaa new-model
   mka policy p1
       key-server priority 1
       macsec-cipher-suite gcm-aes-128
       sak-rekey interval 3600
end
conf t
   key chain cak1 macsec
      key 414243
         cryptographic-algorithm aes-128-cmac
         key-string 0 12345678901234567890123456789012
         lifetime local 00:00:00 29 November 2021 infinite
end
conf t
   int fa 0/0/2
      switchport mode access
      switchport access vlan 77
      mtu 1532
      mka policy p1
      mka pre-shared-key key-chain cak1
      macsec network-link
      macsec replay-protection window-size 128
end
```

Show Commands

Show cpp_cp internal info:

```
show platform hardware cpp active feature soft-macsec server tx [dp] [item]
show platform hardware cpp active feature soft-macsec server rx [dp] [item]
show platform hardware cpp active feature soft-macsec server control [dp] [item]
```

Other show commands:

```
show macsec summary
show macsec status int fa 0/0/2
show macsec statistics int fa 0/0/2A
```

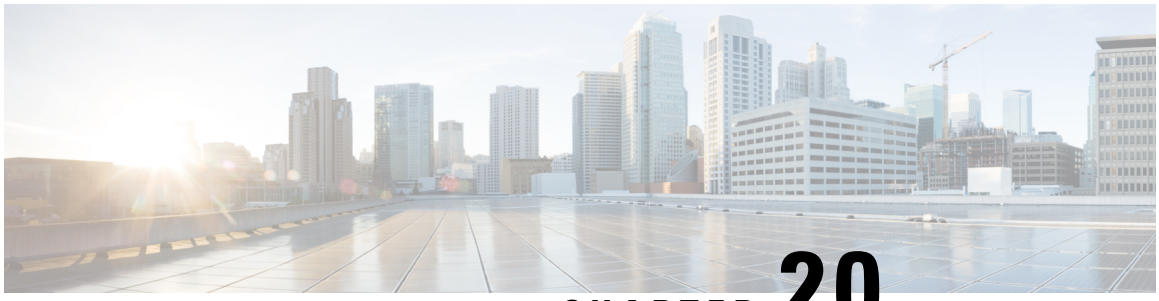
Clear Statistics

```
Clear macsec statis int fa 0/0/2
```

Test Command

Print 10 MKA packet for debug:

```
test platform software smacsec mka-ingress
```



CHAPTER 20

Support for the X25 over TCP

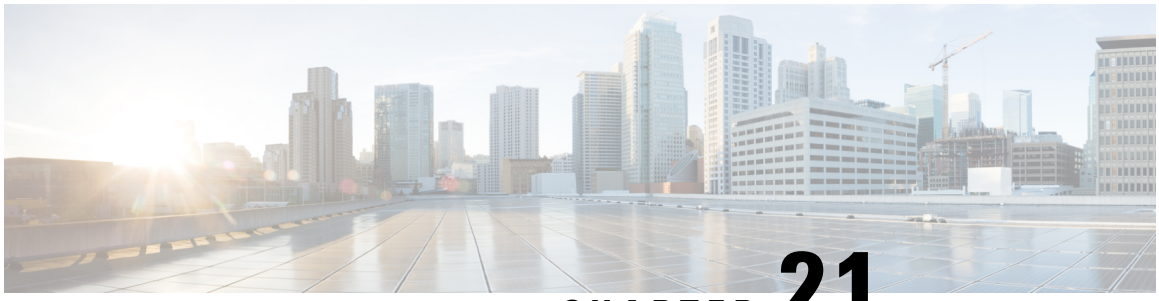
- Support for the X25 over TCP (XOT), on page 201

Support for the X25 over TCP (XOT)

X.25 is an ITU standard for packet switching Wide Area Network (WAN). This is used in the Telecommunication industry over serial interfaces that are replaced by IP Network. An X25 connection can be established by using a PAD connection similar to Telnet/SSH. The IR1101 router has only one asynchronous serial interface where features of X25 are not supported. However, we can communicate to the X25 edge devices using by using feature TCP over X25 (XOT). With XOT, we can directly establish a PAD connection to X25 edge devices. Also, we can assign default or customized profiles to the access-groups by changing various parameters of X25 packets.

For additional information about XOT for IOS-XE, see the following:

Wide-Area Networking Configuration Guide: X.25 and LAPB, Cisco IOS XE



CHAPTER 21

Dying Gasp Support on Ethernet or GigEthernet and SFP Port on IR1101

- Overview, on page 203
- Dying Gasp Support on IR1101 Routers, on page 203
- Dying Gasp Configuration Command, on page 203
- Configure Dying Gasp, on page 204
- Configuration Example for Dying Gasp, on page 204
- Verify Dying Gasp Packet Generation, on page 205

Overview

This chapter provides information about the Dying Gasp feature, which enables a router to send alerts to connected devices as it shuts down due to a power failure.

Dying Gasp Support on IR1101 Routers

From Cisco IOS XE Release 17.15.1, the Dying Gasp support is available for IR1101 router. Dying Gasp is a signal or an alert generated when the router is about to shut down due to a power failure. The IR1101 router sends dying gasp alerts to connected devices as packets through configured Fast Ethernet and Gigabit Ethernet ports. You can configure dying gasp packets for SNMP, Syslog, or Ethernet OAM protocols by specifying the primary and secondary destination protocols.

Note The IR1101 router creates dying gasp packets based on the selected destination protocol.

Dying Gasp Configuration Command

To enable dying gasp notification through Syslog, SNMP, or Ethernet OAM, use the dying gasp command:

Table 17: Dying Gasp Configuration Command

Command	Description
dying-gasp primary {syslog | snmp-trap | ethernet-oam} secondary {syslog | snmp-trap | ethernet-oam} dying-gasp	• primary—Dying Gasp primary notification • secondary—Dying Gasp secondary notification • ethernet-oam—Enable Ethernet OAM notification • snmp-trap—Enable trap notification sent to SNMP server • syslog—Enable system logger

Configure Dying Gasp

Procedure

Step 1 Enter the privileged EXEC mode.

```
router# enable
```

Step 2 Enter the configuration mode.

```
router# configure terminal
```

Step 3 Enable dying gasp notification through Syslog or SNMP or Ethernet OAM.

```
router(config)# dying-gasp primary {ethernet-oam|snmp-trap|syslog} secondary
{ethernet-oam|snmp-trap|syslog}
```

Example:

```
router(config)# dying-gasp primary ethernet-oam secondary snmp-trap
```

Step 4 Exit the global configuration mode and return to privileged EXEC mode.

```
router(config)# exit
```

Configuration Example for Dying Gasp

The example shows how to configure Ethernet OAM as primary destination and SNMP as secondary destination:

```
Router> enable
Router# configure terminal
Router# (config)#int gi0/0/0
Router(config-if)# ethernet-oam
Router(config-if)# exit
Router(config)# int fa0/0/2
Router(config-if)# ethernet-oam
Router(config-if)# exit
```

```
Router(config)# snmp-server host 192.168.10.20 ver 2c xyz
Router(config)# dying-gasp primary ethernet-oam secondary snmp-trap
```

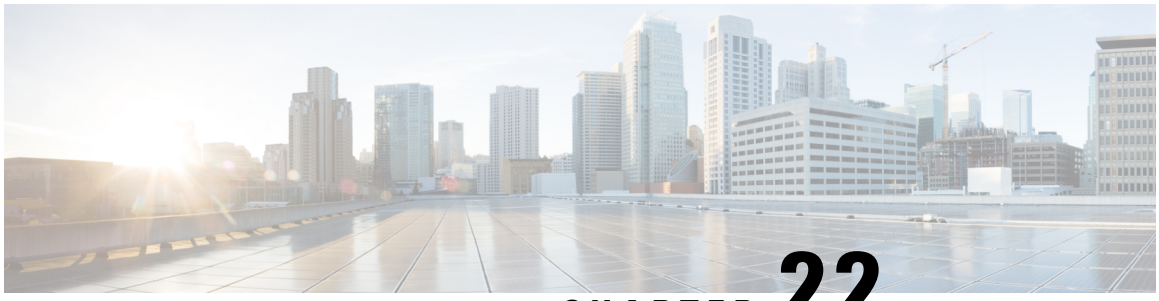
Verify Dying Gasp Packet Generation

Use the **show** command as given in the example to verify primary and secondary destination protocols.

```
Router#show dying-gasp status
Dying Gasp Configuration
Ethernet OAM Enabled
SNMP Trap Enabled
Syslog Disabled
```

Use the **show** command as given in the example to verify the dying gasp packets.

```
Router# show dying-gasp packets
SNMP Trap packet for server 192.168.10.20,
link type IP interface, via GigabitEthernet0/0/0,
local IP address 192.168.10.30 encap type is ARPA,
local hardware address 90eb.5027.7800 next hop IP address 192.168.10.20,
next hop hardware address 90eb.5027.de80
Ethernet OAM packet for interface GigabitEthernet0/0/0,
link type LINK_ETHER_OAM encap type is ARPA,
local hardware address 90eb.5027.7800 remote hardware address 0180.c200.0002 Ethernet OAM
packet for interface FastEthernet0/0/2,
link type LINK_ETHER_OAM encap type is ARPA,
local hardware address 90eb.5027.7802 remote hardware address 0180.c200.0002
Ethernet OAM packet for interface FastEthernet0/0/2,
link type LINK_ETHER_OAM encap type is ARPA,
local hardware address 90eb.5027.7802 remote hardware address 0180.c200.0002
```



CHAPTER 22

2nd L3 WAN Port for IR1101

- 2nd L3 WAN Port for IR1101, on page 207

2nd L3 WAN Port for IR1101

From Cisco IOS XE Release 17.15.1, the IR1101 routers support the Layer 2 G0/0/5 interface as a WAN port, in addition to the existing Layer 3 G0/0/0 interface. This enhancement allows the default Layer 2 interface G0/0/5 to be converted into a Layer 3 WAN port.

Benefits of an Additional WAN Port

- Supports multi-backhaul use cases such as Cisco Catalyst SD-WAN High Availability (HA) capability without cellular connectivity.
- Enables Layer 3 back-to-back ring connections to maintain network resilience and optimize traffic flow.
- Provides dual functionality to increase flexibility in network design, accommodating scenarios that require additional routed interfaces without the need for extra hardware.

Supported Expansion Modules

- IRM-1100-SPMI
- IRM-1100-SP

Configure the Layer 3 WAN Port

Use the **no switchport** command to convert the interface G0/0/5 to Layer 3 interface, as shown in the procedure below:

```
Router#configure terminal
Enter configuration commands, one per line. End with CNTL/Z.
Router(config)#interface g0/0/5
Router(config-if)#no switchport
Router(config-if)#end
```

Verify the Layer 3 WAN Port

Use the **show run interface** command to verify that Layer 3 interface is ready to be configured.

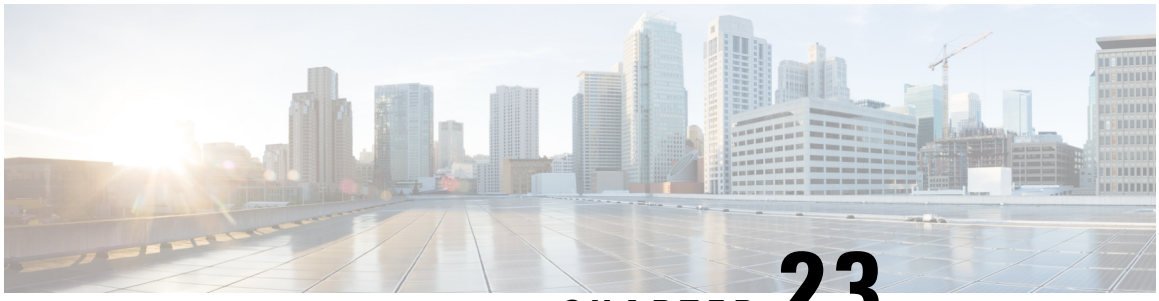
```
Router#show run interface g0/0/5
Building configuration...
Current configuration : 68 bytes
!
interface GigabitEthernet0/0/5
no switchport
no ip address
end
```

You can now assign an IP address to the interface G0/0/5.

Configure the Layer 2 Interface

Use the **switchport** command to switch interface G0/0/5 from Layer 3 to Layer 2 mode, as shown in the procedure below:

```
Router#configure terminal
Enter configuration commands, one per line. End with CNTL/Z.
Router(config)#interface g0/0/5
Router(config-if)#switchport
Router(config-if)#end
```



CHAPTER 23

Managing USB Access Control

•

• Option to Enable or Disable USB Access, on page 209

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. However, the USB port could be considered a potential security risk. Functionality was added to enable or disable the USB access.



CHAPTER 24

Cisco Network Plug and Play Agent

This chapter contains the following sections:

- Prerequisites for Cisco Network Plug and Play Agent, on page 211
- Restrictions for Cisco Network Plug and Play Agent, on page 212
- Information About Cisco Network Plug and Play Agent, on page 212
- Security Methods for the PnP Discovery Process, on page 227
- Security Methods for Post-PnP Discovery Process, on page 233
- How to Configure Cisco Network Plug and Play Agent, on page 236
- Troubleshooting and Debugging, on page 246
- Glossary, on page 247
- Additional References for Open Plug-n-Play Agent, on page 248

Prerequisites for Cisco Network Plug and Play Agent

- Cisco Network Plug and Play (PnP) deployment method depends on the type of discovery process as required by the customer.
- Deploy the discovery mechanism, either a DHCP server discovery process or a Domain Name Server (DNS) discovery process, before launching the PnP.
- Configure the DHCP server or the DNS server before deploying the PnP.
- Ensure that the PnP server talks to the PnP agent.
- Ensue that the Cisco Network PnP Agent has connectivity with the PnP Server. The Cisco Network PnP Agent should be able to PING server.
- The PnP agent enforces the PnP server to send user credentials for every request. Cisco recommends the usage of HTTP secure (HTTPS) protocol.

Note

- The terms Cisco Network Plug and Play, PnP are interchangeably used in this guide and all mean the same.
- The terms PnP agent, agent, and deployment agent are interchangeably used in this guide and all mean the same.
- The terms PnP server, server, and deployment server are interchangeably used in this guide and all mean the same.

Restrictions for Cisco Network Plug and Play Agent

- Cisco Network Plug and Play (PnP) agent facilitates HTTP and HTTP secure (HTTPS) transport based communication with the server.
- HTTPS cannot be used on platforms where crypto-enabled images are not supported (also, do not use Secure Sockets Layer [SSL] or Transport Layer Security [TLS] protocols if crypto-enabled images are used).
- Non-VLAN 1 configuration-Cisco Network Plug and Play supports devices using VLAN 1 by default. To use a VLAN other than 1, adjacent upstream devices must use supported releases and configure the following global CLI command on the upstream device to push this CLI to the upcoming Plug and Play device: **pnp startup-vlan x**. When you execute this command on an adjacent upstream device, the VLAN membership change does not happen on that device. However, all the active interfaces on the upcoming Plug and Play device are changed to the specified VLAN. This guideline applies to both routers and switches.

Note When performing a firmware upgrade during the PNP process, it is best to remove any old images on the router to prevent an incorrect image from loading.

Further details are available by viewing CSCwd68868.

Information About Cisco Network Plug and Play Agent

Cisco Network Plug and Play Deployment Solution

The Cisco Network PnP Agent is a part of Cisco Network Plug and Play solution. The Cisco initiated Network Plug and Play (PnP) deployment solution supports the concept of redirection and includes a PnP agent, a PnP server, and other components. Simplified deployment process of any Cisco device automates the following deployment related operational tasks:

- Establishing initial network connectivity for the device
- Delivering device configuration

- Delivering software and firmware images
- Delivering licenses
- Delivering deployment script files
- Provisioning local credentials
- Notifying other management systems about deployment related events

Simplified deployment reduces the cost and complexity and increases the speed and security of deployments.

Cisco Network Plug and Play (PnP) agent is a software application that is running on a Cisco IOS or IOS-XE device. The PnP agent together with the PnP deployment server provides effortless deployment services. When a device is powered on for the first time, the PnP agent process wakes up in the absence of the startup config, user input on the device's console, and attempts to discover the address of the PnP server. The PnP agent uses methods like DHCP, Domain Name System (DNS), and others to acquire the desired IP address of the PnP server. When the PnP agent successfully acquires the IP address, it initiates a long lived, bidirectional layer 3 connection with the server and waits for a message from the server. The PnP server application sends messages to the agent requesting for information and services to be performed on the device.

The PnP agent converges existing solutions into a unified agent and adds functionality to enhance the current solutions. The main objectives of PnP agent are:

- Provide consistent day 1 deployment solution for all the deployment scenarios.
- Add new features to improve existing solutions.
- Provide day 2 management framework mainly in the context of configuration and image upgrades.

Cisco Network Plug and Play Features

Some of the features that the Cisco Network Plug and Play agent provides:

- Day 0 boot strapping—Configuration, image, licenses, and other files
- Day 2 management—Configuration and image upgrades and on-going monitoring of Simple Network Management Protocol (SNMP) and syslog messages.
- Open communication protocol—Enables customers and partners to write applications
- XML based payload over HTTP between the server and the agent.
- Security—Authentication and encrypted communication channel between the management app and the agent
- Deployment and management of devices behind firewall and Network Address Translation (NAT).
- Support for one-to-one and one-to-many communication
- Support for policy based deployment (product ID or location of the device)
- Deployment based on unique ID (Unique Device Identifier [UDI] or MAC)
- Unified solution across Cisco platforms (including IOS classic)
- Support for various deployment scenarios and use cases
- Zero-touch when possible, low-touch when needed

Cisco Network Plug and Play Agent Services and Capabilities

The services and capabilities of the Cisco Network Plug and Play agent are as follows:

1. Backoff
2. CLI execution
3. Configuration upgrade
4. Device information
5. File transfer
6. Image install
7. License install
8. PnP tagging
9. Script execution
10. Topology information



Note The PnP server provides an optional checksum tag to be used in the image installation and config upgrade service requests by the PnP agent. When the checksum is provided in the request, the image install process compares the checksum against the current running image checksum.

If the checksums are same, the image being installed or upgraded is the same as the current image running on the device. The image install process will not perform any other operation in this scenario.

If the checksums are not same, then the new image will be copied to the local file system, and checksum will again be calculated and compared against the checksum provided in the request. If same, the process will continue to install the new image or upgrade the device to new image. If now, the checksums are not same, the process will exit with error.

Backoff

A Cisco IOS device that supports PnP protocol (that uses HTTP transport), requires the PnP agent to send the work request to the PnP server continuously. In case the PnP server does not have any scheduled or outstanding PnP service for the PnP agent to execute, the continuous no operation work requests exhausts both network bandwidth and device resource. This PnP backoff service allows the PnP server to inform the PnP agent to rest for the specified time and call back later.

CLI Execution

Cisco IOS supports two modes of command execution—EXEC mode and global configuration mode. Most of the EXEC commands are one-time commands, such as **show** commands, which show the current configuration status, and **clear** commands, which clear counters or interfaces. The EXEC commands are not saved when a device reboots. Configuration modes allow user to make changes to the running configuration. If you save the configuration, these commands are saved when a device reboots.

Note For **show** command request and response details and for all PnP configuration commands, see *Cisco Network Plug and Play Agent Command Reference*.

Configuration Upgrade

There are two types of configuration upgrades that can happen in a Cisco device—copying a new configuration files to startup configuration and copying new configuration files to running configuration.

Copying a new configuration files to startup configuration— The new configuration file is copied from the file server to the device through **copy** command and file check is performed to check the validity of the file. If the file is valid, then the file is copied to startup configuration. Backing up the previous configuration file will be done if there is enough disk space available. The new configuration is seen when the device reloads again.

Copying new configuration files to running configuration— The new configuration file is copied from the file server to the device through **copy** command or **configure replace** command. Configuration file replace and rollback may leave the system in an unstable state if rollback is performed efficiently. So configuration upgrade by copying the files is preferred.

Device Information

The PnP agent provides capability to extract device inventory and other important information to the PnP server on request. The following five types of device-profile requests are supported:

1. all—returns complete inventory information, which includes unique device identifier (UDI), image, hardware and file system inventory data.
2. filesystem— returns file system inventory information, which includes file system name and type, local size in bytes, free size in bytes, read flag, and write flag.
3. hardware— returns hardware inventory information, which includes hostname, vendor string, platform name, processor type, hardware revision, main memory size, I/O memory size, board ID, board rework ID, processor revision, midplane revision, and location.
4. image—returns image inventory information, which includes version string, image name, boot variable, return to rommon reason, bootloader variable, configuration register, configuration register on next boot, and configuration variable.
5. UDI— returns device UDI.

File Transfer

The PnP file server hosts files that can be copied over by the deploying devices in the network. The file server can be a dedicated server hosting files or a part of the device hosting the PnP server. The PnP agent uses standard file transfer protocols to copy files from the remote file server to the device. If the device is running a crypto image then secured file transfer protocols such as SFTP, SCP, HTTPS are supported. For devices running non-crypto images, the PnP agent supports unsecured copy protocols such as FTP, TFTP, HTTP.

Image Install

Image installation service enables a PnP-enabled device to perform image upgrade on receiving a request from the PnP server.

Standalone Devices

When the PnP agent on a standalone device receives a request from the PnP server, the agent parses the XML payload and identifies the request as an Image Upgrade request. The agent then creates an ImageInstall process, which identifies the request as a standalone image install request. The PnP agent populates the data structure defined by the ImageInstall service and passes it to the ImageInstall service.

The Image Install service then performs the following operations to successfully load the device with the new image:

1. Copies the image from the file server to a local disk (the file server information is provided by the PnP server in the request).
2. Configures the device to load the new image on next reload by executing the **boot system** command.
3. Reloads the device and sends a message to the PnP server.

PnP Tagging

Cisco IOS provides capability to assign tags to the devices for better grouping and tracking of all Cisco devices. The PnP agent provides XML service for configuring the tag information on the device and for propagating the tag information within the network using Cisco Discovery Protocol (CDP). The purpose of this service is for the PnP agents to get to know their tag information and to pass on this information to the PnP server upon request.

Topology Information

By default, every Cisco device on the network runs Cisco Discovery Protocol (CDP). Through CDP, devices in the network discover their immediate neighbors and populate their databases with the attributes learnt or derived through the protocol. This neighbor information is stored in the database and is available on demand by the device to the PNP server. Typical neighbor information comprises neighboring device ID, software version, hardware platform, interface ip, and the port on which CDP messages are sent or received.

Software Maintenance Upgrade

The software maintenance upgrade (SMU) is a package that contains fixes for a specific defect or security resolution to a released image. SMUs are created to respond to immediate issues and do not include new features. SMUs do not have a large impact on router operations. SMU versions are synchronized to the package major, minor, and maintenance versions they upgrade.

To install and activate a software maintenance upgrade package:

Procedure

Step 1 Use the **install add** <*filename*> command to unpack the package software file and copy it to the boot device (usually disk0). If the file is on a remote source, use the tftp/ftp option to copy the file to the device.

After the file is copied to the device, information within the package is used to verify compatibility with the target cards and with the other active software. Actual activation is performed only after the package compatibility and application program interface (API) compatibility checks are passed.

Step 2 To activate a package, use the **install activate** <*filename*> command. The activate operation will run the compatibility checks and install the software maintenance upgrade package. If it is a reload software maintenance upgrade, it will automatically initiate a reload.

Step 3 Use the **install commit** command to commit the changes

Step 4 To deactivate the package, use the **install deactivate** <*filename*> command.

Step 5 If you find that you prefer a previous package set over the currently active package set, you can use the **install rollback to committed** command to make a previously active package set active again

Step 6 To remove the installed version, use the **install remove** <*filename*> command.

This example shows how to install and remove the software maintenance upgrade package on a device.

```
install add <filename>
install activate <filename>
install commit
install rollback to committed
install remove <filename>
```

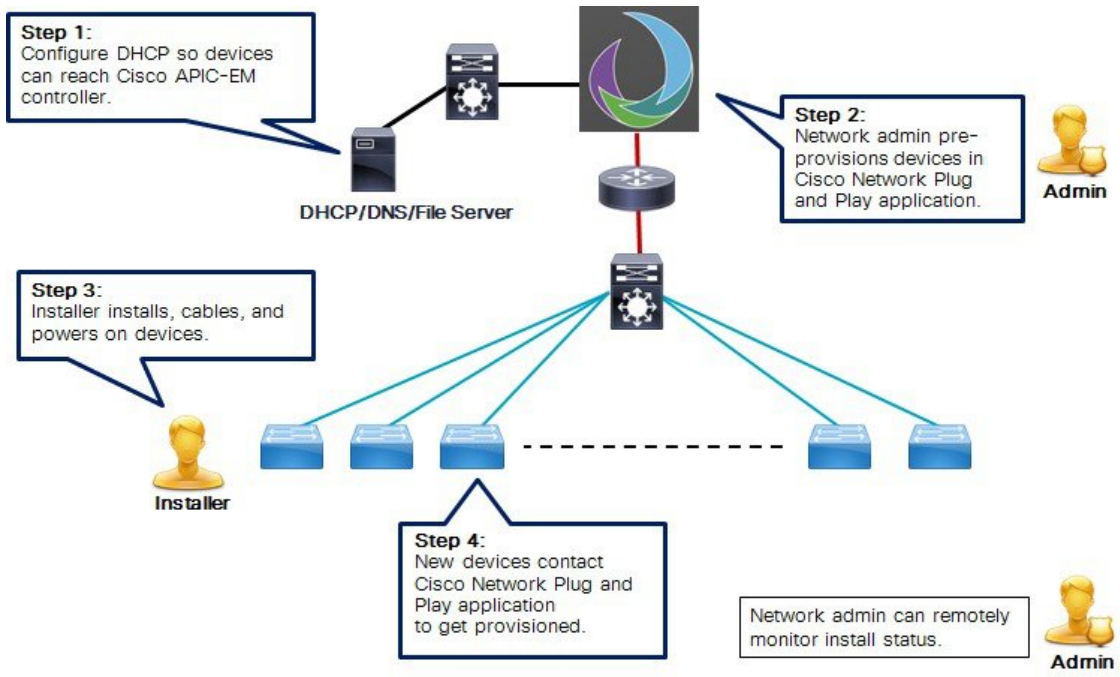
Cisco Network Plug and Play Agent

The Cisco Network Plug and Play agent is an embedded software component that is present in all Cisco network devices that support simplified deployment architecture. The PnP agent understands and interacts only with a PnP server. The PnP agent first tries to discover a PnP server, with which it can communicate. Once a server is found and connection established, the agent performs deployment related activities like configuration, image, license, and file updates by communicating with the server. It also notifies the server of all interesting deployment related events like out-of-band configuration changes and a new device connection on an interface.

Cisco Network Plug and Play Server

The Cisco Network Plug and Play server is a central server that encodes the logic of managing and distributing deployment information (images, configurations, files, and licenses) for the devices being deployed. The server communicates with the agent on the device that supports the simplified deployment process using a specific deployment protocol.

Figure 29: Simplified Deployment Server



The PnP server also communicates with proxy servers like deployment applications on smart phones and PCs, or other PnP agents acting as Neighbor Assisted Provisioning Protocol (NAPP) servers, and other types of proxy deployment servers like VPN gateways.

The PnP server can redirect the agent to another deployment server. A common example of redirection is a PnP server redirecting a device to communicate with it directly after sending the bootstrap configuration through a NAPP server. A PnP server can be hosted by an enterprise. This solution allows for a cloud based deployment service provided by Cisco. In this case, a device discovers and communicates with Cisco's cloud based deployment service for initial deployment. After that, it can be redirected to the customer's deployment server.

In addition to communicating with the devices, the server interfaces with a variety of external systems like Authentication, Authorizing, and Accounting (AAA) systems, provisioning systems, and other management applications.

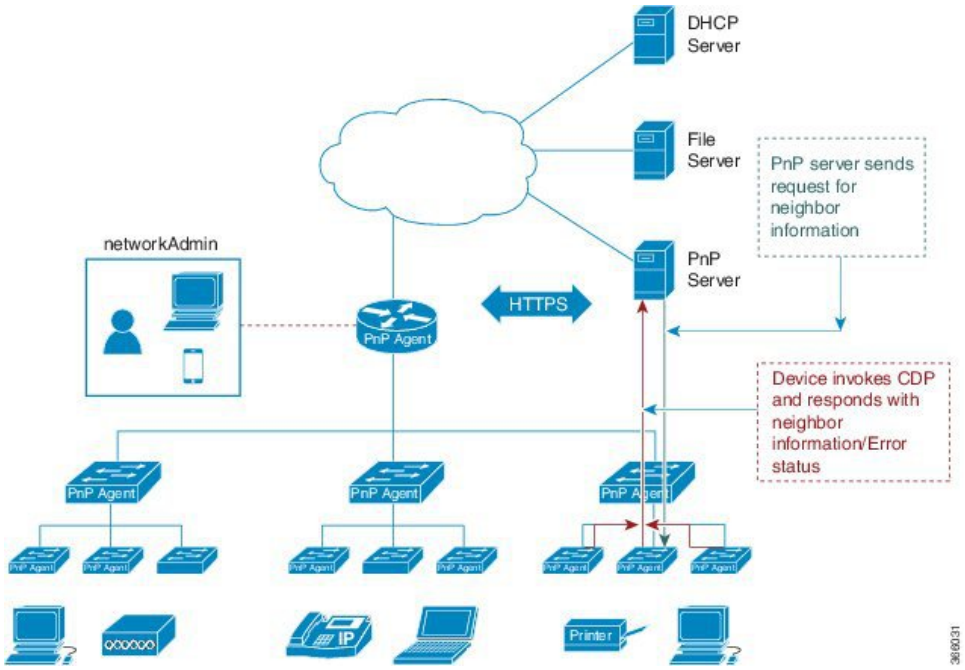
Cisco Network Plug and Play Agent Deployment

The following steps indicate the Cisco Network Plug and Play agent deployment on Cisco devices:

1. The Cisco device, having PnP agent contacts the PnP server requesting for a task, that is, the PnP Agent sends its unique device identifier (UDI) along with a request for work.
2. The PnP server if it has any task for the device, sends a work request. For example, image install, config upgrade, and so on.
3. When the PnP agent receives the work request, executes the task and sends back a reply to the PnP server about the task status, whether it is a success or error, and the corresponding information requested.

Cisco Network Plug and Play Agent Network Topology

Figure 30: Network Topology of Cisco Network Plug and Play Agent Deployment



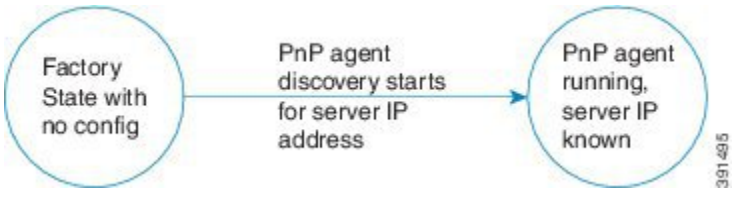
Cisco Network Plug and Play Agent Initialization

The Cisco Network Plug and Play agent software is currently available on all Cisco IOS XE platforms, and is enabled by default. The PnP agent can be initiated on a device by the following ways:

Absence of Startup Configuration

New Cisco devices are shipped to customers with no startup configuration file in the NVRAM of the devices. When a new device is connected to a network and powered on, the absence of a startup configuration and the user input file on the device, it will automatically trigger the Cisco Network Plug and Play agent to discover the PnP server IP address.

Figure 31: State Diagram for PnP Trigger with no Startup Configuration



CLI Configuration for Open Plug-n-Play Agent

Network administrators may use CLI configuration mode to initiate the Open Plug-n-Play (PnP) agent process at any time. By configuring a PnP profile through CLI, a network administrator can start and stop the PnP agent on a device. When the PnP profile is configured with CLI, the device starts the PnP agent process which, in turn, initiates a connection with the PnP server using the IP address in the PnP profile.

Figure 32: State Diagram for PnP Trigger with CLI Configured PnP Profile



Cisco Network Plug and Play Agent Deployment Solutions

This section discusses the functionality of the Cisco Network Plug and Play agent, exposed to the PnP server, for device deployment and management. The PnP agent deployment solution comprises the discovery process initiated by the agent, communication between the device, agent, and the server, and the PnP agent services. The PnP solution is described in detail in the following sections:

Cisco Network Plug and Play Agent Discovery Process

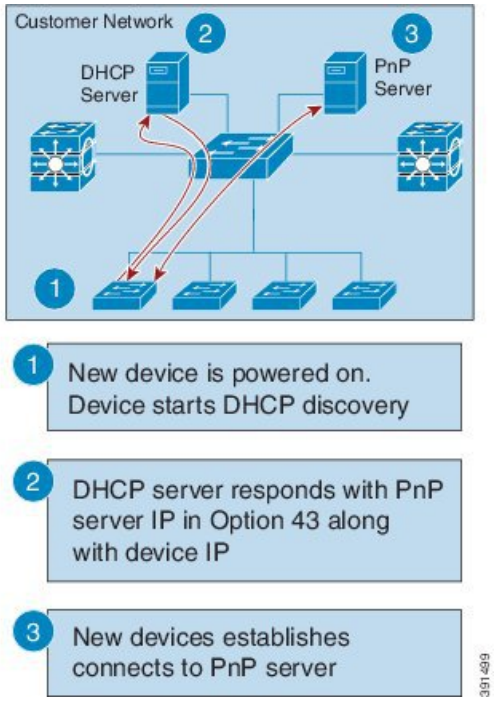
When the device boots up, the absence of any startup config on the NVRAM triggers the PnP discovery agent to acquire the IP address of the PnP server. In order to acquire the IP address of the PnP server, the PnP agent goes through one of the following discovery mechanisms:

1. PnP discovery through DHCP server
2. PnP discovery through DHCP snooping
3. PnP discovery through DNS lookup
4. PnP proxy for layer 2 and layer 3 devices
5. PnP deployment application

Cisco Network Plug and Play Discovery through DHCP Server

Device with no startup configuration in the NVRAM triggers the Cisco Network Plug and Play agent to initiate a DHCP discovery process which acquires the IPv4 configuration from the DHCP server required for the device. The DHCP server can be configured to insert additional information using vendor specific option 43 upon receiving option 60 from the device with the string ‘cisco pnp’, to pass on the IPv4 address or hostname of the PnP server to the requesting device. When the DHCP response is received by the device, the PnP agent extracts the option 43 from the response to get the IP address or the hostname of the PnP server. PnP agent then uses this IPv4 address or hostname to communicate with the PnP server.

Figure 33: DHCP Discovery Process for PnP server



Assumptions:

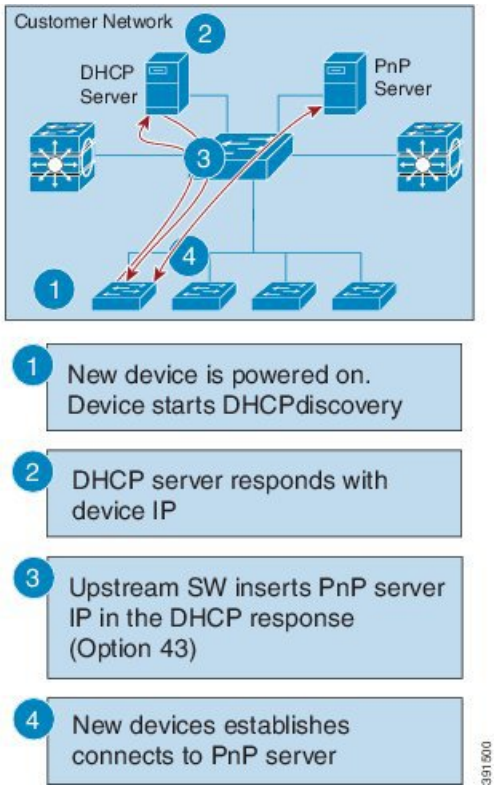
- New devices can reach DHCP server
- Customer is willing to configure DHCP server for network devices

Plug-n-Play Discovery through DHCP Snooping

If a third party DHCP server cannot be configured to insert any vendor specific options, an existing Cisco Open Plug-n-Play (PnP) enabled device can be configured to snoop in to the DHCP response and insert PnP specific option 43 with the PnP server IP address.

Before inserting the option 43, the snooping agent verifies if the DHCP message is from a Cisco device in the network. The remaining DHCP discovery process is same as described in the previous section.

Figure 34: DHCP Snooping by PnP Server



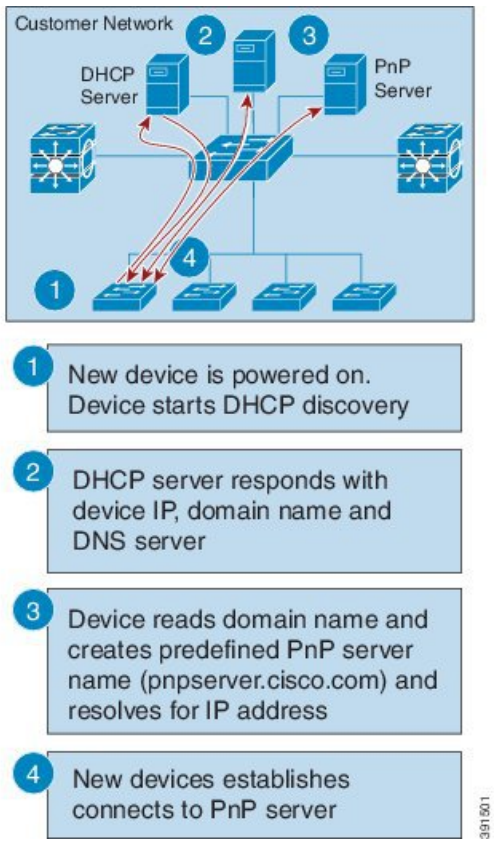
Assumptions:

- New devices can reach DHCP server
- New devices can reach DNS server
- Customer is not willing to configure DHCP server for network devices
- Upstream switch (SW) is configured to snoop DHCP and insert PnP server IP

Cisco Network Plug and Play Discovery through DNS Lookup

When the DHCP discovery fails to get the IP address of the Cisco Network Plug and Play server, the agent falls back on Domain Name System (DNS) lookup method. PnP agent then uses a preset deployment server name. The agent obtains the domain name of the customer network from the DHCP response and constructs the fully qualified domain name (FQDN). The following FQDN is constructed by the PnP agent using a preset deployment server name and the domain name information for the DHCP response, *deployment.customer.com*. The agent then looks up the local name server and tries to resolve the IP address for the above FQDN.

Figure 35: DNS Lookup for deployment.customer.com



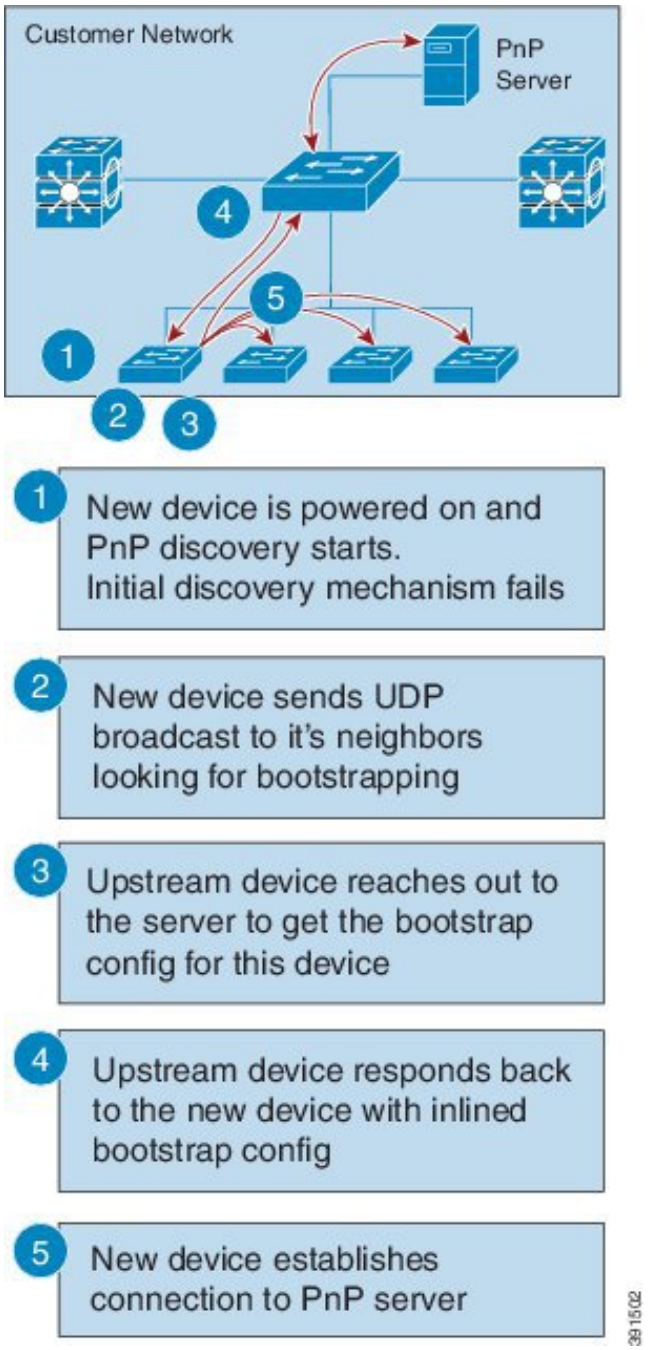
Assumptions:

- New devices can reach DHCP server
- Customer deployed PnP server in the network with the name “pnpserver”

Cisco Network Plug and Play Proxy Server for Layer 3 and Layer 2 Devices

This device listens to a specific port for any incoming PnP messages. The Cisco device which is trying to come up as a PnP device sends a UDP broadcast message to its network every 30 min for ten times. Hence, if the device does not receive a response, the broadcasts stop after 300 min.

Figure 36: DNS Lookup for Layer 3 and Layer 2 Devices



When the device hosting the proxy server process receives the incoming broadcasts, it verifies the version field in the request and forwards the request to the PnP server if version validation is successful. The proxy server process also caches the unique device identifier (UDI) of the requesting client coming in via incoming datagram before forwarding the request to PnP server.

Upon receiving the configlet datagram from PnP server, the proxy server validates UDI in the incoming datagram with the entries in the UDI cache. If validation is successful, proxy server process broadcasts the datagram to a specific port number reserved for the proxy client processes to receive datagrams.

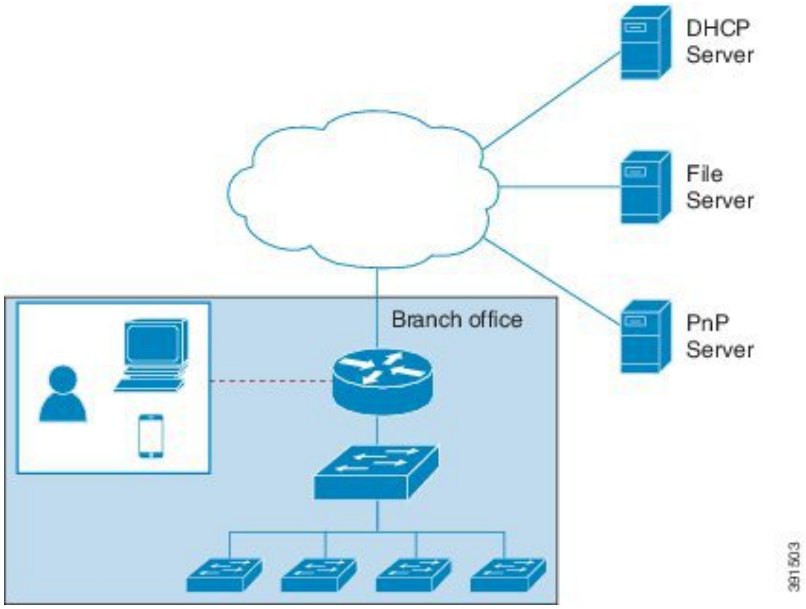
Upon receiving the datagrams, devices running proxy client processes, parse the incoming datagram for the target UDI. If the target UDI in the datagram matches the UDI of the device, proxy client process proceeds with framing, error control and configuring the configlet.

If the target UDI in the datagram fails to match UDI of the device, the packet is dropped.

Plug-n-Play Agent Deployment Application

A Cisco device can alternatively be manually configured by the network administrator using a deployment application running on their PC or on a smart phone. The PC or the smart phone can be connected to the device using an USB or an Ethernet cable.

Figure 37: Manually Configured PnP Agent



Plug-n-Play Agent Deployment Protocol

Deployment can be run over different transports. These transports include Ethernet and IP with Transport Layer Security (TLS). Layer 2 transport is typically used between a deployment agent and a proxy deployment server like a deployment application or as a deployment agent acting as a proxy. Transport between an agent and a server is over an IP connection with TLS for security. Transport between a proxy deployment server and a deployment server is also over IP with TLS.

Plug-n-Play Agent Application Protocol

The Cisco Open Plug-n-Play (PnP) agent application protocol is an XML-based protocol that defines a mechanism that allows network devices to be monitored and controlled by a remote application. The PnP agent is a software module running on a Cisco device. The PnP server is an application running as the network manager that remotely manages the network devices. The main features of the PnP protocol are as follows:

1. Supports HTTP protocols

2. Supports Transport Level Security (TLS) based encryption for HTTP
3. Uses HTTP secure (HTTPS) certificate for TLS handshake

Plug-n-Play Transport over Ethernet

Cisco Open Plug-n-Play (PnP) agent uses the Ethernet based transport in the following two scenarios:

- **Deployment agent communicating with a deployment application on a PC**: In this case, the PC is connected to the device being deployed using an Ethernet cable. The deployment application advertises itself as a deployment server supporting Ethernet transport.
- **Deployment agent communicating with an already deployed device acting as a proxy deployment server**: In this case, the new device being deployed has an Ethernet connection to an already deployed device. The deployment agent on the deployed device responds to the discovery requests and acts as a proxy deployment server for the new device.

Once discovery is complete, the deployment agent starts an unsecured XML stream with the deployment server over Ethernet. This protocol reserves an Ethertype (0xXX TBD) for this purpose. The deployment agent and the server then negotiate to use Extensible Authentication Protocol–Transport Layer Security (EAP-TLS) to protect the communication and complete the EAP-TLS session establishment. The deployment server then authenticates the device with the HTTP secure (HTTPS) certificate or some other supported mechanism.

Plug-n-Play Transport over IP

In Cisco Network Plug-n-Play (PnP) agent, the deployment agent opens TCP connection to the deployment server and starts an XML stream of messages. The server can request the use of Transport Layer Security (TLS) at this time. The agent closes the existing XML stream, initiates a TLS connection to the server, and then restarts the XML stream. The server can request agent authentication over the TLS connection.

Plug-n-Play Agent Security

Security to all Cisco Open Plug-n-Play (PnP) devices is provided at both transport level as well as the application level. The following sections describe the security mechanisms in detail:

Plug-n-Play Transport Layer 3 Security

For non-crypto or non-crypto-enabled images, TLS security choice is not possible. One alternative minimum security is to have the PnP agent initiate the connection to the specified trusted PnP server on port 5222.

Authentication and Authorization between Plug-n-Play Agent and Server

Once the Cisco Open Plug-n-Play (PnP) deployment agent discovers the PnP server, the agent engages the server in a Transport Layer Security (TLS) handshake. In order to authenticate itself to the server, the agent presents its HTTP secure (HTTPS) certificate. The administrator for the PnP server sets device authentication mechanisms which are acceptable for a particular deployment.

The deployment server presents its certificate to the deployment agent so that the agent can authenticate the server. Irrespective of whether the agent is able to verify the server certificate, the agent engages the deployment server in a post-TLS authorization exchange. In this exchange the agent requests the server to present its server authorization token. In response to this request the server presents the authorization token it had obtained from Cisco. The agent verifies the signature on the authorization token. If the authorization token is specific to a Unique Device Identifier (UDI), the agent also ensures its UDI is listed in the authorization-token. At the

end of this step, a secure communication channel is established between the deployment agent and the server. This secure communication channel is leveraged by the server to send deployment information to the agent.

Security Methods for the PnP Discovery Process

This section describes the methods that are used to secure the PnP agent-server communication in various scenarios. The security options are used by the PnP agent during the zero-touch PnP server discovery.

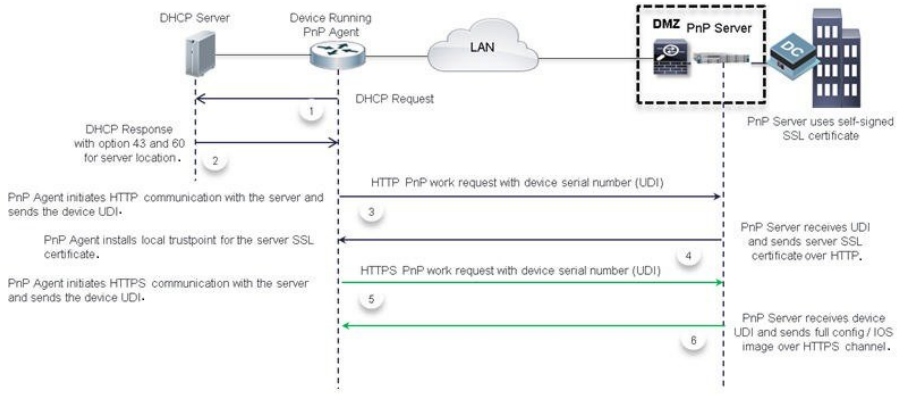
Self-Signed Certificate Based Authentication

The PnP server has an option to use a self-signed SSL certificate for server side authentication. When the PnP server uses a self-signed certificate, the PnP discovery cannot be used for automatically initiating secured communication from the agent to the server. The device goes through usual PnP discovery mechanisms and when it finds the server, the agent sends a work-request over HTTP. The server should use the PnP certificate-install service to instruct the agent to install the server self-signed certificate, and then automatically reconnect back to the server over HTTPs.

To keep the solution secured, it is recommended that you use the unsecured port 80 of server to deliver the one-time certificate installation to the devices. All other services should be sent over the secured port.

The following figure shows the end-to-end secured PnP workflow using a self-signed server SSL certificate.

Figure 38: PnP Deployment with Self-Signed Certificate

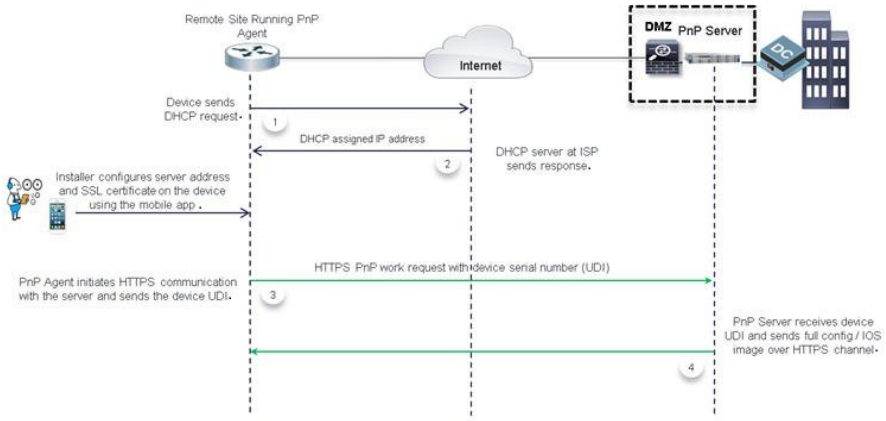


Mobile Device Based Secured Installation

As part of this solution, an application for mobile devices is available to configure the bootstrap on the devices. The mobile application can be used to install the server certificate directly on each device along with other bootstrap configuration and then allow the PnP agent initiate secured communication with the server. In this method, the server does not open up any unsecured port for certificate-install.

The following figure shows the end-to-end secured PnP workflow using the application on the mobile devices.

Figure 39: Secured PnP Deployment with the Mobile Application



CA-Signed Certificate based Authentication

Cisco distributes certificates signed by a signing authorities in a tar file format and signs the bundle with Cisco Certificate Authority (CA) signature. This certificate bundle is provided by Cisco infoSec for public downloads on cisco.com.

The certificates from this bundle can be installed on the Cisco IOS device for server side validation during SSL handshake. It is assumed that the server uses a certificate, which is signed by one of the CA that is available in the bundle.

The PnP agent uses the built-in PKI capability to validate the certificate bundle. As the bundle is signed by Cisco CA, the agent is capable of identifying the bundle that is tampered before installing the certificates on the device. After the integrity of the bundle is ensured by the agent, the agent installs the certificates on the device. After the certificates are installed on the device, the PnP agent initiates an HTTPs connection to the server without any additional steps from the server. The following mechanisms helps the PnP agent to initiate a zero-touch secured communication.

DHCP Option based Discovery over an IPv4 Network

The DHCP option 43 and option 60 is a vendor specific identifier which is used by the PnP agent to locate and connect to the PnP server. To support multiple vendors, the PnP agent in Cisco device sends out a case-sensitive “ciscopnp” as the option 60 string during the DHCP discovery. The DHCP server can be configured with multiple classes matching with a different option 60 string that comes from each network device. After the option 60 string matches, the DHCP server sends out the corresponding option 43 string back to the device. The following is the format for defining the option 43 for PnP deployments:

*option 43 ascii "5A;K5;B2;I10.30.30.10;J443;**Ttftp://10.30.30.10/ ios.p7b**;Z10.30.30.1*

The field ‘T’ in the PnP string provides an option for the network administrator to specify the location of the certificate bundle, which can be hosted on a local or remote file server.

If the certificate bundle is available at the specified location, then the agent:

1. Downloads the bundle from the file server to the device.
2. Checks the signature of the downloaded bundle to ensure it has a genuine Cisco signature.
3. Installs the certificates on the device.

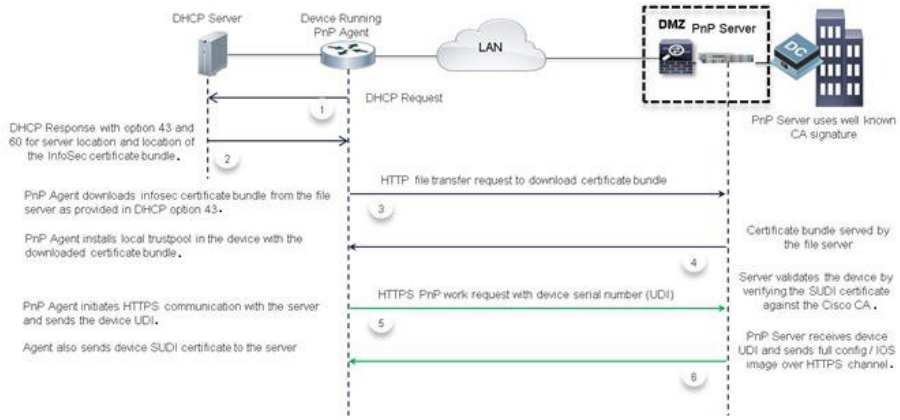
If the ‘T’ option is not specified and the transport mechanism is specified in the option 43 string as HTTPs, the PnP agent looks for the Cisco signed certificate bundle in the default folder of the same server *http://10.30.30.10:443/certificates/default/cert.p7b* .

If the certificates are available at the default location then the agent performs the steps mentioned above to install the certificates.

After the certificates are installed and the server discovery is complete, the agent initiates the HTTPs connection with the server without any additional configuration. During the HTTPs handshake, the device uses the certificates installed from the bundle to validate the server certificate.

The following figure shows the end-to-end secured PnP workflow using the CA bundle-based certificate.

Figure 40: Secured PnP Deployment with Trustpool



This flow works only if the server is using a certificate signed by one of the known signing authorities that is available in the bundle. If the server uses a certificate that is not a part of the bundle then the HTTPs handshake will fail. When you specify the option 43 string with HTTPs as a transport option and if the bundle download fails, the agent will not fall back to any of the unsecured communication protocol even if the server is reachable. If the transport option is specified as HTTP with a parameter 'T' pointing to a valid certificate bundle location, the agent overrides the transport option HTTP and changes it to HTTPs for secured communication. Generally, the agent will choose the most secured communication from the available options.

The path specified in the DHCP option 43 to locate the certificate bundle file can be an absolute URL or a relative URL. If you specify a relative URL, the agent forms a full URL with the server IP address or hostname as specified in the option 43 string and uses HTTP as the file transfer protocol.

Also, to install the certificates, the agent expects the device to have an updated system clock. Because, you configure the DHCP server first, you cannot specify the current time in the DHCP server. In such a scenario, an IP address or a URL can be specified as an alternative parameter in the option 43 with the prefix 'Z', which can point the device to a NTP server. The agent synchronizes the clock on the device with the NTP server and then installs the certificates.

DHCP Option based Discovery over an IPv6 Network

Cisco Network PnP uses the DHCP Option 16 and Option 17 for an IPv6 DHCP discovery process.The Option 16 and Option 17 are vendor specific identifiers. These are used by the Cisco Network PnP agent to locate and connect to the Cisco Network PnP server. The DHCP server can be configured to insert an additional information using the vendor specific Option 17. When the DHCP server receives an Option 16 from the device with the string *cisco pnp* , and if it matches the Option 17 string, the server passes the IP address or the hostname of the Cisco PnP server to the requesting device. When the device receives the DHCPv6 response,

the Cisco Network PnP agent extracts the option from the response and identifies the IPv6 address of the Cisco PnP server. Cisco PnP agent uses this IPv6 address to communicate with the Cisco PnP server. To obtain and install the certificate, use the same process explained in the DHCP Option based Discovery over an IPv4 Network section.

The following example shows how to configure a pool (DHCPv6-pool) with the vendor-specific options:

```
ipv6 dhcp pool dhcpv6-pool
 address prefix 2003::/64 lifetime infinite infinite
 vendor-specific 9
  suboption 16 ascii "ciscopnp"
  suboption 17 ascii "5A1D;K4;B3;IFE80::2E0:81FF:FE2D:3799;J6088"
```

DNS-based Discovery

In DNS-based discovery, a DHCP server receives the domain name of the customer network. The domain name is used to create a PnP-specific, fully qualified domain name (FQDN) such as *pnpserver.<domain_name>*. In this method, the customer network resolves this URL to a valid PnP server IP address. Because, there is no mechanism to specify the certificate location, the agent locates the server certificate to initiate the HTTPs connection without manual intervention.

During the system boot up, the device acquires IP network information from a DHCP server along with the domain name. With the customer specific domain name, the Cisco PnP agent creates the following URL *pnpserver.<domain_name>* and looks for the Cisco signed certificate bundle in a default folder of the server *<domain_name>/ca/trustpool/cabundle.p7b*.

If the certificate bundle is available at the specified location, then the agent:

1. Downloads the bundle from the file server to the device.
2. Checks the signature of the downloaded bundle to ensure it has genuine Cisco signature.
3. Installs the certificates on the device.

If the certificate bundle is not available at the specified location, the PnP agent use a predefined URL,*pnpcertserver.<domain_name>* and looks for the Cisco signed certificate bundle in the default folder of the server, *<domain_name>/ca/trustpool/cabundle.p7b*.

If the certificates are available at the specified location, then the agent performs the steps specified above to install the certificates.

After the certificates are installed and the server discovery is complete, the agent initiates the HTTPs connection with the server at the URL, *pnpserver.<domain_name >* without any additional configuration. During the HTTPs handshake, the device uses the certificates that are installed from the bundle to validate the server certificate.

Also, to install the certificates, the agent expects the device to have an updated system clock. Because, you configure the DHCP server first, you cannot specify the current time in the DHCP server. In such a scenario, the agent uses a predefined URL, *pnpntpserver.<domain_name>* which needs to be mapped to a NTP sever to synchronize the clock on the device, and then installs the certificates.

However, if the certificate is not present at either URL, the Cisco PnP agent will fall back and establish the HTTP connection to the server using the created FQDN pnpserver.<domain_name>. With this workflow, the agent expects the server to use the certificate-install service to install the self-signed certificates first and then start the provisioning steps.

DNS-based Discovery over an IPv6 Network

To enable DNS-based discovery over an IPv6 network:

Procedure

Step 1 Configure the DNS server with an IPv6 option. To enable the Cisco Network PnP DNS discovery, configure the DNS server as shown in this example:

```
ip host pnpntpserver.domain.com 2001::1
ip host pnptrustpool.domain.com 2001::2
ip host pnpserver.domain.com   2001::3
```

Step 2 DHCPv6 server is discovered through DHCP bootstrap process. The following example shows how to configure the DHCP server:

```
ipv6 unicast routing
ipv6 cef

ipv6 dhcp pool test
dns-server 2001::4
domain-name example.com
```

The device sends the DHCPv6 packets to the server over an IPv6 network. After receiving the DHCPv6 packets, the DNS server information and the domain-name are returned to the device as Option 23 and Option 24 respectively.

Step 3 Configure the NTP server. The following example shows how to configure the NTP server:

```
ntp master 1
```

Note
Similarly, the device NTP configuration should use the NTPv4 option.

Step 4 Host the trustpool server on an IPv6 network. Trustpool is supported only on DHCP Options T and Z. If the Option T is configured, specify the URL of the trustpool CA bundle. If the Option Z is configured, specify the NTP server IP address.

Note
When the Cisco Network PnP agent attempts to download the trustpool bundle over HTTP by using an IPv6 option, the trustpool server should support HTTP over an IPv6 network. Also, the clock must be syncronized before configuring the trustpool.

Step 5 Host the Cisco Network PnP server on an IPv6 network.

Cisco Cloud Redirection over an IPv4 and IPv6 Network

Cisco Cloud Redirection service supports Cisco Network PnP zero-touch discovery. It is supported on both IPv4 and IPv6 based Cisco Cloud discovery.

Note Some of the Cisco PnP devices may have root certificate embedded in the devices. These devices will communicate with the CCO server using HTTPS from the beginning. If the device does not have the embedded certificate then the legacy behavior is initiated.

When the device boots up without any start-up configuration or authentication certificates, and if the DHCP and DNS discovery fails, the device tries to contact the Cisco Cloud server at *devicehelper.cisco.com*.

If the *devicehelper.cisco.com* is reachable, the Cisco Network PnP agent downloads the trustpool bundle and establishes a secure HTTP connection with the Cisco Cloud Redirection service. When the device tries the Cisco cloud discovery for the first time, Cisco Network PnP agent downloads the trustpool from this location *devicehelper.cisco.com/ca/trustpool* and saves it to the local flash memory. This location is shared with a Public Key Infrastructure for a trustpool installation. If the Cisco cloud discovery fails, trustpool bundle is retained in the flash memory and Cisco Network PnP checks for a copy of the trustpool bundle in the local device flash memory. If the copy is not available in the local flash memory, it retries to download the trustpool bundle from this location *devicehelper.cisco.com/ca/trustpool download*.

Cisco Network PnP agent sends a HTTPS hello message to the Cisco cloud. The Cisco Network PnP redirection service running at Cisco cloud server replies to the HTTP request. A Cisco cloud server PnP profile is created on the device as shown in this example.

```
pnp profile pnp_cco_profile
transport https host devicehelper.cisco.com port 443
```

After the Cisco cloud profile is created, the device sends a work-information message with its unique device identifier information to the Cisco cloud server. Cisco Cloud Redirection service sends a redirection non-backoff PnP request with the Cisco Network PnP server information. It can be an IPv4 address, IPv6 address, or a hostname. When the redirection is successful, the following redirection profile is configured on the device.

```
pnp profile pnp_redirection_profile
transport https ipv4 172.19.153.133 port 443
```

If the non-backoff PnP request is not received within default wait time, Cisco Network PnP discovery process continues with the next discovery mechanism.

Cisco Network PnP Discovery Over 4G Interface

Cisco Network PnP over 4G interface is available on platforms that have 4G NIMs and running Cisco IOS XE.When a device with an activated SIM card boots up, the 4G interface is activated and used for the Cisco Network PnP cloud discovery process. When a device without an activated SIM card boots up, the non-4G interfaces are preferred for the discovery process. Cisco Network PnP cloud discovery over 4G interfaces is attempted when the non-4G interfaces are not available or if the Cisco Network PnP discovery does not succeed on the non-4G interfaces. When the device has multiple 4G interfaces with active SIM cards, the Cisco Network PnP tries the cloud discovery on all the 4G interfaces one after the other until one of them succeeds.



Note To use the 4G interface for the Cisco Network PnP discovery, the 4G NIMs should have an activated SIM card on it.

Cisco Network PnP Cloud discovery over 4G interfaces works when all the 4G interfaces are activated during the device bootup by default. In the absence of a startup configuration, the device attempts to bring up the 4G inutrafec by default and attempts Cicso PnP over cloud. After the device is redirected, the device connects to the Cisco Network PnP server and downloads the appropriate image and configuration to the device.

Note The DNS server is available as part of the 4G network and the cloud portal should be programmed to redirect the calling device to an appropriate Cisco Network PnP server for provisioning the device. Currently, Cisco Network PnP support over 4G interface uses only the IPv4 network.

Ensure that the configuration pushed through the Cisco Network PnP server contains a route to Cisco Network PnP server over the 4G interface. This can be a default route and should retain the Cisco Network PnP agent and server communication to continue to work over the 4G interface, after the provisioning is completed.

Cisco Network PnP Discovery over a Management Interface

Cisco Network PnP Agent supports discovery and four-way handshake over a management interface with a default VPN Routing/Forwarding (VRF). To send and receive the DHCP traffic over an VRF interface, you have to configure the IOS DHCP server . This feature helps the new devices to access the Cisco Network PnP features when only the management interface is active.

When the device boots up, the management interface under the default VRF is assigned an IP address though DHCP. This interface establishes a connection to a Cisco Network PnP server and the Cisco Network PnP agent on the device records this information (VRF name and source interface). This information is used for future PnP communication with the Cisco Network PnP server. In this case, the Cisco PnP profile that is created on the device will have an extra keyword **VRF** attached to it.

Cisco PnP over an EtherChannel

When you deploy an access switch by using the Cisco Network Plug and Play, the existence of LACP EtherChannels on the provisioned switch (which acts as trunk) does not allow you to configure the device. When the access device tries to connect through the provisioned switch over an L2 EtherChannel using LACP, it breaks the connectivity. Since the configuration does not exists on the access device, the access device cannot bring up the EtherChannel with the switch. This results in keeping the EtherChannel ports in suspended state and breaks the L2 connectivity. Cisco Network PnP Agent detects the presence of EtherChannels and auto-configures the EtherChannel on the device to bring up the Layer 2 connectivity automatically for the day-zero configuration.

Security Methods for Post-PnP Discovery Process

This section explains the methods provided by the Cisco PnP agent which can be, used by the Cisco PnP server to secure the client-server communication after completing the discovery process. This section includes the following topic:

• Certificate Install Service, on page 233

Certificate Install Service

The Cisco PnP agent provides a mechanism to manage SSL certificates on the device by providing the certificate-install service to the Cisco PnP server. The certificate-install service provides a simple XML to install the server's self-signed certificates or certificates signed by standard CA certificates on the device, before initiating an HTTPs connection. The certificate-install service also provides an option to install the

client SSL certificate and instruct the device to use the same SSL certificate during the next device authentication process.

SUDI-based PnP Application Level Authentication

The SSL communication ensures encryption of the data packets exchanged between the server and the device, but does not provide a solution to authenticate the device.

To ensure that the server is talking to a genuine Cisco device, the agent uses the built-in Secure Unique Device Identifier (SUDI) certificate support on the device. SUDI is a X.509 compliant device certificate burnt into the device's secured chip (ACT2) during the manufacture time. The SUDI certificate contains the device's serial number, private-public keys, and the Cisco CA signature. The agent provides the following mechanisms that can be used by the server to authenticate the device as a genuine Cisco device:

- SUDI-based Client Certificate Validation, on page 234
- SUDI-based Serial Number, on page 234

SUDI-based Client Certificate Validation

Before the agent initiates an HTTPs connection with the server, the agent checks whether the device has a built-in SUDI certificate. If the device has a certificate, then the agent sends the SUDI certificate to the client during the SSL handshake for validating. Optionally, the HTTPs server may choose to validate the device using the SUDI certificate during the SSL handshake. After validating, the HTTPs server allows the device to connect to the server. To validate the device's SUDI certificate, the server should use Cisco CA to complete the validation.

SUDI-based Serial Number

If the device is loaded with SUDI certificate, the PnP agent reads the serial number from the SUDI certificate and presents the same information as an additional tag in the work-request body for all communication with the server. To achieve this, the following optional tag is added in the work-info message, which goes out from the device in every work-request. This field is optional and does not show up for devices that does not have SUDI certificate.

There is no change in the existing UDI mechanism that is read from the chassis inventory. The agent continues to be backwards compatible by sending the chassis UDI as the primary identifier. The server can use the additionally provided SUDI-based serial number to authenticate the device and then continue to use the primary UDI. For the devices without a SUDI certificate, the agent does not send this additional SUDI-based serial number. Therefore, the server should continue with the primary UDI for authentication and further communication.

There is no mechanism available to read the SUDI-based serial number from member hardware and there is no change in how UDI is read from other members on a stack or HA unit. The agent will continue to read the UDI from all the hardware units as it does presently.

SUDI-Based Device Authentication

In SUDI-based device authentication, the agent checks whether the device has a built-in SUDI certificate at the boot-up time. If the device is loaded with the SUDI certificate, the agent provides a new PnP service, which allows the server to help the device to identify itself. The availability of this new service depends on the presence of the SUDI certificate and is listed in the agent's capability service.

Along with the above change in the capability-service, the agent adds an additional field under the hardware-info section of the device-info response, to specify and check whether the SUDI certificate is built into the device.

After, the agent initiates an HTTPs connection with the server and sends a work-request, the server should be able to use the device authentication service for a challenge request-response. The device authentication service requires a minimum of one field to generated a string by the server. Optionally, the server can send a list of encryptions and hashing methods that it can support. The agent checks whether it has the capability to use any of the listed encryption methods specified by the server, uses the encryption method and sends a notification to the server. If the agent does not have the capability to use any of the methods specified by the server, then the agent responds with an error message.

When the server sends a device authentication service request to the agent, the agent does the following:

1. Uses one of the specified encryption and hashing methods.
2. If the agent does not have capability to use one of the specified encryption and hashing methods, the agent responds with an error message.
3. Encrypts the challenge string provided by the server using the private key using the PKI APIs.
4. Sends a response back with the following:
   a. Cipher text
   b. Methods used to cipher
   c. Certificate (SUDI or client installed certificate)

After, the server receives the above response from the device, the server does the following:

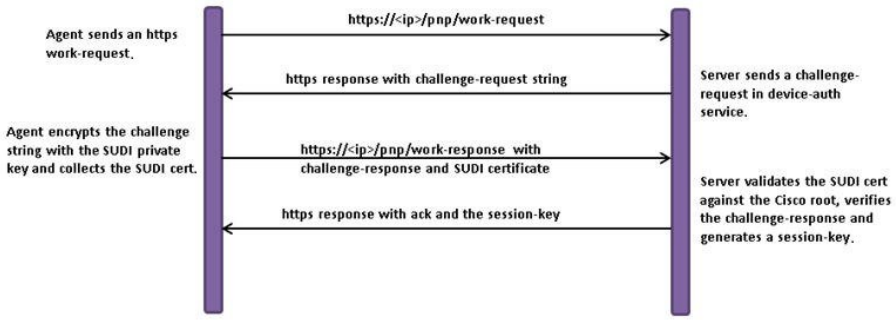
1. Verifies the SUDI or the client certificate against the Cisco or customer CA.
2. Decrypts the cipher-string using the public key that is available in the SUDI or client certificate.
3. Verifies whether the deciphered string matches the original version.
4. Generates a session key (string) and sends it back to the device as an acknowledgment.

After the agent receives the final acknowledgment from the server with the session-key, it associates the corresponding profile with the provided session-key and sends it to the server as an attribute in the root PnP section of all the subsequent messages that the agent sends.

The server validates the session-key before sending any message from the device. Optionally, the server maintains a timer for the session-keys and moves to invalid status when the timer expires. If the agent sends a message with an expired session-key, the server repeats the device authentication process and generate a new session-key before sending to the same device again. If the device sends a request without any session-key, then the server performs the device authentication process and generates a new session-key before sending to the same device.

The following figure displays the message sequence between the agent and the server to accomplish the device authentication using the SUDI certificate.

Figure 41: Message Sequence



How to Configure Cisco Network Plug and Play Agent

Configuring Cisco Network Plug and Play Agent Profile

Perform the following task to create a Cisco Network Plug and Play agent profile:

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	**pnp profile** *profile-name* **Example:** `Device(config)# pnp profile test-profile-1`	Creates a PnP agent profile and enters the PnP profile initialization mode. • String of alphanumeric characters that specify a name for the PnP agent profile. Profile names cannot be duplicated.
Step 4	**end** **Example:** `Device(config-pnp-init)# end`	Exits the PnP profile initialization mode and returns to privileged EXEC mode.

Configuring Network Plug and Play Agent Device

Perform the following task to create a Cisco Network Plug and Play agent device:

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	**pnp profile** *profile-name* **Example:** `Device(config)# pnp profile test-profile-1`	Creates a PnP agent profile and enters the PnP profile initialization mode. • String of alphanumeric characters that specify a name for the PnP agent profile. Profile names cannot be duplicated.
Step 4	**device** {**username** *username* } {**password {0 | 7}** *password*} **Example:** `Device(config-pnp-init)# device username sjohn password 0 Tan123`	Configures the PnP agent on the device. • Establishes a username and password based authentication system. • *username*—User ID • *password*—Password that a user enters • **0**—Specifies that an unencrypted password or secret (depending on the configuration) follows. • **7**—Specifies that an encrypted (hidden) password follows.
Step 5	**end** **Example:** `Device(config-pnp-init)# end`	Exits the PnP profile initialization mode and returns to privileged EXEC mode.

Configuring Cisco Network Plug and Play Reconnect Factor

Perform the following task to configure the time to wait before attempting to reconnect a session in either fixed-interval-backoff, exponential-backoff, or random-exponential-backoff mode:

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	**pnp profile** *profile-name* **Example:** `Device(config)# pnp profile test-profile-1`	Creates a PnP agent profile and enters the PnP profile initialization mode. • String of alphanumeric characters that specify a name for the PnP agent profile. Profile names cannot be duplicated.
Step 4	**reconnect** [*pause-time* [*exponential-backoff-factor* [**random**]]] **Example:** `Device(config-pnp-init)# reconnect 100 2 random`	Specifies the time for the PnP agent initiator profile to wait before attempting to reconnect a session. • The pause-time value is the time to wait, in seconds, before attempting to reconnect after a connection is lost. The range is from 1 to 2000000. The default is 60. • Exponential backoff factor value is the value that triggers the reconnect attempt exponentially. The range is from 2 to 9.
Step 5	**end** **Example:** `Device(config-pnp-init)# end`	Exits the PnP profile initialization mode and returns to privileged EXEC mode.

Configuring Cisco Network Plug and Play HTTP Transport Profile

Perform the following task to create a HTTP transport profile of the Cisoc Plug and Play agent manually on a device.

Both IPv4 and IPv6 addresses can be used for PnP server IP configuration. Alternately, a hostname can also be used in the configuration to connect to the PnP server.

Every profile can have one primary server and a backup server configuration. The Cisco PnP agent attempts to initiate a connection with the primary server first and if it fails, it will try the backup server. If the backup server fails, the Cisco PnP agent will attempt to connect to the primary server again. This will continue until a connection is established with one of the servers.

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	**pnp profile** *profile-name* **Example:** `Device(config)# pnp profile test-profile-1`	Creates a PnP agent profile and enters the PnP profile initialization mode. • String of alphanumeric characters that specify a name for the PnP agent profile. Profile names cannot be duplicated.
Step 4	**transport http host** *host-name* [**port** *port-number*] [**source** *interface-type*] **Example:** `Device(config-pnp-init)# transport http host hostname-1 port 1 source gigabitEthernet 0/0/0`	Creates a HTTP transport configuration for the PnP agent profile based on the hostname of the server on which the PnP agent is deployed. • The value of the host specifies the host name, port, and source of the server. • The value of the port-number specifies the port that is used. • The value of the interface-type specifies the interface on which the agent is connected to the server.
Step 5	**transport http ipv4** *ipv4-address* [**port** *port-number*] [**source** *interface-type*] **Example:** `Device(config-pnp-init)# transport http ipv4 10.0.1.0 port 221 source gigabitEthernet 0/0/0`	Creates a HTTP transport configuration for the PnP agent profile based on the IPv4 address of the server on which the PnP agent is deployed.
Step 6	**transport http ipv6** *ipv6-address* [**port** *port-number*] [**source** *interface-type interface-number*] **Example:** `Device(config-pnp-init)# transport http ipv6 2001:DB8:1::1 port 331 source gigabitEthernet 0/0/1`	Creates a HTTP transport configuration for the PnP agent profile based on the IPv6 address of the server on which the PnP agent is deployed.
Step 7	**end** **Example:**	Exits the PnP profile initialization mode and returns to privileged EXEC mode.

	Command or Action	Purpose
	`Device(config-pnp-init)# end`	

Configuring Cisco Network Plug and Play HTTPS Transport Profile

Perform the following task to create a HTTP Secure (HTTPS) transport profile of the Cisco Network Plug and Play agent manually on a device.

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	**pnp profile** *profile-name* **Example:** `Device(config)# pnp profile test-profile-1`	Creates a PnP agent profile and enters the PnP profile initialization mode. • String of alphanumeric characters that specify a name for the PnP agent profile. Profile names cannot be duplicated.
Step 4	**transport https host** *host-name* [**port** *port-number*][**source** *interface-type*][**localcert** *trustpoint-name*][**remotecert** *trustpoint-name*] **Example:** `Device(config-pnp-init)# transport https host example.com port 231 source gigabitEthernet 0/0/0 localcert abc remotecert xyz`	Creates a HTTPS transport configuration for the PnP agent profile based on the hostname of the server on which the PnP agent is deployed. • The value of *localcert* specifies the trustpoint used for client-side authentication during the transport layer security (TLS) handshake. • The value of *remotecert* specifies the trustpoint used for server certificate validation. **Note** Configure the trustpoint-name using the **crypto pki trustpoint** command.
Step 5	**transport https ipv4** *ipv4-address* [**port** *port-number*][**source** *interface-type*][**localcert** *trustpoint-name*][**remotecert** *trustpoint-name*] **Example:**	Creates a HTTPS transport configuration for the PnP agent profile based on the IPv4 address of the server on which the PnP agent is deployed.

	Command or Action	Purpose
	`Device(config-pnp-init)# transport https ipv4 10.0.1.0 port 221 source gigabitEthernet 0/0/0 localcert abc remotecert xyz`	
Step 6	**transport https ipv6** *ipv6-address* [**port** *port-number*][**source** *interface-type interface-number*][**localcert** *trustpoint-name*][**remotecert** *trustpoint-name*] **Example:** `Device(config-pnp-init)# transport https ipv6 2001:DB8:1::1 port 331 source gigabitEthernet 0/0/1 localcert abc remotecert xyz`	Creates a HTTPS transport configuration for the PnP agent profile based on the IPv6 address of the server on which the PnP agent is deployed.
Step 7	**end** **Example:** `Device(config-pnp-init)# end`	Exits the PnP profile initialization mode and returns to privileged EXEC mode.

Configuring Backup Cisco Network Plug and Play Device

Perform the following task to create a backup profile and to enable or disable Cisco Network Plug and Play agent manually on a device:

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	**pnp profile** *profile-name* **Example:** `Device(config)# pnp profile test-profile-1`	Creates a PnP agent profile and enters the PnP profile initialization mode. • String of alphanumeric characters that specify a name for the PnP agent profile. Profile names cannot be duplicated.
Step 4	**backup device** {**username** *username* } {**password {0 | 7}** *password*} **Example:**	Configures the PnP agent backup profile on the device. • Establishes a username and password based authentication system.

	Command or Action	Purpose
	`Device(config-pnp-init)# backup device username sjohn password 0 Tan123`	• *username*-User ID • *password*-Password that a user enters • 0—Specifies that an unencrypted password or secret (depending on the configuration) follows. • 7—Specifies that a hidden password follows.
Step 5	**end** **Example:** `Device(config-pnp-init)# end`	Exits the PnP profile initialization mode and returns to privileged EXEC mode.

Configuring Backup Cisco Network Plug and Play Reconnect Factor

Perform the following task to configure backup reconnection of the Cisco Network Plug and Play (PnP) agent to the server in either fixed-interval-backoff, exponential-backoff, or random-exponential-backoff manner :

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	**pnp profile** *profile-name* **Example:** `Device(config)# pnp profile test-profile-1`	Creates a PnP agent profile and enters the PnP profile initialization mode. • String of alphanumeric characters that specify a name for the PnP agent profile. Profile names cannot be duplicated.
Step 4	**backup reconnect** [*pause-time* [*exponential-backoff-factor* [**random**]]] **Example:** `Device(config-pnp-init)# backup reconnect 100 2 random`	Specifies the time for the PnP agent initiator profile to wait before attempting to reconnect a session. • The pause-time value is the time to wait, in seconds, before attempting to reconnect after a connection is lost. The range is from 1 to 2000000. The default is 60.

	Command or Action	Purpose
		• Exponential backoff factor value is the value that triggers the reconnect attempt exponentially. The range is from 2 to 9.
Step 5	**end** **Example:** `Device(config-pnp-init)# end`	Exits the PnP profile initialization mode and returns to privileged EXEC mode.

Configuring Backup Cisco Network Plug and Play HTTP Transport Profile

Perform the following task to create a backup HTTP transport profile of the Cisco Network Plug and Play agent manually on a device.

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	**pnp profile** *profile-name* **Example:** `Device(config)# pnp profile test-profile-1`	Creates a PnP agent profile and enters the PnP profile initialization mode. • String of alphanumeric characters that specify a name for the PnP agent profile. Profile names cannot be duplicated.
Step 4	**backup transport http host** *host-name* [**port** *port-number*] [**source** *interface-type*] **Example:** `Device(config-pnp-init)# backup transport http host hostname-1 port 1 source gigabitEthernet 0/0/0`	Creates a backup HTTP transport configuration for the PnP agent profile based on the hostname of the server on which the PnP agent is deployed. • The value of the host specifies the host name, port, and source of the server. • The value of the port-number specifies the port that is used. • The value of the interface-type specifies the interface on which the agent is connected to the server.

	Command or Action	Purpose
Step 5	**backup transport http ipv4** *ipv4-address* [**port** *port-number*] [**source** *interface-type*] **Example:** `Device(config-pnp-init)# backup transport http ipv4 10.0.1.0 port 221 source gigabitEthernet 0/0/0`	Creates a backup HTTP transport configuration for the PnP agent profile based on the IPv4 address of the server on which the PnP agent is deployed.
Step 6	**backup transport http ipv6** *ipv6-address* [**port** *port-number*] [**source** *interface-type interface-number*] **Example:** `Device(config-pnp-init)# backup transport http ipv6 2001:DB8:1::1 port 331 source gigabitEthernet 0/0/1`	Creates a backup HTTP transport configuration for the PnP agent profile based on the IPv6 address of the server on which the PnP agent is deployed.
Step 7	**end** **Example:** `Device(config-pnp-init)# end`	Exits the PnP profile initialization mode and returns to privileged EXEC mode.

Configuring Backup Cisco Network Plug and Play HTTPS Transport Profile

Perform the following task to create a backup HTTPS transport profile of the Cisco Network Plug and Play agent manually on a device.

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	**pnp profile** *profile-name* **Example:** `Device(config)# pnp profile test-profile-1`	Creates a PnP agent profile and enters the PnP profile initialization mode. • String of alphanumeric characters that specify a name for the PnP agent profile. Profile names cannot be duplicated.

	Command or Action	Purpose
Step 4	**backup transport https host** *host-name* [**port** *port-number*][**source** *interface-type*][**localcert** *trustpoint-name*][**remotecert** *trustpoint-name*] **Example:** `Device(config-pnp-init)# backup transport https host example.com port 231 source gigabitEthernet 0/0/0 localcert abc remotecert xyz`	Creates a HTTPS backup transport configuration for the PnP agent profile based on the hostname of the server on which the PnP agent is deployed. • The value of *localcert* specifies the trustpoint used for client-side authentication during the transport layer security (TLS) handshake. • The value of *remotecert* specifies the trustpoint used for server certificate validation.
Step 5	**backup transport https ipv4** *ipv4-address* [**port** *port-number*][**source** *interface-type*][**localcert** *trustpoint-name*][**remotecert** *trustpoint-name*] **Example:** `Device(config-pnp-init)# backup transport https ipv4 10.0.1.0 port 221 source gigabitEthernet 0/0/0 localcert abc remotecert xyz`	Creates a HTTPS backup transport configuration for the PnP agent profile based on the IPv4 address of the server on which the PnP agent is deployed.
Step 6	**backup transport https ipv6** *ipv6-address* [**port** *port-number*][**source** *interface-type interface-number*][**localcert** *trustpoint-name*][**remotecert** *trustpoint-name*] **Example:** `Device(config-pnp-init)# backup transport https ipv6 2001:DB8:1::1 port 331 source gigabitEthernet 0/0/1 localcert abc remotecert xyz`	Creates a HTTPS backup transport configuration for the PnP agent profile based on the IPv6 address of the server on which the PnP agent is deployed.
Step 7	**end** **Example:** `Device(config-pnp-init)# end`	Exits the PnP profile initialization mode and returns to privileged EXEC mode.

Configuring Cisco Network Plug and Play Agent Tag

Perform the following task to create a Cisco Network Plug and Play agent tag information:

Procedure

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.

	Command or Action	Purpose
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**pnp tag** *tag-name* **Example:** `Device(config)# pnp tag xyz`	Use the **pnp tag** command to configure the tag for the device. The neighboring Cisco devices will receive this tag information through Cisco Discovery Protocol (CDP). **Note** If there is an existing tag for the device, the tag name can be only changed when the xml schema is sent by the PnP server to change the tag name. The tag name cannot be overwritten. • String of alphanumeric characters that specify a name for the PnP agent tag.
Step 4	**end** **Example:** `Device(config)# end`	Exits the global configuration mode and returns to privileged EXEC mode.

Troubleshooting and Debugging

To run the debugging on the Cisco Network Plug and Play (server, start the server, configure the PnP profile and PnP transport. For example, start the service interaction between PnP agent and PnP server.

You can capture the debugs by executing the **debug pnp service** command. When you report a problem, collect all the pnp* files in the PnP agent flash" to the guide.

Note To collect Cisco Plug and Play server log, see the Cisco Application Policy Infrastructure Controller Enterprise Module Deployment Guide.

To troubleshoot the device, server and Cisco PnP Agent, use the following commands:

Table 18: Troubleshooting the Device, Server, and Cisco PnP Agent

Command	Description
dir nvram	Use this command to ensure tht the device does not have left over certificates.
ping vrf interface-name *<controller_ip>*	Use this command to ensure the the device can ping the controller.

Command	Description
show auto install trace	Use this command to view auto install trace log.
show boot	Use the show boot command to display the current value for the BOOTLDR variable.
show cdp neighbor	Use this command to display all CDP neighbors.
Show crypto pki trustpoint	Use this command to view the PKI trustpoint.
Show crypto pki trustful	Use this command to view the PKI trustful.
show ip interface brief	Use this command to view a summary of the router interfaces.
show ipv6 interface brief	Use this command to display the IPv6 interfaces.
show run | inc pnp	Uset this command to ensure that only one PnP profile installed
show pnp trace	Use this command to ensure that the device does not have start-up configuration.
show pnp tech	Use this command to view active connections for the Cisco Plug and Play IOS Agent.
show vlan	Use this command to view the VLAN information.
show ntp status	Use this command to view the NTP status.
show version	Use this command to ensure that the device is running the latest CCO image

Glossary

PnP Agent: An embedded agent on the device to automate deployment process

PnP Helper Applications: Applications on smart phones and personal computers that facilitate deployment. PnP helper applications are not specific to a customer or device and can be used in any deployment scenario. May be needed in limited scenarios

PnP Protocol: Protocol between the PnP agent and PnP server. This is an open protocol allowing third-party development of PnP servers

PnP Server: A central server that manages and distributes deployment information (images, configurations, files, and licenses) for the devices being deployed. Cisco Network Plug and Play server provides a north bound interface for management applications and communicates with the PnP agents on the devices using the PnP protocol.

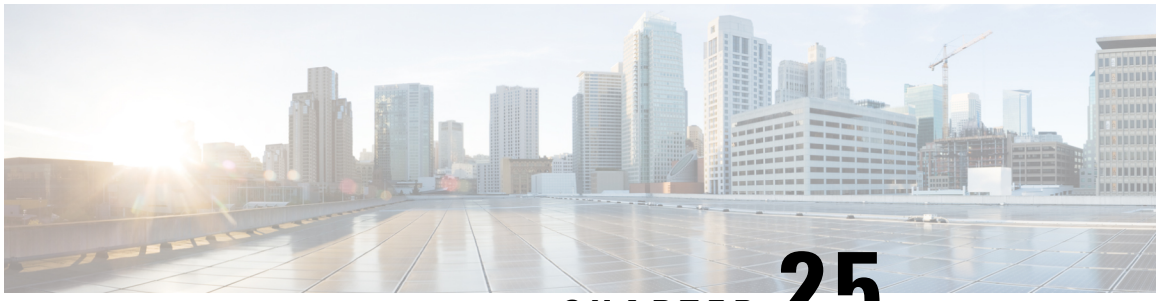
Additional References for Open Plug-n-Play Agent

Related Documents

Related Topic	Document Title
Cisco IOS commands	Cisco IOS Master Command List, All Releases
PnP commands: Complete command syntax, command mode, command history, defaults, usage guidelines, and examples	Cisco IOS PnP Command Reference
Cisco Network Plug and Play solution	Solution Guide for Cisco Network Plug and Play.
How to use the Cisco Network Plug and Play in the APIC-EM to configure Cisco network devices.	Configuration Guide for Cisco Network Plug and Play on Cisco APIC-EM.
How to deploy the APIC-EM.	Cisco Application Policy Infrastructure Controller Enterprise Module Deployment Guide.
Getting started with the APIC-EM.	Cisco APIC-EM Quick Start Guide.

MIBs

MIB	MIBs Link
• CISCO-BULK-FILE-MIB • CISCO-DATA-COLLECTION-MIB • CISCO-PROCESS-MIB • Expression-MIB	To locate and download MIBs for selected platforms, Cisco software releases, and feature sets, use Cisco MIB Locator found at the following URL: http://www.cisco.com/go/mibs



CHAPTER 25

Cellular Boot Time Improvements

- Cellular boot time improvements, on page 249

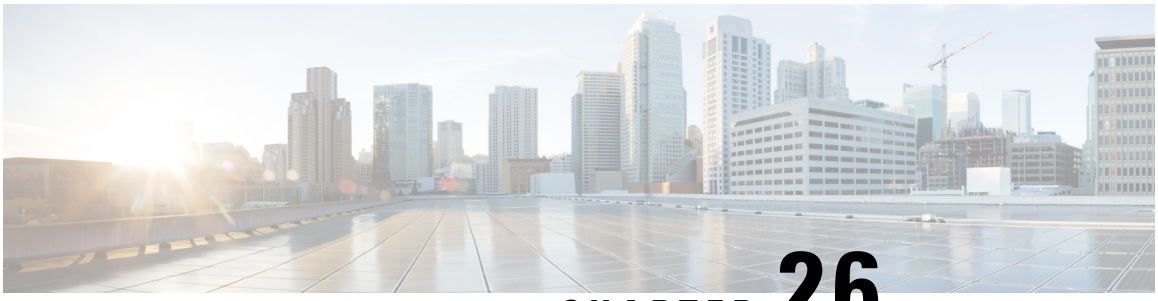
Cellular boot time improvements

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 26

CLI Output for the FN980 5G Modem

• Change in CLI output for the FN980 5G modem, on page 251

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 0/1/0 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 0/1/0 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) = 898600005020001807 22
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 0/1/0 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>

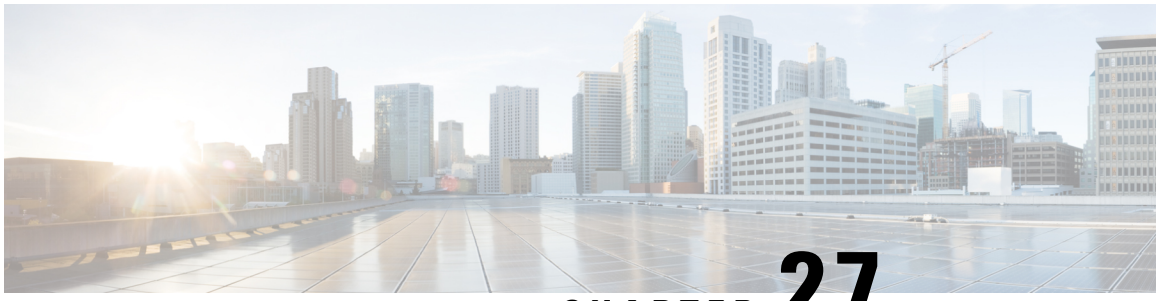
3G bands supported by modem:
Index:
  23 - UMTS Band 1: 2100 MHz (IMT)
  24 - UMTS Band 2: 1900 MHz (PCS A-F)
  26 - UMTS Band 4: 1700 MHz (AWS A-F)
  27 - UMTS Band 5: US 850 MHz  (CLR)
  50 - UMTS Band 8: 900 MHz  (E-GSM)
  51 - UMTS Band 9: Japan 1700 MHz
  61 - UMTS Band 19: 800 MHz (800 Japan)
3G 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 27

Supervisory Control and Data Acquisition

This chapter contains the following sections:

- SCADA Overview, on page 253
- Role of the IR1101, on page 254
- Key Terms, on page 254
- Protocol Translation Application, on page 255
- Prerequisites, on page 255
- Guidelines and Limitations, on page 256
- Default Settings, on page 256
- Configuring Protocol Translation, on page 257
- Configuring the T101 Protocol Stack, on page 258
- Configuring the T104 Protocol Stack, on page 260
- SCADA enhancement for TNB, on page 263
- Configuration Example, on page 264
- Yang Data Model Support for Scada, on page 265
- Configuring the DNP3 Protocol Stacks, on page 267
- DNP3 Enhancement, on page 271
- Starting and Stopping the Protocol Translation Engine, on page 271
- Verifying Configuration, on page 272
- Debug Commands, on page 273

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 IR1101, line is 0/2/0 same as the Async interface.

Role of the IR1101

In the network, the Control Center always serves as the master in the network when communicating with the IR1101. The IR1101 serves as a proxy master station for the Control Center when it communicates with the RTU.

The IR1101 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 IR1101 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 IR1101:

- Channel–A channel is configured on each IR1101 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 primary or secondary 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 IR1101:

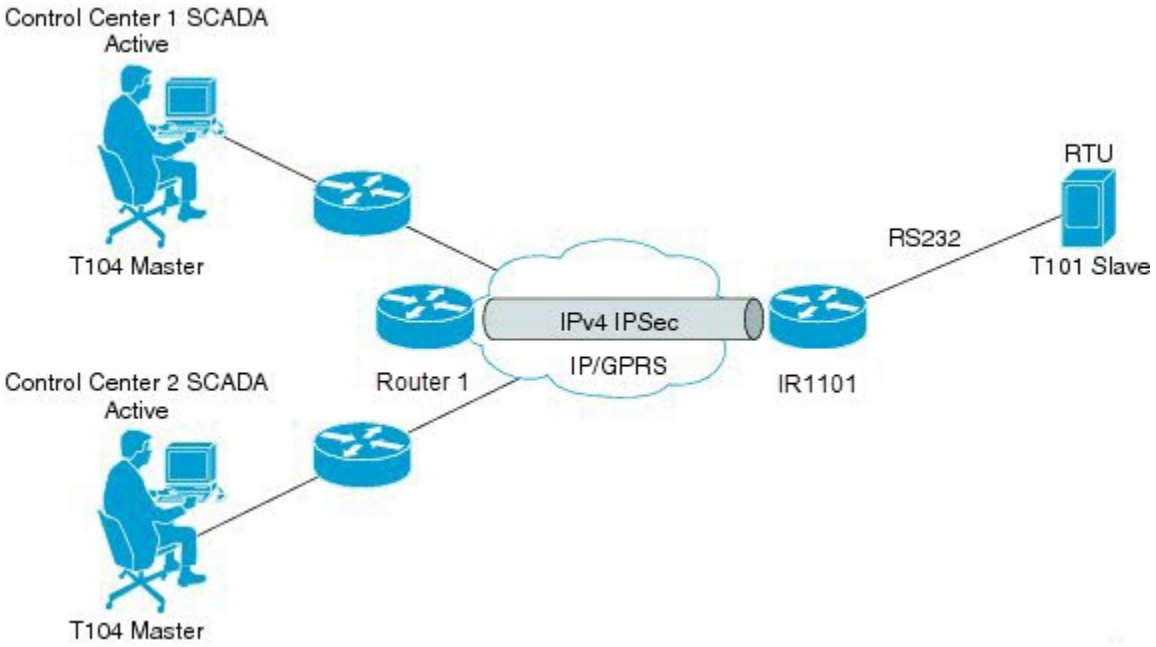
- Channel–A channel is configured on the IR1101 serial port interface to provide a connection to a single RTU for each IP connection to a remote Control Center. Each connection transports a single DNP3 serial (RTU) or DNP3 IP (Control Center) protocol stack.
- Link Address–Refers to the device or station address.
- Sessions–Represents a single connection to a remote site.

Protocol Translation Application

In the figure below, the IR1101 (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 IR1101 connects to the RTU (slave) through a RS232 connection. To protect the traffic when forwarded over public infrastructures (for example, cellular), the IR1101 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 IR1101 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 42: Routers Within a SCADA System



Prerequisites

RTUs must be configured and operating in the network.

For each RTU that connects to the IR1101, 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 IR1101, 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 sessions 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 IR1101 operating with Protocol Translation, please review the section on Starting and Stopping the Protocol Translation Engine, on page 271.

Enabling the IR1101 Serial Port and SCADA Encapsulation

Before you can enable and configure Protocol Translation on the IR1101, you must first enable the serial port on the IR1101 and enable SCADA encapsulation on that port.

Before you begin

Determine availability of serial port on the IR1101.

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
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.

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 258
- Configuring the T104 Protocol Stack, on page 260
- Starting and Stopping the Protocol Translation Engine, on page 271

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 IR1101 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 IR1101 serial interface on which the system sends its T101 protocol traffic.

	Command or Action	Purpose
		slot –Value of 0 *port* –Value of 2 *interface* –Value of 0
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 . *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.

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)#
```

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 255)

Enable the serial port and SCADA encapsulation. (See Enabling the IR1101 Serial Port and SCADA Encapsulation, on page 257)

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.

	Command or Action	Purpose
		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.
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.

	Command or Action	Purpose
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.
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.

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)#
```

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.

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 IR1101.

```
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 IR1101:

```
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
```

Note IOA addresses obtained from T101 side are sent to T104 side without any modification by the SCADA Gateway

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 .

Scada Yang Models

There are two feature modules available for Scada that belong to the main Cisco-IOS-XE-native model:

- Cisco-IOS-XE-scada-gw.yang

This module contains a collection of YANG definitions for Scada Gateway Configuration commands.

- Cisco-IOS-XE-scada-gw-oper.yang

This module contains a collection of YANG definitions for Scada Gateway operational data.

There are eight dependent modules (also belonging to the main Cisco-IOS-XE-native model), that should be imported for the Scada models to work. The following section shows the Scada Yang Models list, configuration CLI commands, and the dependent modules that each feature module covers.

Cisco-IOS-XE-scada-gw

This module has the following corresponding CLI commands:

```
(config)# scada-gw protocol t101
(config-t101)# channel <channel-name>
(config-t101)# bind-to-interface <interface-name>
(config-t101)# link-mode <link-mode>
(config-t101)# link-addr-size <size>
(config-t101)# day-of-week <enable>
(config-t101)# session <session_name>
(config-t101)# attach-to-channel <channel-name>
(config-t101)# cot-size <size>
(config-t101)# common-addr-size <size>
(config-t101)# info-obj-addr-size <size>
(config-t101)# link-addr <addr>
(config-t101)# request
(config-t101)# sector <sector_name>
(config-t101)# attach-to-session <session-name>
(config-t101)# asdu-addr <addr>
(config-t101)# request
(config)# scada-gw protocol t104
(config-t104)# channel <channel-name>
(config-t104)# tcp connection
(config-t104)# to-timeout <value>
(config-t104)# t1-timeout <value>
(config-t104)# t2-timeout <value>
(config-t104)# t3-timeout <value>
(config-t104)# k-value <value>
(config-t104)# w-value <value>
(config-t101)# day-of-week <enable>
(config-t101)# send-ei <enable>
(config-t104)# session <session_name>
(config-t104)# attach-to-channel <channel_name>
(config-t104)# sector <sector_name>
(config-t104)# attach-to-session <session-name>
config-t104)# map-to-sector <sector-name>
(config) scada-gw enable
```

The Cisco-IOS-XE-scada-gw module has the following 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

Cisco-IOS-XE-scada-gw-oper

This module has the following corresponding Cli commands:

```
# show scada statistics
# show scada tcp
```

The Cisco-IOS-XE-scada-gw-oper module has the following 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

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

	Command or Action	Purpose
		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.
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.

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.

	Command or Action	Purpose
		Note: Every <local-port, remote-ip> must be unique per channel. If remote_subnet is specified, 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.

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
```

DNP3 Enhancement

In some cases, older RTUs were previously used in peer-to-peer mode. These RTUs dynamically swapped the roles of DNP3 Serial subordinate and primary by setting the bit DIR=1 in the message header. ASE's SCADA stack used in Cisco routers are always configured to be DNP3 Serial primary. In this case, all the packets received from DNP3 serial with DIR=1 were ignored causing many messages from RTU to be discarded. To handle these scenarios, a new SCADA configuration CLI has been added:

scada-gw protocol ignore direction.

Enabling this CLI will allow the router to accept incoming packets from RTU even when DIR=1. The new CLI will also be added to the Cisco-IOS-XE-scada-gw.yang config model.

The following is an example usage:

```
Router# config term
Router(config)# scada-gw protocol ignore direction
```

Configuration Example

Configuration example with scada-gw protocol ignore direction on T101/T104

```
scada-gw protocol t101
channel rt-chan
link-addr-size two
bind-to-interface Async0/2/0
session rt-sess
attach-to-channel rt-chan
common-addr-size one
cot-size two
info-obj-addr-size three
link-addr 31
sector rt-sec
attach-to-session rt-sess
asdu-addr 100
scada-gw protocol t104
channel mt-chan
t3-timeout 20
tcp-connection 0 local-port 8001 remote-ip 192.168.1.0/24
session mt-sess
attach-to-channel mt-chan
sector mt-sec
attach-to-session mt-sess
asdu-addr 101
map-to-sector rt-sec
scada-gw protocol ignore direction
scada-gw enable
```

Starting and Stopping the Protocol Translation Engine

You must start the Protocol Translation Engine to use Protocol Translation on the IR1101.

Starting–After enabling SCADA encapsulation on the IR1101 serial port and configuring the T101 and T104 protocols on the IR1101, you can start the Protocol Translation Engine.

Stopping–Before you can make any configuration changes to Protocol Translation on the IR1101 with an active Protocol Translation Engine, you must stop the engine.

Before you begin

Before **starting** the Protocol Translation Engine on the router for the **first time**, make sure you complete the following items:

Enabling the IR1101 Serial Port and SCADA Encapsulation, on page 257

Configuring T101 and T104 Protocol Stacks, on page 258

Procedure

	Command or Action	Purpose
Step 1	**configure terminal**	Enters global configuration mode.
Step 2	**[no] scada-gw enable**	Starts (**scada-gw enable**) or stops (**no scada-gw enable**) the Protocol Translation Engine on the IR1101.

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
```

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
```

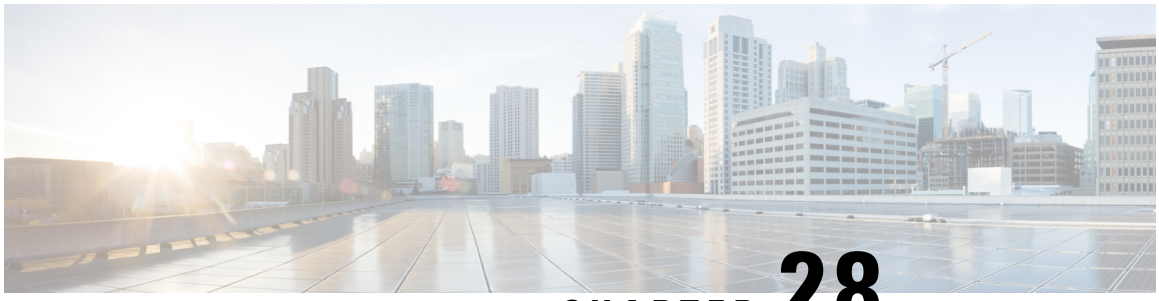
```
DNP3 network channel [test]: 4 max simultaneous connections
conn: local-ip: 3.3.3.21        local-port 20000        remote-ip 3.3.3.15      data-socket
 1
Total:
  1 current client connections
  0 total closed connections
router# show scada statistics
DNP3 network Channel [test]:
  5 messages sent, 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
```

Debug Commands

This section lists some debug commands that are helpful when troubleshooting.

Table 19: SCADA Function Level Debug Commands

Command	Purpose
debug scada function config	Configuration trace
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 28

Raw Socket Transport

This chapter contains the following sections:

- Information About Raw Socket Transport, on page 275
- Prerequisites, on page 278
- Guidelines and Limitations, on page 278
- Raw socket feature enhancement, on page 278
- Default Settings, on page 278
- Configuring Raw Socket Transport, on page 279
- Rawsocket Keepalive Configuration CLI, on page 284
- Yang Data Model Support for Raw Socket Transport, on page 285
- Verifying Configuration, on page 286
- Configuration Examples, on page 286

Information About Raw Socket Transport

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 IR1101 and provides a reference section describing the Raw Socket Transport commands.

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.

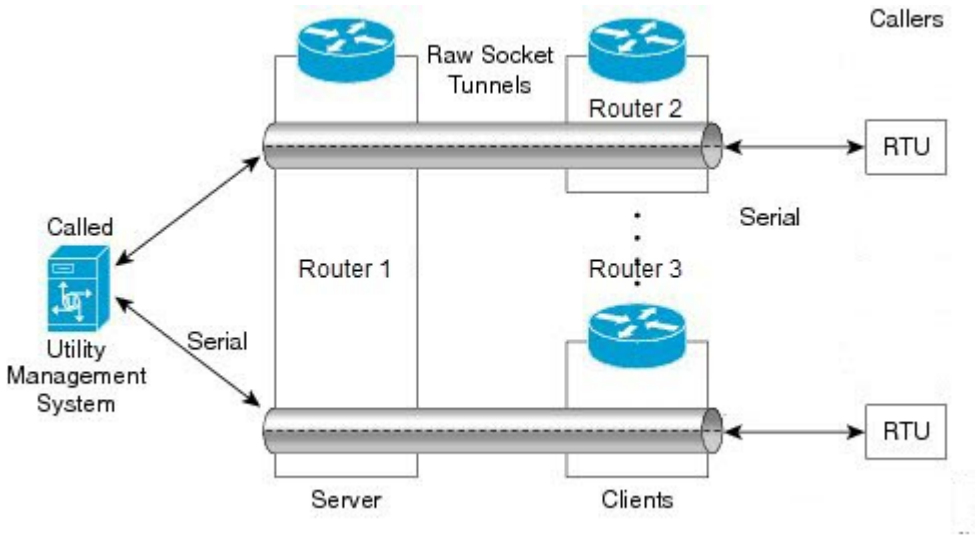
This section includes the following topics:

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 IR1101 can initiate up to 32 TCP sessions to Raw Socket servers, which can be other IR1101 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 IR1101 routers. One IR1101 router (Router 1) acts as a Raw Socket server, listening for TCP connection requests from the other IR1101 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 IR1101 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 IR1101 to transmit the serial data to the peer. Once the IR1101 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 IR1101 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 IR1101 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 IR1101 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 IR1101 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 288 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.

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.

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 IR1101.
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 IR1101 router, you must first enable an asynchronous serial port and enable Raw Socket TCP or UDP encapsulation for that port.

Before you begin

Determine availability of the serial port on the IR1101.

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.

EXAMPLE: Enable Serial Port

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 279.

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 IR1101 to transmit the data to the peer. When the IR1101 accumulates this much data in its buffer, it packetizes the data and forwards it to the Raw Socket peer. *length*— 2 to 1400 bytes. By default, the packet-length trigger is disabled.
Step 4	**raw-socket packet-timer** *timeout*	Specifies the maximum time in milliseconds the IR1101 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 IR1101 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.

EXAMPLE: Common Raw Socket Line Options

```
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 279.

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 server** *port* [*ip_address*]	Starts the Raw Socket Transport TCP server for an asynchronous line interface. In Raw Socket server mode, the IR1101 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.

EXAMPLE: Raw Socket TCP Server

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 279.

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	**raw-socket 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.

EXAMPLE: Raw Socket TCP Client

This example shows how to configure a Raw Socket TCP client for an asynchronous serial line. The IR1101 (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)#
```

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 279.

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. *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.

EXAMPLE: Raw Socket UDP Connection

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.

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

This module contains a collection of YANG definitions for Raw Socket Transport Configuration commands.

This 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>
```

- Cisco-IOS-XE-rawsocket-oper.yang

This module contains a collection of YANG definitions for Raw Socket Transport operational data.

This 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

Verifying Configuration

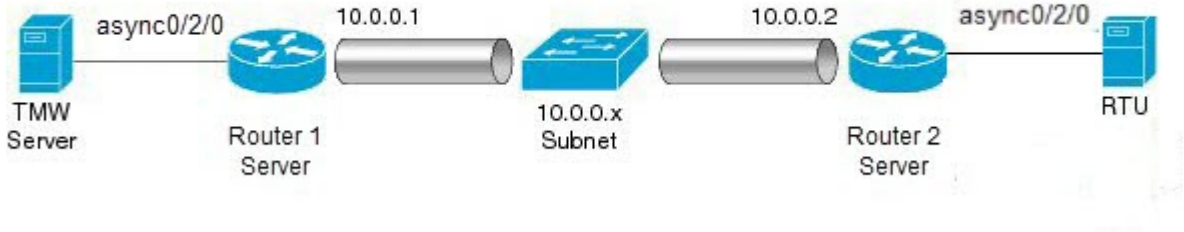
Command	Purpose
show running-config	Shows the configuration of the IR1101, including those features that are active and their settings.
show raw-socket tcp detail	Displays information about Raw Socket Transport TCP activity.
show raw-socket tcp sessions	Displays information about Raw Socket Transport 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.

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 IR1101 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 IR1101s highlighted in the above figure:

IR1101 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

This example shows the configuration for a Raw Socket UDP connection between two IR1101 routers:

From Router1

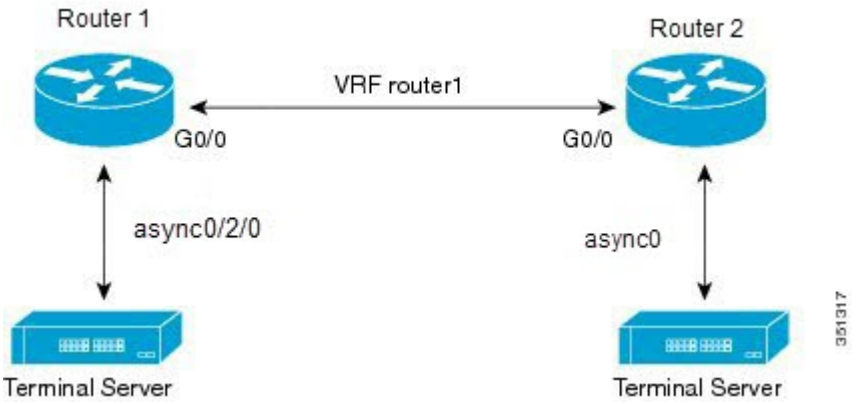
```
interface GigabitEthernet0/1
ip address 192.168.0.8 255.255.255.0
duplex auto
speed auto
interface async0/2/0
no ip address
encapsulation raw-udp
line 0/2/0
raw-socket udp connection 192.168.0.2 2 2
```

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 IR1101, 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:

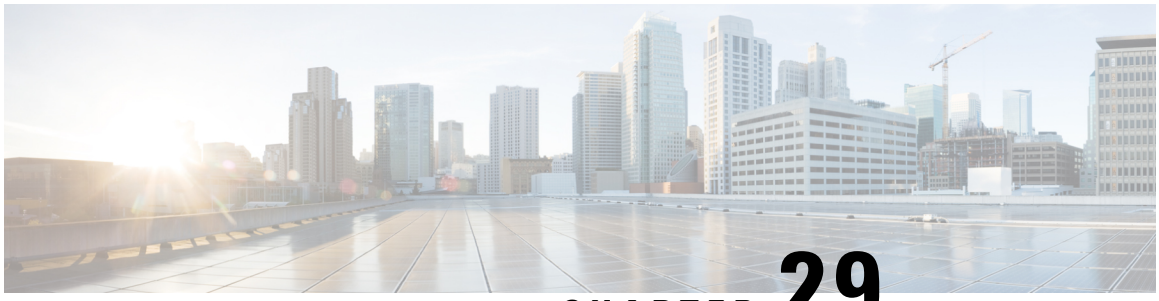
```
interface GigabitEthernet0/0
 vrf forwarding router1
 ip address 100.100.100.1 255.255.255.0
 duplex auto
 speed auto
```

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 29

IRM-1101 Expansion Module

This section contains the following topics:

- IRM-1100 Expansion Module Overview, on page 291
- mSATA Overview, on page 293
- Digital IO, on page 297
- New Cellular Pluggable Modules, on page 300
- IRM-1100 Expansion Module on the Compute Side, on page 301
- SFP Support, on page 301

IRM-1100 Expansion Module Overview

The IR1101 Router has an Expansion Module that adds key capabilities such as dual LTE Pluggables, mSATA SSD FRU, SFP, and Digital GPIO connections.

The Expansion Module comes in two types:

- IRM-1100-SPMI
- IRM-1100-SP

Warning It is important to note that just like the Base IR1101, Online Insertion and Removal (OIR) is not supported on The Expansion Module. If the 4G module (or mSATA) is inserted or pulled out while the device is powered up, it may damage the module.

The following figure shows the front panel of the IRM-1100-SPMI and highlights some of its capabilities:

Figure 43: IRM-1100-SPMI Expansion Module Details



Item	Description
1	4 GPIO + 1 Return (Digital I/O) **Note** Functionality is available on Cisco IOS-XE release 16.12.1 and above.
2	SFP Connector
3	Pluggable Module
4	mSATA SSD Slot
5	Digital I/O LEDs

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

Hardware Interface	Naming Convention
Gigabit Ethernet SFP port on Expansion Module	gigabitethernet 0/0/5
Cellular Interface on Expansion Module	cellular 0/3/0 and 0/3/1
GPIO on Expansion Module	alarm contact 1-4

mSATA Overview

IOx/Guest-OS legacy systems on which end users can host applications, typically came with a disk storage of 4GB to store user data. Functionality has been added allowing for a Cisco supported Pluggable mSATA SSD PID to add 50 GB of available storage. Support for a 100 GB mSATA SSD has the following limitations:

- There is no support for the **show inventory** command.
- Supports 55GB (IOx allocation for applications and packages alike), 32B (IOS allocation for storage can be viewed using the **dir msata** command).



Warning Online Insertion and Removal (OIR) is not supported. If the mSATA SSD is inserted or pulled out while the device is powered up, it may damage the module.



Note As with any IoT platform, for IOx, use the Fog Director, Local Manager, or app-hosting CLI's to install applications and access the new mSATA disk storage provided.

50 GB mSATA Partitioning

IOS-XE divides the mSATA SSD into 2 partitions. One for IOS-XE and the other for IOx. The percentage of usage is:

- IOS: 33.33 %
- IOx: 66.66 %

Using these percentages, the space allocation breaks down as follows:

50GB mSATA:

- IOS: 16.51 GB
- IOx: 31.43 GB

Using the mSATA SSD

Functionality-wise, there are no configuration and troubleshooting differences to the end-user in IOS or IOx, with or without mSATA. The system simply recognizes the additional storage. There are some CLI commands that will show information that pertains to the mSATA storage. Examples are show inventory, and show platform msata.

```
Router#show inventory
++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
INFO: Please use "show license UDI" to get serial number for licensing.
++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
```

```
Router#show platform hardware msata lifetime
SSD Lifetime Remaining: 99% -> 99% of the net disk read/write lifetime is remaining

Router#show platform hardware msata status
SSD is present

Router#show platform hardware msata
SSD Lifetime remaining(%): 99
```

Display the mSATA Partitioning:

Display mSATA partition 1 in IOS-XE:

```
Router#dir msata:
Directory of msata:/
11 drwx 16384 Jun 4 2019 17:59:45 +00:00 lost+found
33820622848 bytes total (32052379648 bytes free)
```

Copy contents to and from mSATA partition:

```
Router#copy bootflash: msata:
Source filename []? ir1101-uefi-rommon.SSA
Destination filename [ir1101-uefi-rommon.SSA]?
Copy in progress...CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
2097152 bytes copied in 0.164 secs (12787512 bytes/sec)
```

Display disk space allocated by mSATA to IOx:

```
Router#show app-hosting resource
CPU:
Quota: 1000(Units)
Available: 1000(Units)
Memory:
Quota: 862(MB)
Available: 862(MB)
Storage space:
Total: 58313(MB)
Available: 58313(MB)
```

Displaying the Wear Leveling Data for the mSATA SSD

IOx Local Manager/ Fog Director can now display the wear leveling data for the mSATA SSD on the IR1101.

In the IOx Local Manager, it is observed by selecting **System > Storage**.

From the IOS command line, you can monitor the lifetime using the **show platform hardware msata** command.

```
Router#show platform hardware msata lifetime
SSD Lifetime remaining(%): 98
```

After a router reload, it will take a few minutes (approximately 5) before this data will be populated again.

When the SSD lifetime reduces to 15% and 5% of the lifetime limit, errors start getting reported in syslog.

For example:

```
*Jan 30 19:03:00.257: %IOX-4-IOX_SSD_LIFETIME_WARN: SSD Lifetime remaining in module:15
*Jan 30 19:02:30.157: %IOX-2-IOX_SSD_LIFETIME_CRITICAL: SSD Lifetime remaining in module:5
```

MIB support for mSATA Wear Ratio and Usage

mSATA functionality was added to the router to add extra storage for IOx apps. The following table shows the router with the OID:

Table 20: mSATA OIDs

SKU	OID
IR1100-SSD-100G	1.3.6.1.4.1.9.12.3.1.9.96.176

As part of this enhancement, SNMP support has been added for the following mSATA parameters on the router:

- lifetime remaining (wear leveling)
- memory usage for the mSATA SSD

The **show platform hardware msata** command gives information about this MIB.

Related documentation:

https://www.cisco.com/c/en/us/support/cloud-systems-management/iox/tsd-products-support-series-home.html

https://developer.cisco.com/docs/iox/

Example: Actual OID and output of SNMP get/walk on OID

<OID> = STRING: "Lifetime Remaining: 99%, Usage: 30%"

Feature Details

The following conditions must be met before performing SNMP requests on the Router:

- An active mSATA module must be configured in the router.
- The Integrator must have incorporated the supported pluggable mSATA into their design.
- Verify this using the **show platform hardware msata** CLI.

Feature Assumptions

- After a router reload it will take approximately 5 minutes before mSATA data will be populated again. Only SNMP get is allowed on the OID and is marked as read-only. Setting its value will not be allowed.
- Configurations to enable SNMP on the router are necessary for fetching MIB value.

Yang model for mSATA

YANG is a popular data modeling language to represent data sent over network management protocols such as NETCONF and RESTCONF. The Cisco-IOS-XE-device-hardware-oper YANG model has been modified to show mSATA information. mSATA has two CLIs to display associated data.

These two CLIs are:

show platform hardware msata status

- The CLI gives information on whether the SSD is present or not.
- If the SSD is present, a message "SSD is present" is displayed.
- If the SSD is not present, a message "SSD is not present" is displayed.

show platform hardware msata lifetime

- If SSD is present an output representing the SSD lifetime in % is displayed: "SSD lifetime remaining (%): 99"
- If SSD is not present, a message "SSD is not present" is displayed.

A typical YANG response for mSATA in device-inventory is as shown below:

```
 <device-inventory>
          <hw-type>hw-type-ssd</hw-type>
          <hw-dev-index>5</hw-dev-index>
          <version>V00</version>
          <part-number>IR-SSD-MSATA-100G</part-number>
          <serial-number>FOC21520XFV</serial-number>
          <hw-description>mSATA Module</hw-description>
          <dev-name>Expansion module 2 - mSATA Module</dev-name>
          <field-replaceable>true</field-replaceable>
          <hw-class>hw-class-virtual</hw-class>
          <lifetime>99</lifetime>
</device-inventory>
```

Cisco IOS-XE Yang Data Models are found here:

https://github.com/YangModels/yang/tree/master/vendor/cisco/xe

Each release has a directory, and the 17.5.1 release is found under 1751.

Support for mSATA and IO Support for IRM-1100-SPMI in CM Side

With previous software releases, the mSATA and Digital I/O on the IRM-1100-SPMI were only supported on the Expansion Module side of the IR1101. With 17.8.1, support is available on the Compute Module (CM) side with the following restrictions:

IRM-1100-SPMI installed on both sides:

- This combination is not supported.
- Only the mSATA and Digital I/O from the EM side will work.
- The Digital I/O from the CM side will NOT work.

IRM-1100-SPMI installed on the CM side:

- The mSATA and Digital I/O will work.
- The Digital I/O instances will be numbered 1-4.

Digital IO

The IR1101 has two different Expansion Modules, the IRM-1100-SP and IRM-1100-SPMI. The IRM-1100-SPMI comes with a Digital I/O connector which has 4 GPIO connections plus 1 Return connection. Both Dry and Wet contacts up to 60Volts.

- Dry contact is isolated from a voltage source (or “No Volt”), with an embedded relay function (NPN transistor), usually used to indicate an event. For example: open/close, alarm.
- Wet contact is a contact with external power (+3.3V to +60V, max 150mA of current allowed at high voltage) applied, usually used to energize something. For example: solenoid, light.

Digital IO is similar to the ALARM IN and ALARM OUT supported on the IR800 series routers. The differences are that on the IR800 series, ALARM IN is a dedicated input, the ALARM OUT is a dedicated output. With Digital IO, it can be input or output. ALARM OUT includes a relay to provide the Normally Open (NO) or Normally Close (NC) terminals. Digital IO does not include a relay.

There are no traps for alarms on the GPIO.

More information on the Digital IO hardware capabilities can be found in the Cisco Catalyst IR1101 Rugged Series Router Hardware Installation Guide .

Configuration Commands

You can set the alarm severity to critical, major, minor, or none. The severity is included in the alarm message when the alarm is triggered.

To configure and show alarms on the IR1101, use the Command Line Interface (CLI).

Command	Purpose
configure terminal	Enters global configuration mode.
alarm contact *contact-number* **enable**	Enables the alarm contact number. o The contact-number value is from 0 to 4. <0-4> Alarm contact number (0: Alarm port, 1-4: Digital I/O). Alarm contact 0 is located in the base unit (pins 3 and 4) and always in Output Mode. Additional configurations for Alarm 0 include *severity* , *threshold* and *trigger* . Alarm contact 1-4 (pins 1-4) are located in the IRM-1100 Expansion Module and can be in Input or Output Mode. Pin 5 is for ground. Additional configurations for Alarms 1-4 include *application* , *output* , *severity* , *threshold* and *trigger* .

Command	Purpose
alarm contact {*contact-number* {**application** {dry | wet} | **description** | **enable** | {**output** {1 for High | 0 for Low} | **severity** {critical | major | minor | none} | **threshold** {1600-2700} | **trigger** {closed | open}}	• Enter a *contact number* (0-4) that you are configuring. • The **description** string is up to 80 alphanumeric characters in length and is included in any generated system messages. • For **application**, select dry (default) or wet. Only applicable for Digital I/O ports 1-4. • **enable** is for enabling the alarm port. A no alarm contact contact-number x will disable the alarm port. • The **output** is either 1 for High or 0 for Low. Only application for Digital I/O ports 1-4. • For **severity**, enter *critical* , *major* , *minor* or *none* . If you do not configure a severity, the default is minor. • For **threshold**, select a value between 1600-2700. The default value is 1600 mv. • For **trigger**, enter *open* or *closed* . If you do not configure a trigger, the alarm is triggered when the circuit is closed.
end	Returns to privileged EXEC mode.
show alarm	Shows the configured alarm contacts.
copy running-config startup-config	(Optional) Saves your entries in the configuration file.

Verify alarm contacts using the CLI:

```
Router(config)#alarm contact ?
<0-4> Alarm contact number (0: Alarm port, 1-4: Digital I/O)
```

Configuration Examples

Configure an alarm.

```
ir1101#conf term
Enter configuration commands, one per line.   End with CNTL/Z.
ir1101(config)#alarm contact 1 description

Your Descriptive Text Here
ir1101(config)#alarm contact 1 severity critical

ir1101(config)#alarm contact 1 trigger closed

ir1101#
```

To show the alarm status:

```
ir1101#show alarm
```

```
Alarm contact 0:
Enabled: Yes
Status: Not Asserted
Application: Dry
Description: test
Severity: Critical
Trigger: Open
Threshold: 2000
```

Example of an alarm being generated:

```
ir1101# !
*Nov 27 14:54:52.573: %IR1101_ALARM_CONTACT-0-EXTERNAL_ALARM_CONTACT_ASSERT: External alarm
 asserted, Severity: Critical
```

To show the alarm status during an event:

```
ir1101#show alarm
ALARM CONTACT
Enabled: Yes
Status: Asserted
Application: Dry
Description: test
Severity: Critical
Trigger: Open
Threshold: 2000
Digital I/O 1:
Enabled: No
Status: Not Asserted
Application: Dry
Description: External digital I/O port 1
Severity: Minor
Trigger: Closed
Threshold: 1600
Digital I/O 2:
Enabled: No
Status: Not Asserted
Application: Dry
Description: External digital I/O port 2
Severity: Minor
Trigger: Closed
Threshold: 1600
Digital I/O 3:
Enabled: No
Status: Not Asserted
Application: Dry
Description: External digital I/O port 3
Severity: Minor
Trigger: Closed
Threshold: 1600
Digital I/O 4:
Enabled: Yes
Status: Not Asserted
Description: External digital I/O port 4
Mode: Output
Router#
```

Example of an alarm being cleared:

```
ir1101# !
```

```
*Nov 27 14:55:02.573: %IR1101_ALARM_CONTACT-0-EXTERNAL_ALARM_CONTACT_CLEAR: External alarm
 cleared
ir1101#
```

New Cellular Pluggable Modules

Release 16.12.1 supports new pluggable modules/modems. The IR1101 with an Expansion Module supports DUAL LTE (Active/Active), DUAL SIM and DUAL Radio.

- Dual LTE (active/active or active/backup) is supported on the IR1101 equipped with an expansion module and two LTE pluggable interfaces. One on the base unit, the other on the expansion module.
- With DUAL SIM, the two SIMs operate in active/backup mode on the single LTE pluggable module. With DUAL Radio the two LTE pluggable modules operate in active/active mode with each of the two SIMs assigned to a specific cellular radio on the DUAL Radio.

See the following table for details on the new SKUs.

SKU ID	Modem Used	Description	Technology Supported
P-LTE-VZ	WP7601-G	U.S. (Verizon) Single Micro SIM	LTE CAT4: B4, B13
P-LTE-US	WP7603-G	North America (AT&T) Dual Micro SIM	LTE CAT4:B2,B4,B5,B12HSPA+,UMTS: B2,B4,B5
P-LTE-GB	WP7607-G	Europe Dual Micro SIM	LTE CAT4: B3, B5, B8, B20, B28 HSPA+: B1, B5, B8 EDGE: 900/1800
P-LTEA-LA	EM7430	APAC	**LTE Bands**: B1, B3, B5, B7, B8, B18, B19, B21, B28, B38, B39, B40, B41. **Non-LTE Bands**: B87 - WCDMA (Europe, Japan, and China) 2100 band B91 - WCDMA US 850 band B92 - WCDMA Japan 800 band B114 - WCDMA Europe and Japan 900 band B115 - WCDMA Japan 1700 band B125 - WCDMA Japan 850 band

SKU ID	Modem Used	Description	Technology Supported
P-LTEA-EA	EM7455	USA, Canada, Europe, Latin America	**LTE bands**: Bands B2, B4, B5, B13 **Non-LTE bands**: B87 - WCDMA (Europe, Japan, and China) 2100 band B88 - WCDMA US PCS 1900 band B89 - WCDMA (Europe and China) DCS 1800 band B90 - WCDMA US 1700 band B91 - WCDMA US 850 band B114 - WCDMA Europe and Japan 900 band

IRM-1100 Expansion Module on the Compute Side

The IR1101 has two attachment points for expansion modules. The top side of the router is referred to as the Expansion side. The bottom side of the router is referred to as the Compute side.

Prior to IOS XE release 17.7.1, support was only on the Expansion side.

Starting with the 17.7.1 release, additional modules can be connected on the Compute side.

Features and Limitations

The following apply to the IRM-1100 with release 17.7.1:

- Switchports will not work if anything is connected on the Compute side
- MSATA and GPIO Pins for IRM-1100-SPMI are not supported when it is connected to the Compute side (bottom) for 17.7.1
- The IR1101 can only support a maximum of two LTE interfaces. Connecting an IRM-1100 on both sides is not supported. If connected in this configuration, only the Expansion side will be active.
- LTE interfaces when connected on Compute side, are enumerated cellular 0/4/0 and cellular 0/4/1
- CAT18 LTE module is not supported on the Compute side.
- Only the LTE interface functions when the IRM-1100-SP or IRM-1100-SPMI is connected on the Compute side.

SFP Support

The SFP interface on the Expansion Module operates differently than on the Base unit. The SFP interface on the IR1101 base module is part of the combo port (SFP/RJ45) for GigabitEthernet0/0/0. It may be configured as Layer-3 (default) or Layer-2 interface.

The SFP interface on the Expansion Module is only an SFP interface. It is named GigabitEthernet0/0/5, and is a Layer-2 interface. For Layer-3 feature set, it must be assigned to a VLAN interface.

Note For IRM-1100-4S8IO and IRM-1100-SPMI/IRM-1100-SP, copper SFPs support only 1G speed and full duplex operation. 10/100M speed and half duplex operating modes are not supported.

Details about the SFP Interface can be displayed using the **show interfaces transceiver detail** CLI, for example:

```
Router#show interfaces transceiver detail
IDPROM for transceiver Gigabitethernet0/0/0:
  Description                               = SFP or SFP+ optics (type 3)
  Transceiver Type:                         = GE T (26)
  Product Identifier (PID)                  = ABCU-5710RZ-CS4
  Vendor Revision                           =
  Serial Number (SN)                        = AGM151124J4
  Vendor Name                               = CISCO-AVAGO
  Vendor OUI (IEEE company ID)              = 00.17.6A (5994)
  CLEI code                                 =
  Cisco part number                         =
  Device State                              = Enabled.
  Date code (yy/mm/dd)                      = 11/03/21
  Connector type                            = Unknown.
  Encoding                                  = 8B10B (1)
  Nominal bitrate                           = GE (1300 Mbits/s)
  Minimum bit rate as % of nominal bit rate = not specified
  Maximum bit rate as % of nominal bit rate = not specified

Socket Verification

  SFP IDPROM Page 0xA0:
        000:        03 04 00 08 00 00 00 00 00 00
        010:        00 01 0D 00 00 00 00 00 64 00
        020:        43 49 53 43 4F 2D 41 56 41 47
        030:        4F 20 20 20 20 20 01 00 17 6A
        040:        41 42 43 55 2D 35 37 31 30 52
        050:        5A 2D 43 53 34 20 20 20 20 20
        060:        41 0C C1 15 00 10 00 00 41 47
        070:        4D 31 35 31 31 32 34 4A 34 20
        080:        20 20 20 20 31 31 30 33 32 31
        090:        20 20 00 00 00 99 00 00 06 17
        100:        C5 44 22 B7 DE 02 63 0F 59 73
        110:        64 EC A5 37 19 00 00 00 00 00
        120:        00 00 00 00 0F 2C 6D 22 FF FF
        130:        FF FF FF FF FF FF FF FF FF FF
        140:        FF FF FF FF FF FF FF FF FF FF
        150:        FF FF FF FF FF FF FF FF FF FF
        160:        FF FF FF FF FF FF FF FF FF FF
        170:        FF FF FF FF FF FF FF FF FF FF
        180:        FF FF FF FF FF FF FF FF FF FF
        190:        FF FF FF FF FF FF FF FF FF FF
        200:        FF FF FF FF FF FF FF FF FF FF
        210:        FF FF FF FF FF FF FF FF FF FF
        220:        FF FF FF FF

  SFP IDPROM Page 0xA2:
        000:        00 00 00 00 00 00 00 00 00 00
        010:        00 00 00 00 00 00 00 00 00 00
        020:        00 00 00 00 00 00 00 00 00 00
        030:        00 00 00 00 00 00 00 00 00 00
        040:        00 00 00 00 00 00 00 00 00 00
        050:        00 00 00 00 00 00 00 00 00 00
        060:        00 00 00 00 00 00 00 00 00 00
```

```
        070:        00 00 00 00 00 00 00 00 00 00
        080:        00 00 00 00 00 00 00 00 00 00
        090:        00 00 00 00 00 00 00 00 00 00
        100:        00 00 00 00 00 00 00 00 00 00
        110:        00 00 00 00 00 00 00 00 00 00
        120:        00 00 00 00 00 00 00 00 00 00
        130:        00 00 00 00 00 00 00 00 00 00
        140:        00 00 00 00 00 00 00 00 00 00
        150:        00 00 00 00 00 00 00 00 00 00
        160:        00 00 00 00 00 00 00 00 00 00
        170:        00 00 00 00 00 00 00 00 00 00
        180:        00 00 00 00 00 00 00 00 00 00
        190:        00 00 00 00 00 00 00 00 00 00
        200:        00 00 00 00 00 00 00 00 00 00
        210:        00 00 00 00 00 00 00 00 00 00
        220:        00 00 00 00 00 00 00 00 00 00
        230:        00 00 00 00 00 00 00 00 00 00
        240:        00 00 00 00 00 00 00 00 00 00
        250:        00 00 00 00 00 00
  Link reach for 9u fiber (km)                    = SX(550/270m) (0)
                                                    1xFC-MM(500/300m) (0)
                                                    2xFC-MM(300/150m) (0)
                                                    ESCON-MM(2km) (0)
  Link reach for 9u fiber (m)                     = SX(550/270m) (0)
                                                    1xFC-MM(500/300m) (0)
                                                    2xFC-MM(300/150m) (0)
                                                    ESCON-MM(2km) (0)
  Link reach for 50u fiber (m)                    = SR(2km) (0)
                                                    IR-1(15km) (0)
                                                    IR-2(40km) (0)
                                                    LR-1(40km) (0)
                                                    LR-2(80km) (0)
                                                    LR-3(80km) (0)
                                                    DX(40KM) (0)
                                                    HX(40km) (0)
                                                    ZX(80km) (0)
                                                    VX(100km) (0)
                                                    1xFC, 2xFC-SM(10km) (0)
                                                    ESCON-SM(20km) (0)
  Link reach for 62.5u fiber (m)                  = SR(2km) (0)
                                                    IR-1(15km) (0)
                                                    IR-2(40km) (0)
                                                    LR-1(40km) (0)
                                                    LR-2(80km) (0)
                                                    LR-3(80km) (0)
                                                    DX(40KM) (0)
                                                    HX(40km) (0)
                                                    ZX(80km) (0)
                                                    VX(100km) (0)
                                                    1xFC, 2xFC-SM(10km) (0)
                                                    ESCON-SM(20km) (0)
  Nominal laser wavelength                  = 16652 nm.
  DWDM wavelength fraction                  = 16652.193 nm.
  Supported options                          = Tx disable
```

Assigning L3 SVI with IP address to Extended Module GE 0/0/5 SFP interface:

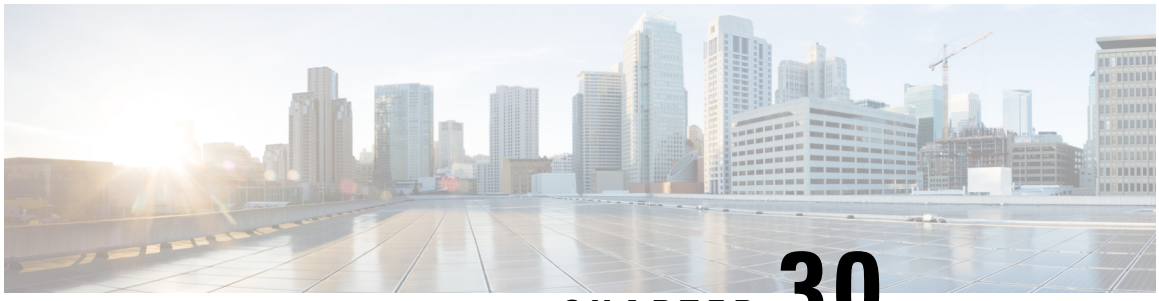
```
IR1101#config t
IR1101(config)#interface g0/0/5
IR1101(config-if)#switchport access vlan 2
IR1101(config-if)#no shut
IR1101(config-if)#interface vlan2
IR1101(config-if)#ip address 192.168.1.2 255.255.255.0
IR1101(config-if)#no shut
```

You can find all of the supported SFP Interfaces in the Cisco Catalyst IR1101 Rugged Series Router Hardware Installation Guide

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 30

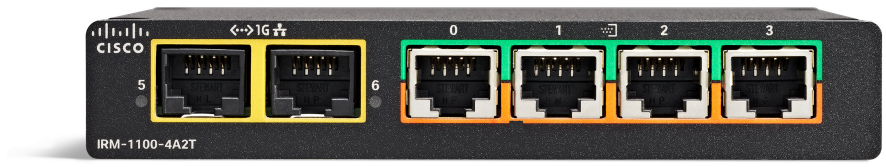
IRM-1100-4A2T Expansion Module

This chapter contains the following sections:

- IRM-1100-4A2T Overview, on page 305
- Guidelines and Limitations, on page 307
- Deployment Scenarios, on page 308
- Inventory Details based on Deployment, on page 311
- Gigabit Ethernet Switch Ports, on page 312
- LEDs, on page 312
- Async Ports, on page 314
- GPIO Configuration Pins, on page 316
- Configuration Examples for Additional Async Interfaces, on page 318
- Scada Protocol Translations, on page 319
- Serial Relay, on page 321
- Using the WebUI to Configure Async Ports, on page 322

IRM-1100-4A2T Overview

The IRM-1100-4A2T is an expansion module that can be attached to the IR1101. It offers an additional four asynchronous serial ports and two Ethernet interfaces to the IR1101. The following graphic shows the IRM-1100-4A2T.



The IRM-1100-4A2T Ethernet interfaces are Layer 2 RJ45 10/100/1000 Mbps ports.

The IRM-1100-4A2T serial ports are RJ45 combo ports (RS232/RS485/RS422).

The IR1101 has two sides that expansion modules mount to. The top is called the Expansion side, and the bottom is called the Compute side. If the additional module is connected to the top, then it is referenced as

the Expansion Module (EM) side. If the additional module is connected on the bottom, then it is referenced as the Compute Module (CM) side. Functionality differs depending on which side the expansion module is attached to, and how many and type of expansion modules are in use.

The IRM-1100-4A2T can be managed from the following tools:

- Cisco DNA Center
- WebUI

Router Switch Path

The switch path that is detected on the platform, is based on the type of additional module connected on the Expansion module (EM) side. Refer to the following table:

Additional Module	Switch Path
No Module Connected	IR1101-ES-5
IRM-1100-SPMI	IR1101-ES-6S
IRM-IR1100-4A2T	IR1101-ES-7G

Note When an IRM-IR1100-4A2T is connected on both sides of the IR1101-K9, there is a maximum of nine Async interfaces which can be enumerated. The switch path for the IR1101-K9 will be IR1101-ES-7G.

Serial Port Pinouts and Characteristics

The serial ports are intended as a DCE port, capable of both RS232 and RS485. RS485 can support full or half duplex.

The RJ45 pinouts are shown in the following figure and table:

Figure 44: Pinouts

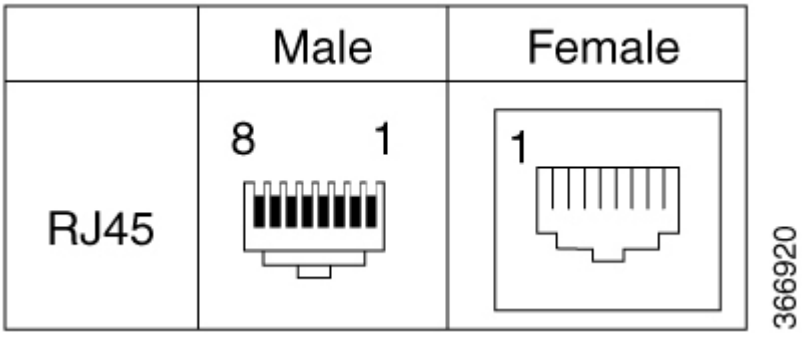


Table 21: Serial Port Characteristics

RS232					RS485 Full Duplex		RS485 Half Duplex	
Pin #	**Signal Description**	**Abbr.**	**S0 (DTE)**	**S1 (DCE)**	**Signal**	**DIR**	**Signal**	**Direction**
1	DCE ready. Used as DSR in Cisco IOS.	DSR/RI	Input	Output	TX-	Output	TX/RX+	<->
2	Received Line Signal Detector	DCD	Input	Output	TX+	Output	TX/RX-	<->
3	DTE Ready	DTR	Output	Input	RX-	Input	—	—
4	Signal Ground	COM	—	—	COM	—	COM	—
5	Received Data	RxD	Input	Output	—	—	—	—
6	Transmitted Data	TxD	Output	Input	RX+	Input	—	—
7	Clear To Send	CTS	Input	Output	—	—	—	—
8	Request To Send	RTS	Output	Input	—	—	—	—

Guidelines and Limitations

The IRM-1100-4A2T has the following guidelines and limitations:

- Available with IOS-XE release 17.7.1
- Supports four deployment scenarios
- No support for OIR
- Ethernet ports are L2 switchport only
- Switchports will not work if anything is connected to the Compute module (bottom) side

Both the IRM-1100-SPMI Expansion Module and the IRM-1100-4A2T Expansion Module have the following guidelines and limitations:

- The CAT18 LTE module is not supported on the Compute module (bottom) side
- MSATA and GPIO pins are not supported when attached to the Compute Module side.
- The IR1101 can only support a maximum of two LTE interfaces. This means connecting an Expansion Module with LTE interfaces on both the EM and CM side is not supported. If connected, only the EM side will be active.

Deployment Scenarios

The IRM-1100-4A2T supports four different deployment scenarios. This section discusses the differences in functionality between the four.

Interface numbering are enumerated based on the deployment of the IRM-1100-4A2T module.

Scenario One

In this scenario, the IRM-1100-4A2T is mounted on the Expansion side, or the top. See the following figure:



In this configuration, you get full functionality out of the Serial and Ethernet ports.

There is support for 4 additional Async interfaces, and 2 Gigabit ethernet interfaces.

Interface numbering in this scenario is as follows:

- async 0/3/0 (corresponding line is: line 0/3/0) [Serial]
- async 0/3/1 (corresponding line is: line 0/3/1) [Serial]
- async 0/3/2 (corresponding line is: line 0/3/2) [Serial]
- async 0/3/3 (corresponding line is: line 0/3/3) [Serial]
- gigabitetherenet 0/0/5 [Layer 2]
- gigabitetherenet 0/0/6 [Layer 2]

Scenario Two

In this scenario, the IRM-1100-4A2T is mounted on the Compute side, or the bottom. In addition, the solution also has the IRM-1100-SPMI expansion module mounted on the Expansion side, or the top. See the following figure:



In this configuration, the ethernet ports on the IRM-1100-4A2T will not function. The serial ports have full functionality.

There is support for 4 Async interfaces and no support for additional layer 2 interfaces.

Interface numbering in this scenario is as follows:

- async 0/4/0 (corresponding line is: line 0/4/0) [Serial]
- async 0/4/1 (corresponding line is: line 0/4/1) [Serial]
- async 0/4/2 (corresponding line is: line 0/4/2) [Serial]
- async 0/4/3 (corresponding line is: line 0/4/3) [Serial]

Scenario Three

In this scenario, the IRM-1100-4A2T is mounted on the Expansion side, or the top. In addition, the configuration also has the IRM-1100-SPMI expansion module mounted on the Compute side, or the bottom. See the following figure:



In this configuration, the IRM-1100-4A2T is mounted on the Expansion side, or top, and has full functionality. The SFP port on the IRM-1100-SPMI mounted on the Compute side, or bottom, will not function.

Interface numbering in this scenario is as follows:

- Async 0/3/0 – 0/3/3 [Connected on EM side]
- Async 0/4/0 – 0/4/3 [Connected on CM side]
- Gi0/0/5 and Gi0/0/6 [Layer 2 interfaces from EM side]
- LTE interface on CM side, cellular 0/4/0 and cellular 0/4/1

Scenario Four

In this scenario, there are two IRM-1100-4A2T expansion modules mounted on both the Expansion side and the Compute side. See the following figure:



In this configuration, the IRM-1100-4A2T mounted on the Expansion side, or top, has full functionality. The Ethernet ports on the IRM-1100-4A2T mounted on the Compute side, or bottom, will not function.

There is support for 8 more Async interfaces, and 2 Gigabit ethernet interfaces.

Interface numbering in this scenario is as follows:

- Async 0/3/0 – 0/3/3 [Connected on EM side]
- Async 0/4/0 – 0/4/3 [Connected on CM side]
- Gi0/0/5 and Gi0/0/6 [Layer 2 interfaces from EM side]

Inventory Details based on Deployment

The output to the **show inventory** command will show different details based upon which side of the IR1101 base unit it is attached to.

```
Router#sh inv
++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
INFO: Please use "show license UDI" to get serial number for licensing.
++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++

NAME: "Chassis", DESCR: "IR1101 Base Chassis"
PID: IR1101-K9          , VID: V03  , SN: FCW2452P561

NAME: "Module 0 - Mother Board", DESCR: "Cisco IR1101 motherboard"
PID: IR1101-K9          , VID: V03  , SN: FOC245126XR

NAME: "module subslot 0/0", DESCR: "IR1101-ES-7G"
```

```
PID: IR1101-ES-7G      , VID: V01  , SN:

NAME: "module subslot 0/4", DESCR: "P-LTE-MNA Module"
PID: P-LTE-MNA         , VID: V01  , SN: FOC24230U79

NAME: "Modem on Cellular0/4/0", DESCR: "Sierra Wireless WP7610"
PID: WP7610            , VID: 10000, SN: 356307100162618

NAME: "Module 2 - Compute Module", DESCR: "IR1100 expansion module with Pluggable slot,
SFP, mSATA SSD slot and Digital GPIO"
PID: IRM-1100-SPMI     , VID: V02  , SN: FCW2502PAP0

NAME: "Module 3 - Expansion Module", DESCR: "IR1100 expansion module with 4 Async ports and
 2 copper ports"
PID: IRM-1100-4A2T     , VID: V00  , SN: FOC25150ZRJ

Router# sh ip int bri
Interface              IP-Address      OK? Method Status                Protocol
GigabitEthernet0/0/0   unassigned      YES NVRAM  administratively down down
FastEthernet0/0/1      unassigned      YES unset  administratively down down
FastEthernet0/0/2      unassigned      YES unset  administratively down down
FastEthernet0/0/3      unassigned      YES unset  administratively down down
FastEthernet0/0/4      unassigned      YES unset  down                  down
GigabitEthernet0/0/5   unassigned      YES unset  administratively down down
GigabitEthernet0/0/6   unassigned      YES unset  down                  down
Cellular0/1/0          unassigned      YES NVRAM  administratively down down
Cellular0/1/1          unassigned      YES NVRAM  administratively down down
Async0/2/0             unassigned      YES unset  up up
Async0/3/0             unassigned      YES unset  up ip
Async0/4/0             unassigned      YES unset  administratively down down
Async0/3/1             unassigned      YES unset  administratively down down
Async0/4/1             unassigned      YES unset  administratively down down
Async0/3/2             unassigned      YES unset  administratively down down
Async0/4/2             unassigned      YES unset  administratively down down
Async0/3/3             unassigned      YES unset  administratively down down
Async0/4/3             unassigned      YES unset  administratively down down
Vlan1                  unassigned      YES unset  up                    down
```

Gigabit Ethernet Switch Ports

The Ethernet ports are Layer 2 RJ45 10/100/1000 Mbps ports.

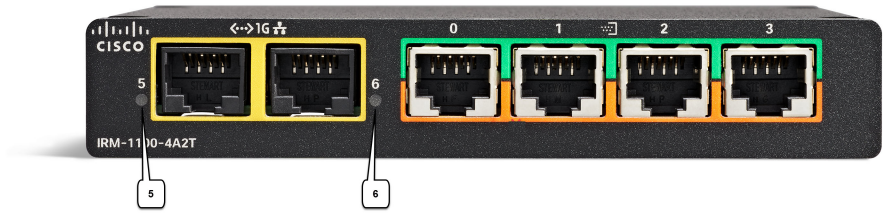
The base router (IR1101) GE port is named gigabitethernet 0/0/0. When the IRM-1100-4A2T is mounted on the Expansion side, or top, two additional ports are available:

- gigabitethernet 0/0/5
- gigabitethernet 0/0/6

LEDs

There are two LEDs on the front associated with the two Ethernet ports (5 and 6). See the following figure:

Figure 45: Ethernet Port LEDs



See the following table for the LED functionality:

Color/State	Description
Green	Port link, no activity
Flashing Green	Link healthy with activity
Off	No link

LED status is also available through the command line:

```
Router# show led

 SYSTEM LED : Green

 Custom LED : Off

 VPN LED : Off

 ALARM LED : Off

 GigabitEthernet0/0/0 LED : On
 FastEthernet0/0/1    LED : On
 FastEthernet0/0/2    LED : Off
 FastEthernet0/0/3    LED : Off
 FastEthernet0/0/4    LED : Off
 GigabitEthernet0/0/5 LED : On
 GigabitEthernet0/0/6 LED : Off

 *Cellular 0/1*
 LTE module Enable LED : Green
 LTE module SIM 0 LED  : Off
 LTE module SIM 1 LED  : Off
 LTE module GPS LED    : Off
 LTE module RSSI 0 LED : Off
 LTE module RSSI 1 LED : Off
 LTE module RSSI 2 LED : Off
 LTE module RSSI 3 LED : Off
```

Async Ports

IOS-XE release 17.7.1 software provides support for an additional module (IRM-1100-4A2T) that has 4 Async ports and 2 gigabit ethernet interfaces. The software enumerates the interface numbers depending on which side of the Base IR1101 the expansion module is attached to.

The base router (IR1101) async port is async 0/2/0, with the out of bound management port being async 0/2/1.

When the IRM-1100-4A2T is mounted on the Expansion side, or top, the async ports are numbered as:

- async 0/3/0 (corresponding line is: line 0/3/0)
- async 0/3/1 (corresponding line is: line 0/3/1)
- async 0/3/2 (corresponding line is: line 0/3/2)
- async 0/3/3 (corresponding line is: line 0/3/3)

When the IRM-1100-4A2T is mounted on the Compute side, or bottom, the async ports are numbered as:

- async 0/4/0 (corresponding line is: line 0/4/0)
- async 0/4/1 (corresponding line is: line 0/4/1)
- async 0/4/2 (corresponding line is: line 0/4/2)
- async 0/4/3 (corresponding line is: line 0/4/3)

The async ports on the IRM-1100-4A2T support:

- media-type RS232 (DCE) and RS485 (RS422 and RS485 share the same configuration)
- full-duplex/half-duplex

Serial RJ45 Pin-Outs

All Serial ports can be in three operational modes:

- RS232
- RS485 Full Duplex
- RS485 Half Duplex

All ports follow the RS232 signal standard, with a max baud rate of 115Kbps supported. The following table shows pinouts for the four ports:

Pin Number	Description	Mode	Direction
1	Data Set Ready	DCE	OUT
2	DCD/Ring	DCE	OUT
3	Data Terminal Ready	DCE	IN

Pin Number	Description	Mode	Direction
4	Signal Ground	—	—
5	Receive Data	DCE	OUT
6	Transmit Data	DCE	IN
7	Clear to Send	DCE	OUT
8	Request to Send	DCE	IN

DCE Interface Configuration Steps

The default interface configuration for all ports on the serial expansion module is RS232. If the interface is configured for media-type RS485, the default configuration is in full-duplex mode.

- The configuration of GI0/0/5 and Gi0/0/6 are similar to the L2 ports in the IR1101 base unit.
- The async ports support both RS232 and RS485/full/half-duplex. Additionally, "media-type", "full-duplex", and "half-duplex" are supported on the expansion module, compared to the async 0/2/0 in the IR1101 base unit.

Default Configuration

The default configuration for all ports of the serial expansion module is RS232.

```
Router#sh run int Async0/3/0
Building configuration...

Current configuration : 92 bytes

interface Async0/3/0
no ip address
encapsulation scada
shutdown
media-type rs232
```

Configuration Example for Media-Type RS232

The CLI **media-type ?** shows rs232 and rs485 available.

```
Router(config)#int Async0/3/3
Router(config-if)#media
Router(config-if)#media-type ?
 rs232 Set RS232 media type
 rs485 Set RS485 media type
```

Configure the media-type for RS232.

```
Router(config-if)#media-type rs232
Router(config-if)#no shut
Router(config-if)#end

Router#sh run int Async0/3/3
Building configuration...
!
```

```
Current configuration : 82 bytes
!
interface Async0/3/3
 no ip address
 encapsulation scada
 media-type rs232
 end
```

Configuration Example for Media-Type RS485

Configure the media-type for RS485.

```
Router#conf t
Enter configuration commands, one per line.
Router(config)#int Asyn0/3/0
Router(config-if)#media
Router(config-if)#media-type rs485
Router(config-if)#end

Router# sh run int Async0/3/0

Building configuration...

Current configuration : 105 bytes
!
interface Async0/3/0
 no ip address
 encapsulation scada
 shutdown
 media-type rs485
 full-duplex
end
```

Configuration Example for Media-Type RS485 (half-duplex)

Configure the media-type for RS485 running half duplex.

```
Router(config)#int Async0/4/2
Router(config-if)#media
Router(config-if)#media-type rs485
Router(config-if)#half-duplex
Router(config-if)#end

Router#sh run int Async0/4/2
Building configuration...

Current configuration : 105 bytes
!
interface Async0/4/2
 no ip address
 encapsulation scada
 shutdown
 media-type rs485
 half-duplex
```

GPIO Configuration Pins

The IRM-1100-4A2T has four Async ports that send signals to hardware using GPIO pins, through which media-type and duplex settings are configured. The following are examples of standard signals if GPIO pin is set to ‘6’ configured as RS232, ‘4’ configured as RS485 full-duplex and ‘C’ configured as RS485 half-duplex.

```
Router#sh controllers Async0/3/0
Line: 0/3/0(74) Interface:Async0/3/0
State=6 encapsulation=95 speed=9600 maxmtu=1500
Duplex=0 ACCM_TX=0xFFFFFFFF ACCM_RX=0xFFFFFFFF
Max_idle=10 frame_size=100
Buffered bytes=0 tty capabilities=0x8 tty statbits=0x40 databits=8
TX packet cnt:0 Scattered: 0 Particle cnt:0 Request cnt:0
PPP in total:0
PPP Rx head:0x0 tail:0x0
GPIO read: 6666
```

Note Based on the above output, all the Async ports 0/3/0 to 0/3/3 are configured with default media-type RS232.

```
Router#sh controllers Async0/4/2
Line: 0/4/2(100) Interface:Async0/4/2
State=6 encapsulation=95 speed=9600 maxmtu=1500
Duplex=0 ACCM_TX=0xFFFFFFFF ACCM_RX=0xFFFFFFFF
Max_idle=10 frame_size=100
Buffered bytes=0 tty capabilities=0x8 tty statbits=0x40 databits=8
TX packet cnt:0 Scattered: 0 Particle cnt:0 Request cnt:0
PPP in total:0
PPP Rx head:0x0 tail:0x0
GPIO read: 6C66
```

Note Based on the above output, Async port 0/4/2 is configured with RS485 Half-duplex, and remaining ports Async0/4/0,0/4/1 and 0/4/3 are configured with default media-type RS232.

```
Router# sh controllers Async0/3/3
Line: 0/3/3(77) Interface:Async0/3/3
State=4 encapsulation=97 speed=9600 maxmtu=1500
Duplex=0 ACCM_TX=0xFFFFFFFF ACCM_RX=0xFFFFFFFF
Max_idle=10 frame_size=100
Buffered bytes=0 tty capabilities=0x8 tty statbits=0x440 databits=8
TX packet cnt:0 Scattered: 0 Particle cnt:0 Request cnt:0
PPP in total:0
PPP Rx head:0x0 tail:0x0
GPIO read: 4666
```

Note Based on the above output, Async port 0/3/3 is configured with RS485 Full-duplex, and the remaining ports Asyn0/3/0, Async0/3/1 and Async0/3/2 are configured with default media-type RS232.

Debug Commands

There is a debug command available for troubleshooting the GPIO configuration:

```
Router# debug condition interface <ASYNC_INTERFACE_SLOT> event
```

Note This command is not supported for the Async 0/2/0 interface.

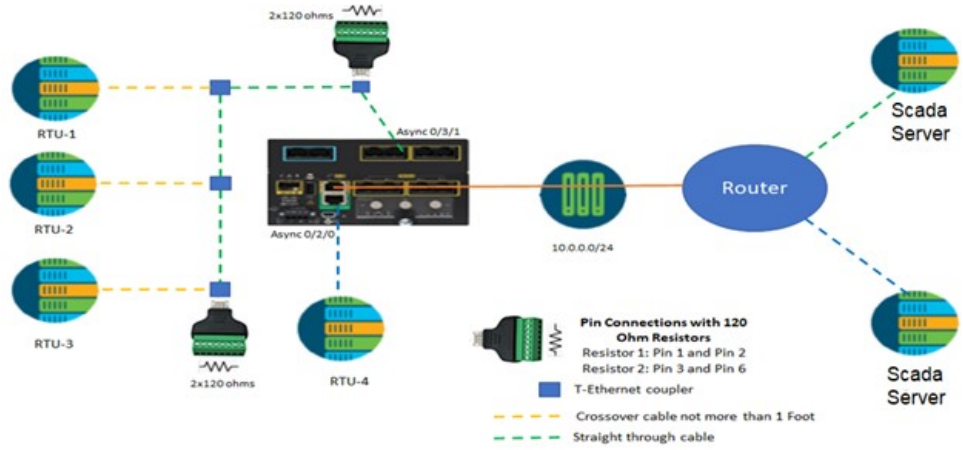
Configuration Examples for Additional Async Interfaces

For more information see the chapter Raw Socket Transport in IR1101 Rugged Series Router Software Configuration Guide

Raw-TCP Multi-hop (daisy chain)

For raw-tcp, the user needs to configure encapsulation raw-tcp under the Async interface, and needs to configure associated line interface either as server or client. Maximum number of sessions per server is 32.

Figure 46: Raw-TCP Multi-hop Example



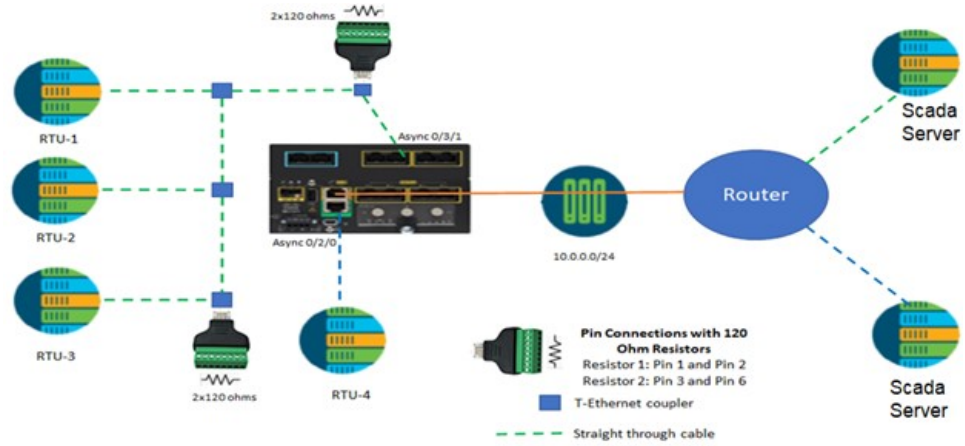
The following is an example configuration for two routers as shown above.

IR1101	Other Router **Note** **Can be any IOS-XE router or IOS router, that supports at least 2 serial interfaces.**

`int Async0/2/0`
`  encapsulation raw-tcp`
`  no shut`
`int Async 0/3/1`
`  encapsulation raw-tcp`
`  media-type rs485`
`  full-duplex`
`  no shut`
 	
`line 0/2/0`
`  raw-socket tcp client 10.0.0.2 6000 10.0.0.1 6001`
`line 0/3/1`
`  raw-socket tcp client 10.0.0.2 5000 10.0.0.1 5001`
	`int Async 0/2/0`
	`  encapsulation raw-tcp`
	`  no shut`
	`int Async 0/2/1`
	`  encapsulation raw-tcp`
	`  no shut`
	
	`line 0/2/0`
	`  raw-socket tcp server 6000`
	`line 0/2/1`
	`  raw-socket tcp server 5000`

Raw-UDP Multi-hop (daisy chain)

Figure 47: Raw-UDP Multi-hop Example



The following is an example configuration for two routers as shown above.

IR1101	Other Router **Note** **Can be any IOS-XE router or IOS router, that supports at least 2 serial interfaces.**

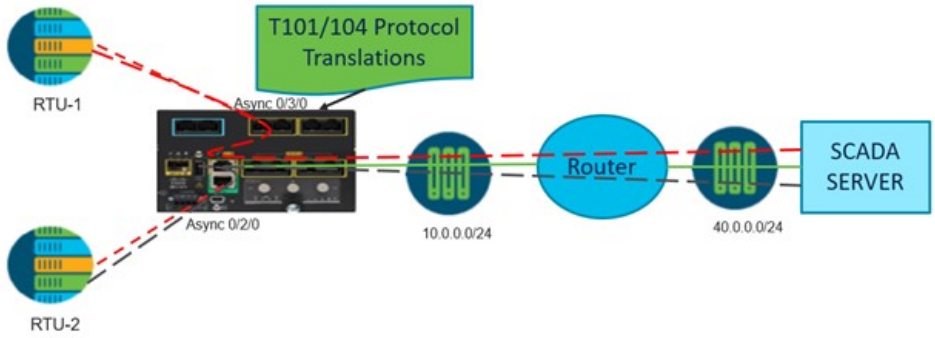
`int Async0/2/0`
`  encapsulation raw-udp`
`  no shut`
`int Async 0/3/1`
`  encapsulation raw-udp`
`  media-type rs485`
`  half-duplex`
`  no shut`
 	
`line 0/2/0`
`  raw-socket udp connection 10.0.0.2 6001 6000`
`  10.0.0.1`
`line 0/3/1`
`  raw-socket udp connection 10.0.0.2 5001 5000`
`  10.0.0.1`
	`int Async 0/2/0`
	`  encapsulation raw-udp`
	`  no shut`
	`int Async 0/2/1`
	`  encapsulation raw-udp`
	`  no shut`
	
	`line 0/2/0`
	`  raw-socket udp connection 10.0.0.1 6000 6001`
	`  10.0.0.2`
	`line 0/2/1`
	`  raw-socket udp connection 10.0.0.1 5000 5001`
	`  10.0.0.2`

Scada Protocol Translations

Further information can be found in the Information About SCADA chapter of the IR1101 Rugged Series Router Software Configuration Guide

T101/T104

Figure 48: T101/T104 Configuration Example



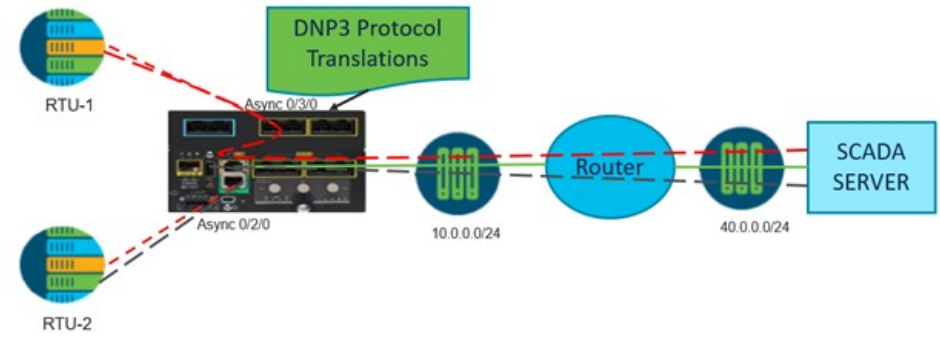
The following is an example configuration for the figure shown above.

Table 22: T101/T104 Configuration Example

int Async0/2/0 encapsulation scada no shut	scada-gw protocol t101 channel rt-chan-1 link-mode balanced bind-to-interface Async0/2/0 session rt-sess-1 attach-to-channel rt-chan-1 common-addr-size one cot-size two info-obj-addr-size three link-addr 31 sector rt-sec-1 attach-to-session rt-sess-1 asdu-addr 100	Scada-gw protocol t104 channel mt-chan-1 t3-timeout 20 tcp-connection 0 local-port 5000 remote-ip any session mt-sess-1 attach-to-channel mt-chan-1 sector mt-sec-1 attach-to-session mt-sess-1 asdu-addr 120 map-to-sector rt-sec-1
int Async0/3/0 encapsulation scada media-type rs485 half-duplex no shut	channel rt-chan-2 link-mode balanced bind-to-interface Async0/3/0 session rt-sess-2 attach-to-channel rt-chan-2 common-addr-size one cot-size two info-obj-addr-size three link-addr 32 sector rt-sec-2 attach-to-session rt-sess-2 asdu-addr 101	channel mt-chan-2 t3-timeout 20 tcp-connection 0 local-port 6000 remote-ip any session mt-sess-2 attach-to-channel mt-chan-2 sector mt-sec-2 attach-to-session mt-sess-2 asdu-addr 121 map-to-sector rt-sec-2 scada-gw enable

DNP3 IP/Serial

Figure 49: DNP3 IP/Serial Configuration Example



The following is an example configuration for the figure shown above.

Table 23: DNP3 IP/Serial Configuration Example

int Async0/2/0 encapsulation scada no shut	scada-gw protocol dnp3-serial channel dnp3_serial_channel_1 link-addr source 5 request-timeout 60 link-timeout 6 unsolicited-response enable bind-to-interface Async0/2/0 no protocol test-link session dnp3_serial_session_1 attach-to-channel dnp3_serial_channel_1 link-addr dest 1	scada-gw protocol dnp3-ip channel dnp3_ip_channel_1 link-addr dest 3 send-unsolicited-msg enable tcp-connection local-port 5000 remote-ip any session dnp3_ip_session_1 attach-to-channel dnp3_ip_channel_1 link-addr source 7 map-to-session dnp3_serial_session_1
int Async0/3/0 encapsulation scada media-type rs485 half-duplex no shut	channel dnp3_serial_channel_2 link-addr source 6 request-timeout 60 link-timeout 6 unsolicited-response enable bind-to-interface Async0/3/0 no protocol test-link session dnp3_serial_session_2 attach-to-channel dnp3_serial_channel_2 link-addr dest 2	channel dnp3_ip_channel_2 link-addr dest 5 send-unsolicited-msg enable tcp-connection local-port 6000 remote-ip any session dnp3_ip_session_2 attach-to-channel dnp3_ip_channel_2 link-addr source 8 map-to-session dnp3_serial_session_2 scada-gw protocol ignore direction scada-gw enable

Serial Relay

Serial relay can be supported on all of the Async ports of IRM-1100-4A2T. You can map in any order. Mapping of Async interfaces with “encapsulation relay-line” configured on interface. For Example:

- relay line 0/0/0 0/2/0
- relay line 0/0/1 0/3/2

- relay line 0/0/2 0/3/0
- relay line 0/0/3 0/3/1
- relay line 0/0/4 0/4/0

Refer to the Serial Relay Service chapter in the IR1101 Configuration Guide for additional detail.

Using the WebUI to Configure Async Ports

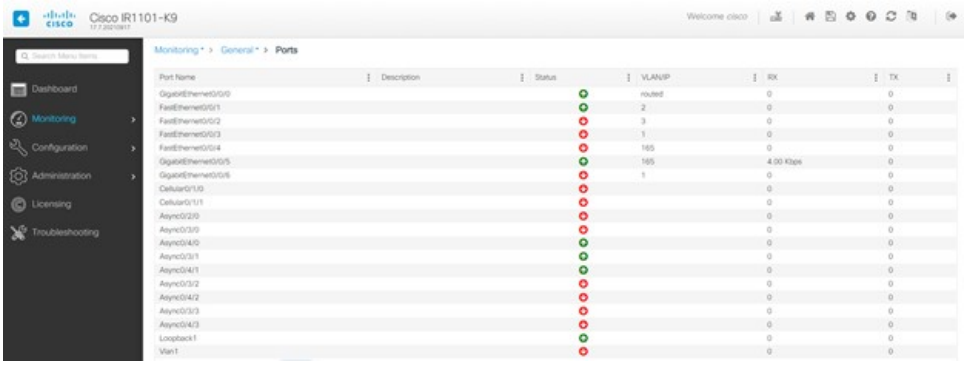
Use the following steps to configure Async ports through the WebUI.

Before you begin

Cisco IOS XE release supports WebUI support (Day-1) as basic template for configuration and validation for Async interfaces.

Ports can be monitored by navigating to **Monitoring > General > Ports**:

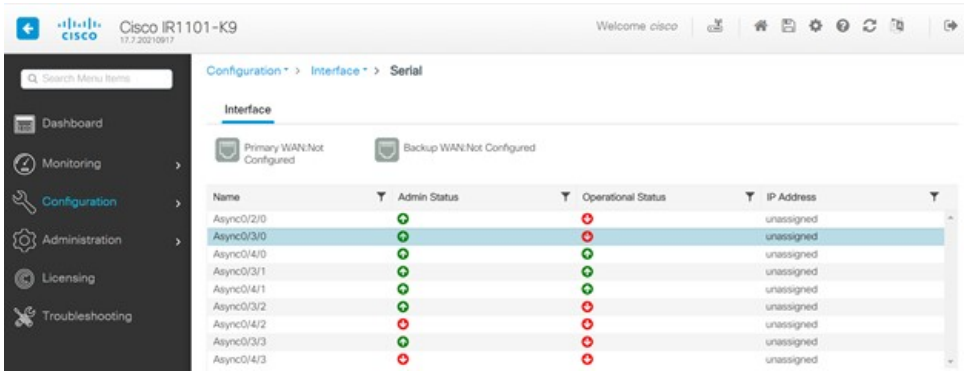
Figure 50: Monitor Ports



Procedure

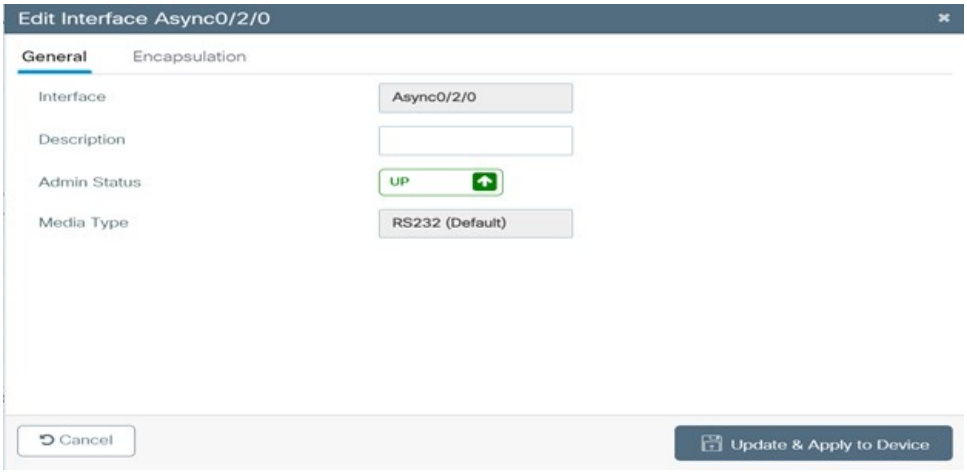
Step 1 Navigate to **Configuration > Interface > Serial**.

Figure 51: Serial Ports



Step 2 Double click on the interface you want to edit. The **Edit Interface** <*Interface Number*> window appears.

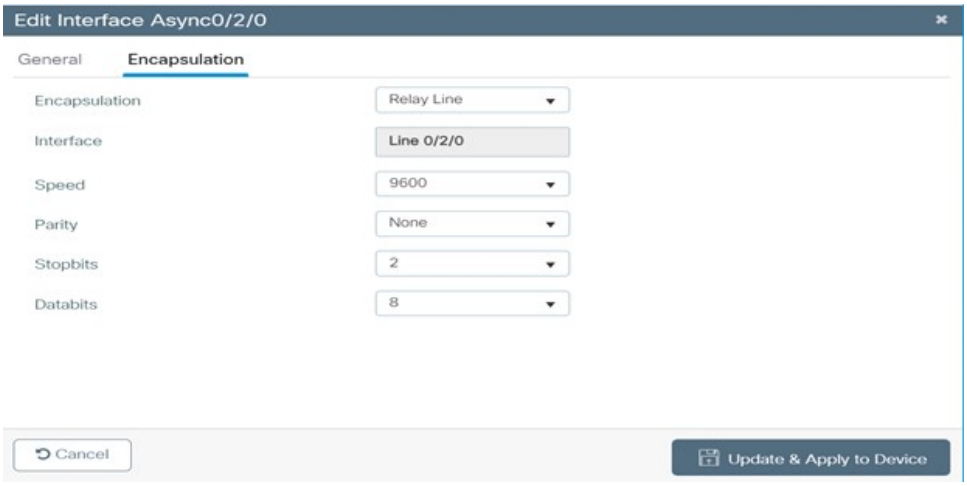
Figure 52: Edit Interface



The Async0/2/0 interface on the base IR1101 supports media-type RS232 by default. You cannot change any media-type associated with this interface.

Step 3 Click on the Encapsulation tab of the Edit Interface window.

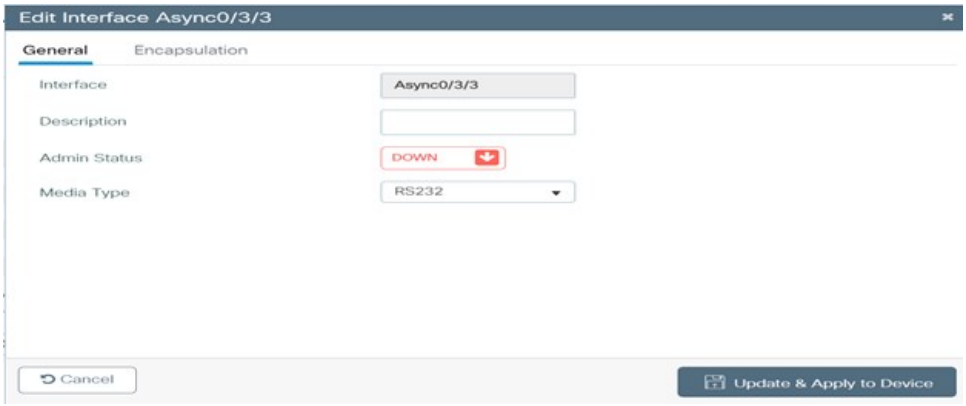
Figure 53: Edit Interface (Encapsulation)



If needed, you can change the encapsulation for the Async0/2/0 interface, and the associated line interface. Select any value from the drop down list that is supported for the Async interface on the IR1101.

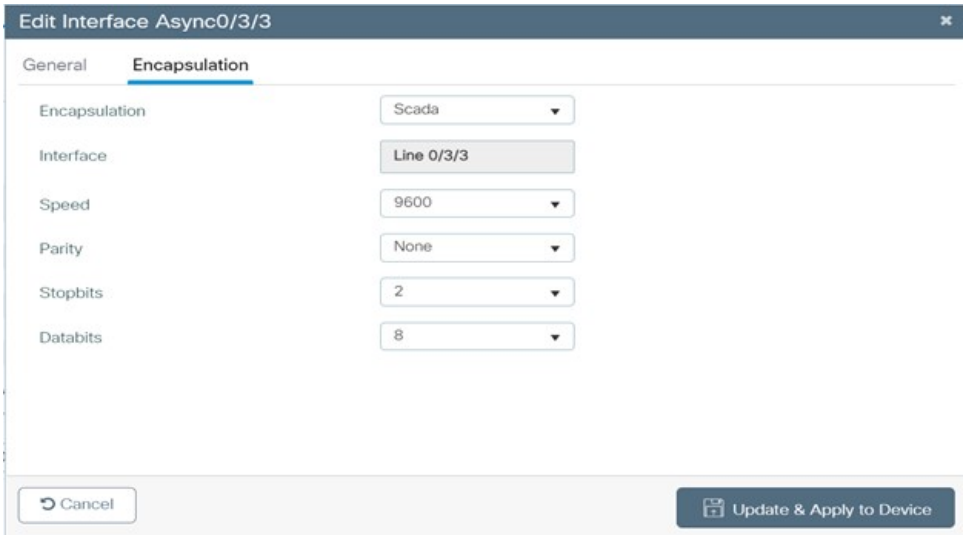
Step 4 Perform the same steps to navigate to the Edit Interface window to configure the Async ports on the IRM-1100-4A2T. For example, edit the Async0/3/3 interface:

Figure 54: Edit Interface Async0/3/3

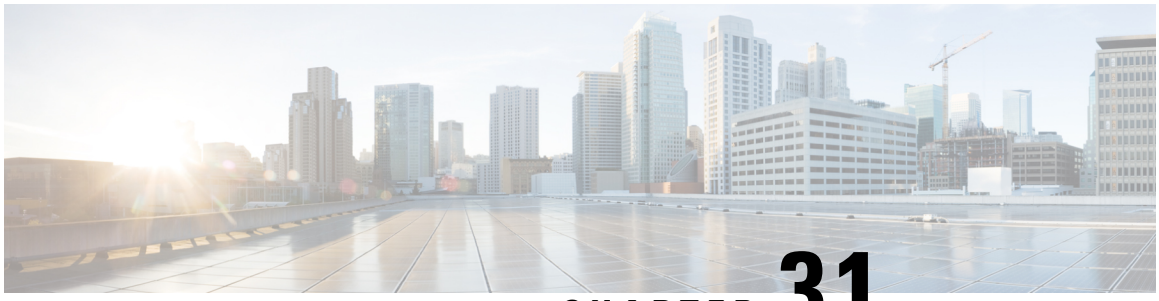


The ports on the IRM-1100-4A2T can have the media type changed from the drop down box. If RS485 is selected, you can select either full or half duplex.

Figure 55: Edit Interface Async0/3/3 (Encapsulation Tab)



Step 5 When satisfied with your selections, click on **Update & Apply to Device**.



CHAPTER 31

IRM-1100-4S8I expansion module

- Expansion module for IR1101 routers, on page 325
- Guidelines and limitations, on page 326
- Support for jumbo frames, on page 326
- Digital IOs, on page 327
- Configure and manage alarms on the router, on page 328
- Configuration and verification examples, on page 330
- Support for L2/L3 SFP interfaces, on page 331

Expansion module for IR1101 routers

The IRM-1100-4S8I expansion module, introduced in Cisco IOS XE Release 17.18.1a, is a hardware add-on with these capabilities:

- adds four SFP ports and 8 Digital IO connections,
- expands the networking and interface capabilities of the IR1101 router,
- supports advanced configurations like jumbo frames and alarm management.

The table displays supported hardware interfaces and naming conventions.

Table 24: Hardware interfaces

Hardware interface	Naming convention
Gigabit Ethernet SFP port on Expansion Module	gigabitethernet 0/0/5 gigabitethernet 0/0/6 gigabitethernet 0/0/7 gigabitethernet 0/0/8
GPIO on Expansion Module	alarm contact 1 to 8

Table 25: Feature History Table

Feature name	Release information	Feature description
IRM-1100-4S8I Expansion Module	Release 17.18.1a	This release introduces the IRM-1100-4S8I Expansion Module for IR1101 routers, adding four SFP ports and eight Digital IO connections for enhanced networking capabilities.

Guidelines and limitations

Follow these guidelines and understand the limitations for using SFPs on the IRM-1100-4S8I Expansion Module:

- Supports both 1G and 100M speeds for fiber SFPs.
- Supports only 1G speed and full-duplex operation for copper SFPs. They do not support 10/100M speed or half-duplex modes.
- Compatibility is restricted to the Expansion Module (EM) and is not available on the Compute Module (CM).
- Online Insertion and Removal (OIR) is not supported (similar to the other expansion modules on Base IR1101).

Support for jumbo frames

Jumbo frames are Ethernet frames that improve network efficiency by reducing the overhead caused by headers.

Advantages of jumbo frame support

On the IRM-1100-4S8I Expansion Module, jumbo frames allow each port to handle larger packets with a Maximum Transmission Unit (MTU) size of up to 9216 bytes, compared to the default Ethernet MTU of 1500 bytes.

- Improved efficiency: Fewer frames are required to transmit the same amount of data, reducing the processing workload on the device.
- Optimized bandwidth usage: A smaller percentage of each frame is dedicated to headers, leaving more room for payload data.

For instance, when sending packets larger than 1500 bytes on GigabitEthernet interfaces, enabling jumbo frames ensures seamless transmission without fragmenting the data into smaller packets.

Configure the MTU size on a router interface

This task enables you to configure the Maximum Transmission Unit (MTU) size on a router interface.

Procedure

Step 1 Enter the global configuration mode.

Example:

```
Router#configure terminal
```

Step 2 Access the interface configuration mode and specify the interface where you want to configure the MTU size.

Example:

```
Router(config)#interface gigabitEthernet0/0/6
```

Step 3 Set the MTU size.

You can choose the MTU size from **1500** to **9216**.

Example:

```
Router(config-if)#mtu 9216
```

Step 4 Exit global configuration mode.

Example:

```
Router#end
```

Step 5 (Optional) Save the configuration.

Example:

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

Step 6 Verify the configuration. Use the **show running-config interface** command to confirm the MTU size.

Example:

```
Router#show running-config interface gigabitEthernet0/0/6
```

Output:

```
Building configuration...
Current configuration : 96 bytes
!
interface GigabitEthernet0/0/6
no switchport
mtu 9216
no ip address
negotiation auto
end
```

Digital IOs

Digital IOs function as electrical connectors that provide various capabilities such as:

- Enable General Purpose Input/Output (GPIO) functionality,
- Support both dry and wet contact configurations, and

- Provide flexible input or output capabilities.

Technical summary of Digital IO features

Digital IOs are designed to facilitate event signaling or power delivery through configurable connections.

- The IRM-1100-4S8I includes a Digital IO connector with:
  - 8 GPIO connections, and
  - 1 Return connection
- Dry contact configurations:
  - Isolated from a voltage source (**No Volt**).
  - Embedded relay function (NPN transistor).
  - Typically used to indicate events, for example, open or close states, alarms.
- Wet contact configurations:
  - External power applied (+3.3V to +60V, max 150mA at high voltage).
  - Typically used to energize devices, for example, solenoids and lights.
- Comparison with IR800 Series:
  - The IR800 series routers support **ALARM IN** (dedicated input) and **ALARM OUT** (dedicated output) with an external power applied (+3.3V to +60V, max 150mA at high voltage).
  - Digital IOs on the IR1101 differ as they allow configurable input or output modes.
  - **ALARM OUT** on the IR800 series includes a relay for **Normally Open** (NO) or **Normally Closed** (NC) terminals; Digital IOs do not include a relay.
- Limitations:

  Digital IOs do not generate traps for alarms concerning GPIO.

For more details on Digital IO hardware capabilities, see Cisco Catalyst IR1101 Rugged Series Router Hardware Installation Guide.

Configure and manage alarms on the router

Set up and customize alarm configurations on the IR1101 router using the CLI.

Before you begin

Familiarize yourself with the IR1101 alarm ports and their locations:

- Alarm contact 0 is located in the base unit and operates in Output Mode.
- Alarm contacts 1 to 8 are in the IRM-1100 4S8I Expansion Module and can operate in Input or Output mode.

Procedure

Step 1 Enter the global configuration mode.

Example:

```
Router#configure terminal
```

Step 2 Enable an alarm contact by specifying its contact number (1 to 8) on the IRM-1100 4S8I Expansion Module.

Example:

```
Router#alarm contact 1 enable
```

Step 3 Configure additional parameters such as application mode, description, output state, severity, threshold, and trigger condition.

See Configure additional parameters on the router.

Step 4 Exit the global configuration mode.

Example:

```
Router#end
```

Step 5 (Optional) Save the configuration to the startup configuration file.

Example:

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

Configure additional parameters on the router

The purpose of this task is to configure and customize alarm contact parameters on the router to ensure proper monitoring and alerting functionality. This includes setting the application mode, providing descriptive information, defining output states, assigning severity levels, establishing threshold values, and specifying trigger conditions for alarm contacts 1-8.

Procedure

Step 1 Set the application mode for alarmcontacts 1-8 with **alarm contact contact-number application {dry | wet}** command.

- **dry**: Default application mode.
- **wet**: Alternate application mode.

Example:

```
Router#alarm contact 1 application dry
```

Step 2 Add a description of up to 80 alphanumeric characters with **alarm contact contact-number description <description>** command.

Example:

```
Router#alarm contact 1 description test
```

Step 3 Specify the output state for alarm contacts 1-8 with **alarm contact contact-number output {1 | 0}** command.

- **1**: High state
- **0**: Low state

Example:

```
Router#alarm contact 1 output 1
```

Step 4 Define the severity level with **alarm contact contact-number severity {critical | major | minor | none}** command.

Example:

```
Router#alarm contact 1 severity minor
```

Default severity: **minor**

Step 5 Set the threshold value with **alarm contact contact-number threshold <value>** command.

Range: **1600-2700** mV (default: **1600** mV)

Example:

```
Router#alarm contact 1 threshold 1600
```

Step 6 Configure the trigger condition with **alarm contact contact-number trigger {closed | open}** command.

Example:

```
Router#alarm contact 1 trigger closed
```

Configuration and verification examples

Alarm contacts in the IR1101 router are configurable monitoring elements that generate system messages when triggered. They allow you to monitor the status and respond to events.

Configuration example

```
Router#conf terminal
Enter configuration commands, one per line.   End with CNTL/Z.
Router(config)#alarm contact 1 description

Your Descriptive Text Here
Router#(config)#alarm contact 1 severity critical

Router(config)#alarm contact 1 trigger closed
```

Verification example

```
Router#show alarm
Alarm contact 1:
Enabled: Yes
Status: Not Asserted
Application: Dry
Description: test
Severity: Critical
```

```
Trigger: Open
Threshold: 2000
```

Example of an alarm being generated:

```
Router#show alarm
Digital I/O 1:
Enabled: Yes
Status: Asserted
Application: Dry
Description: External digital I/O port 1
Severity: Critical
Trigger: Closed
Threshold: 1600
```

Example of an alarm being asserted:

```
Router#
*Jun 17 08:38:26.709: %IIOT_ALARM_CONTACT-5-EXTERNAL_ALARM_CONTACT_ASSERT_MINOR: External
alarm/digital IO port (External digital I/O port 1) asserted
```

Example of an alarm being cleared:

```
Router#
*Jun 17 08:38:22.979: %IIOT_ALARM_CONTACT-5-EXTERNAL_ALARM_CONTACT_CLEAR_MINOR: External
alarm/digital IO port (External digital I/O port 1) cleared
```

Support for L2/L3 SFP interfaces

SFP interfaces support different types of transceivers, allowing users to choose the appropriate module for their specific requirements, such as speed, distance, or media type.

Table 26: SFP features in base and expansion modules

Feature	Base Module (IR1101)	Expansion Module
Interface name	GigabitEthernet0/0/0 (combo port)	GigabitEthernet0/0/5 GigabitEthernet0/0/6 GigabitEthernet0/0/7 GigabitEthernet0/0/8
Default configuration	Layer-3	Layer-2
Layer-3 functionality setup	Default	Use the **no switchport** command to convert these ports to layer 3 mode.

The L2/L3 features supported on IRM-1100-4S8I interfaces are on parity with the GigabitEthernet0/0/0.

For more details on the supported ethernet SFP modules, see Cisco Small Form-Factor Pluggable (SFPs) modules in Cisco Catalyst IR1101 Rugged Series Router Data Sheet.

Examples for SFP modules insertions

Use **show interfaces transceiver detail** command to display details about SFP interfaces.

```
Router#show interfaces transceiver detail
```

```
IDPROM for transceiver Gigabitethernet0/0/0:
  IDPROM for transceiver Gigabitethernet0/0/5:
  Description                               = SFP or SFP+ optics (type 3)
  Transceiver Type:                         = GE T (26)
  Product Identifier (PID)                  = GLC-T-RGD
  Vendor Revision                           = 1.0
  Serial Number (SN)                        = ACW26400ZX0
  Vendor Name                               = CISCO-ACCELINK
  Vendor OUI (IEEE company ID)              = 00.1C.AD (7341)
  CLEI code                                 = COUIBG2CAA
  Cisco part number                         = 3010016701
  Vendor part number                        = RTXL185-200-C87
  Device State                              = Enabled.
  Date code (yy/mm/dd)                      = 22/10/07
  Connector type                            = Unknown.
  Encoding                                  = 8B10B (1)
  Nominal bitrate                           = GE (1300 Mbits/s)
  Minimum bit rate as % of nominal bit rate = not specified
  Maximum bit rate as % of nominal bit rate = not specified

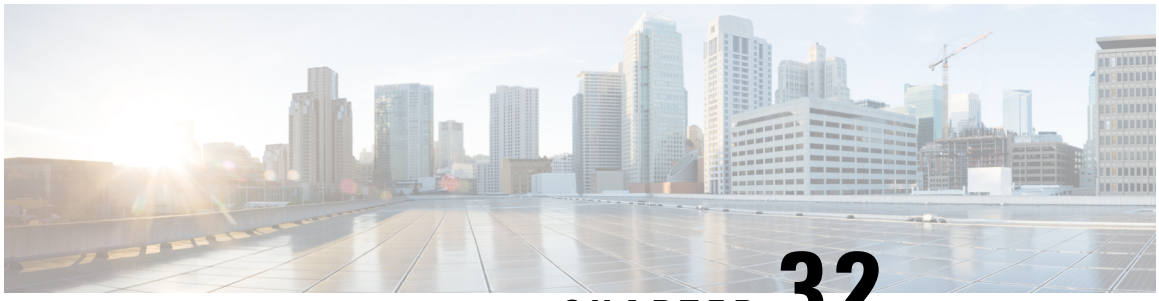
Socket Verification


  SFP IDPROM Page 0xA0:
        000:        03 04 00 08 00 00 00 00 00 00
        010:        00 01 0D 00 00 00 00 00 64 00
        020:        43 49 53 43 4F 2D 41 43 43 45
        030:        4C 49 4E 4B 20 20 01 00 1C AD
        040:        52 54 58 4C 31 38 35 2D 32 30
        050:        30 2D 43 38 37 20 31 2E 30 20
        060:        41 0C C2 C7 00 10 00 00 41 43
        070:        57 32 36 34 30 30 5A 58 30 20
        080:        20 20 20 20 32 32 31 30 30 37
        090:        20 20 00 00 00 D5 00 00 1F 0E
        100:        8A 32 FB 13 F9 88 DA E0 71 05
        110:        DB 26 4D F3 C7 00 00 00 00 00
        120:        00 00 00 00 EC D6 6D 8E 43 4F
        130:        55 49 42 47 32 43 41 41 33 30
        140:        31 30 30 31 36 37 30 31 56 30
        150:        31 20 CA D8 55 00 00 00 00 71
        160:        00 00 00 00 00 00 00 00 00 00
        170:        00 00 00 00 00 00 00 00 00 00
        180:        00 00 00 00 00 00 00 00 00 00
        190:        00 00 47 4C 43 2D 54 2D 52 47
        200:        44 20 20 20 20 20 20 20 20 20
        210:        20 20 00 00 00 00 00 00 00 00
        220:        00 00 00 C1

  SFP IDPROM Page 0xA2:
        000:        00 00 00 00 00 00 00 00 00 00
        010:        00 00 00 00 00 00 00 00 00 00
        020:        00 00 00 00 00 00 00 00 00 00
        030:        00 00 00 00 00 00 00 00 00 00
        040:        00 00 00 00 00 00 00 00 00 00
        050:        00 00 00 00 00 00 00 00 00 00
        060:        00 00 00 00 00 00 00 00 00 00
        070:        00 00 00 00 00 00 00 00 00 00
        080:        00 00 00 00 00 00 00 00 00 00
        090:        00 00 00 00 00 00 00 00 00 00
        100:        00 00 00 00 00 00 00 00 00 00
        110:        00 00 00 00 00 00 00 00 00 00
        120:        00 00 00 00 00 00 00 00 00 00
        130:        00 00 00 00 00 00 00 00 00 00
        140:        00 00 00 00 00 00 00 00 00 00
```

```
      150:        00 00 00 00 00 00 00 00 00 00
      160:        00 00 00 00 00 00 00 00 00 00
      170:        00 00 00 00 00 00 00 00 00 00
      180:        00 00 00 00 00 00 00 00 00 00
      190:        00 00 00 00 00 00 00 00 00 00
      200:        00 00 00 00 00 00 00 00 00 00
      210:        00 00 00 00 00 00 00 00 00 00
      220:        00 00 00 00 00 00 00 00 00 00
      230:        00 00 00 00 00 00 00 00 00 00
      240:        00 00 00 00 00 00 00 00 00 00
      250:        00 00 00 00 00 00
Link reach for 9u fiber (km)                    = SX(550/270m) (0)
                                                  1xFC-MM(500/300m) (0)
                                                  2xFC-MM(300/150m) (0)
                                                  ESCON-MM(2km) (0)
Link reach for 9u fiber (m)                     = SX(550/270m) (0)
                                                  1xFC-MM(500/300m) (0)
                                                  2xFC-MM(300/150m) (0)
                                                  ESCON-MM(2km) (0)
Link reach for 50u fiber (m)                    = SR(2km) (0)
                                                  IR-1(15km) (0)
                                                  IR-2(40km) (0)
                                                  LR-1(40km) (0)
                                                  LR-2(80km) (0)
                                                  LR-3(80km) (0)
                                                  DX(40KM) (0)
                                                  HX(40km) (0)
                                                  ZX(80km) (0)
                                                  VX(100km) (0)
                                                  1xFC, 2xFC-SM(10km) (0)
                                                  ESCON-SM(20km) (0)
Link reach for 62.5u fiber (m)                  = SR(2km) (0)
                                                  IR-1(15km) (0)
                                                  IR-2(40km) (0)
                                                  LR-1(40km) (0)
                                                  LR-2(80km) (0)
                                                  LR-3(80km) (0)
                                                  DX(40KM) (0)
                                                  HX(40km) (0)
                                                  ZX(80km) (0)
                                                  VX(100km) (0)
                                                  1xFC, 2xFC-SM(10km) (0)
                                                  ESCON-SM(20km) (0)
Nominal laser wavelength                    = 16652 nm.
DWDM wavelength fraction                    = 16652.194 nm.
Supported options                                = Tx disable
Supported enhanced options                       = none
Diagnostic monitoring                            = none
```

For more details on the supported SFP Interfaces, see Cisco Catalyst IR1101 Rugged Series Router Hardware Installation Guide.



CHAPTER 32

IOx Application Hosting

This chapter contains the following sections:

- Information About Application Hosting, on page 335
- Application Hosting on the IR1101 Router, on page 338
- How to Configure Application Hosting, on page 342
- Installing and Uninstalling Apps, on page 348
- Overriding the App Resource Configuration, on page 349
- Verifying the Application Hosting Configuration, on page 350
- Digital IO Enhancement, on page 352
- Configuration Examples for Application Hosting, on page 352
- IOx Access to USB Storage, on page 353
- Cisco ThousandEyes Enterprise Agent, on page 354

Information About 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.

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 IR1101 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 IR1101.

Cisco Application Hosting Overview

The IR1101 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.

GPS access to IOx Apps

Previously, when a modem has GPS enabled, the NMEA stream was not forwarded to IOx. This release allows the NMEA stream to be forwarded to IOx from the ngiolite module. There are two steps to enable this.

- Create a tunnel between Linux and IOx
- Forward all NMEA messages over the tunnel to IOx.

The system code checks for the presence of the tunnel, and if it is not present, data cannot be sent to IOx.

To support this feature there will be two new tunnels created for two cellular modems on the IR1101 and IR1800. Two tunnels are created by default and whichever modem has the GPS/NMEA enabled, the NMEA stream will be sent over the corresponding tunnel as follows:

Modem0:

[Linux] /dev/ttyTun5 and /dev/ttyTun6 [IOx]. Soft link to /dev/ttyTun5 will be created named /dev/ttyTunNMEA0, soft link to /dev/ttyTun6 will be created named /dev/ttyNMEA0 which can be accessed from IOx.

Modem1:

[Linux] /dev/ttyTun7 and /dev/ttyTun8 [IOx]. Soft link to /dev/ttyTun7 will be created named /dev/ttyTunNMEA1, soft link to /dev/ttyTun8 will be created named /dev/ttyNMEA1 which can be accessed from IOx.

The following command shows the state of the GPS:

```
IR1101#show app-hosting list
App id State
---------------------------------------------------------
gps RUNNING
```

Guest Shell as IOx Container APP

The Guest Shell is a virtualized Linux-based environment, designed to run custom Linux applications, including Python for automated control and management of Cisco devices. Using the Guest Shell, the user can also install, update, and operate third-party Linux applications and access the IOS CLI.

The Guest Shell environment is intended for tools, Linux utilities, and manageability rather than networking.

Guest Shell shares the kernel with the host (router) system. Users can access the Linux shell of Guest Shell and update scripts and software packages in the container rootfs. However, users within the Guest Shell cannot modify the host file system and processes.

The Guest Shell container is managed using IOx. IOx is Cisco's Application Hosting Infrastructure for Cisco IOS XE devices. IOx enables hosting of applications and services developed by Cisco, partners, and third-party developers in network edge devices, seamlessly across diverse and disparate hardware platforms.

The Guest Shell is typically bundled with the system image and can be installed using the **guestshell enable** Cisco IOS command. However, this approach leads to an increase of roughly 75MB in the size of the image. This is a problem for some users who have limited bandwidth, or download images through LTE.

With these users in mind, guestshell will be made available as a single tar file which can then be downloaded and installed on the system like any other IOX application. As a result, there won't be any increase in the size of the universal release image.



Note Day 0 guestshell provisioning will not work with this approach.

By default, Guest Shell allows applications to access the management network via the management interface. For platforms like the IR1101, which don't have a dedicated management port, a VirtualPortGroup can be associated with Guest Shell in the IOS configuration.

Sample guestshell configuration can be found here.

To install guestshell on the device, copy the tar file to the router and run the following command:

```
app-hosting install appid guestshell package <path to tar file>
```

Use the following command to check the status:

```
show app-hosting list
```

Once guestshell has been deployed successfully, standard guestshell commands such as **guestshell enable**, **guestshell run bash**, and **guestshell run python3** should work.

The following resource talks about running python scripts using guestshell:

CLI Python Module

Note Only python3 is supported in 17.5.1.

Important - Before You Install

Before attempting to install Guest shell on your device, please verify that the device has IOx container keys programmed on it by running the following command:

```
Router#show software authenticity keys | i Name
Product Name : SFP-VADSL2-I
Product Name : SFP-VADSL2-I
Product Name : IR1101
Product Name : IR1101
Product Name : Cisco Services Containers
Product Name : Cisco Services Containers
```

The output should contain one or more lines with the Product Name “Cisco Services Containers”. If the device doesn’t have container keys programmed on it, then you won’t be able to install guest shell.

You will see an error like the following:

```
*Aug 26 15:47:21.484: %IOSXE-3-PLATFORM: R0/0: IOx: App signature verification failed with
 non-zero exit code
*Aug 26 15:47:21.588: %IM-6-INSTALL_MSG: R0/0: ioxman: app-hosting: Install failed: App
package signature (package.sign)
verification failed for package manifest file package.mf. Re-sign the application and then
 deploy again.
```

There is no software based mechanism to install container keys on the device. The keys have to be programmed at the manufacturing facility. IR1100 devices shipped after January 1, 2020, should have the container keys programmed.

The guest shell tar file is published along with the IOS-XE image for a given release. More information can be found here:https://developer.cisco.com/docs/iox/#!iox-resource-downloads/downloads

Application Hosting on the IR1101 Router

This section describes the application-hosting characteristics specific to the IR1101 Industrial Router.

Note

- The IR1101 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.

IOx URL Access Methods

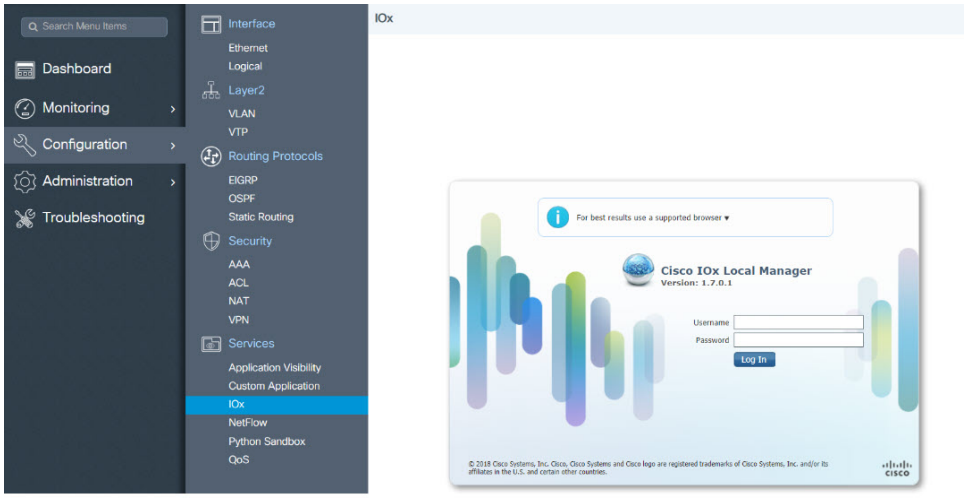
The IOx URL can be accessed in two different ways.

1. Using the direct URL to the IOx login.
2. Navigate to the IOx login through the Web User Interface (WebUI)

The syntax for the first method is **https://***IR1101-IP-ADDRESS***/iox/login**

The syntax for the second method is **https://***IR1101-IP-ADDRESS* and then navigate to IOx as shown in the following:

Figure 56: 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 life-cycle in the Cisco IOx Local Manager Reference Guide.

IOX URL User Restriction

The second method will make the configuration of the entire router available to IOx users. In some organizations, the IOx users are different from those that manage and administer the router. In this case, there is a need for restricting the access of the IOx users to ONLY the IOx Local Manager WebUI and not the entire WebUI of the router.

Currently, IOx users are configured as privilege 15 users. To restrict the IOx users to ONLY the Local Manager, the following commands can be used:

```
Router(conf)# no ip http server
Router(conf)# ip http secure-server
Router(conf)# ip http session-module-list list2 OPENRESTY_PKI,NG_WEBUI
Router(conf)# ip http secure-active-session-modules list2
```

The command **no ip http server** will turn off the web server without https. The next command **ip http secure-server** is to turn on the https mode.

If you include only **OPENRESTY_PKI AND NG_WEBUI**, then you will be enabling ONLY the IOX local manager modules, and hence ALL users can ONLY access the IOX local manager if they have privilege 15, https://IR1101-IP-ADDRESS/iox/login.

And for ALL user, the WebUI access , **https://***IR1101-IP-ADDRESS* will be disabled.

Note This method will disable the main web page **https://***IR1101-IP-ADDRESS* for all users and will enable only **https://***IR1101-IP-ADDRESS***/iox/login** for all users. Use this method if you do not use the IR1101 main router WebUI for general administration and configuration.

VirtualPortGroup

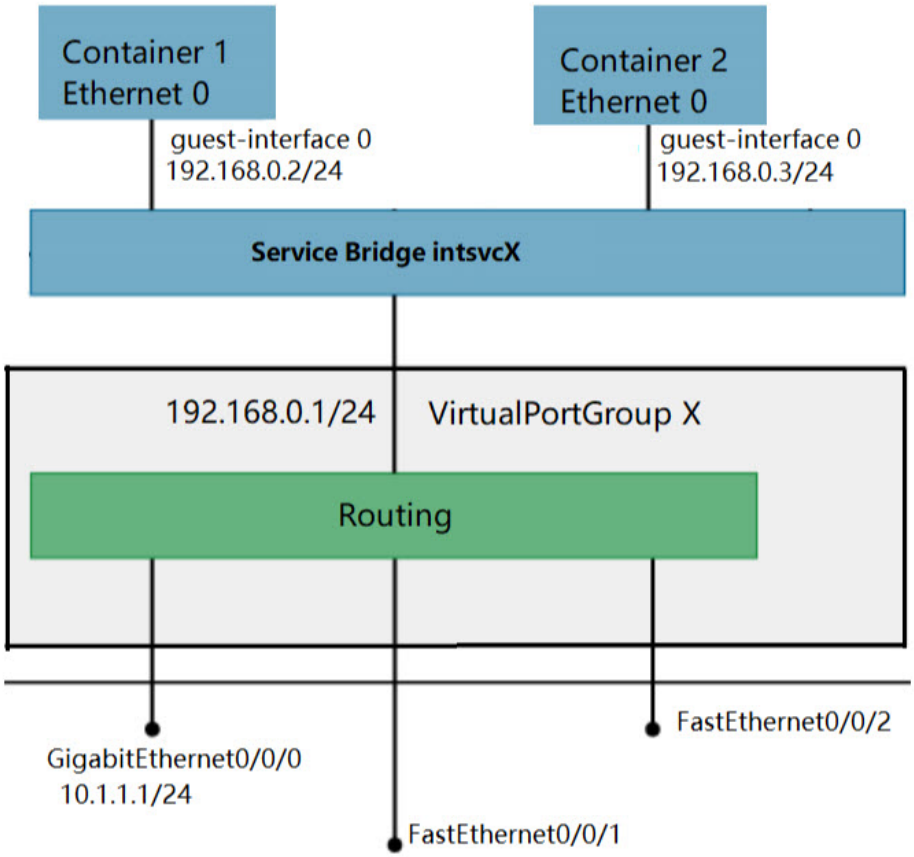
The VirtualPortGroup is a software construct on Cisco IOS that maps to a Linux bridge IP address. As such, the VirtualPortGroup represents the switch virtual interface (SVI) of the Linux container. Each bridge can contain multiple interfaces; each mapping to a different container. Each container can 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 57: 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

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.

DETAILED STEPS

Steps	Command	Purpose
1.	**enable** Example: `Device>enable`	Enables pr EXEC mo Enter your password i prompted.
2.	**configure terminal** Example: `Device#configure terminal`	Enters glo configurat mode.
3.	**iox** Example: `Device(config)#iox`	Enables IO
4.	**ip http server** Example: `Device(config)#ip http server`	Enables th server on y or IPv6 sy
5.	**ip http secure-server** Example: `Device(config)#ip http secure-server`	Enables a HTTP (HT server.

Steps	Command	Purpos
6.	**username** *name* **privilege** *level* **password** {**0** | **7** | *user-password* }*encrypted-password* Example: `Device(config)#username cisco privilege 15 password 0 cisco`	Establi usernar authen system privileg the use The us privileg must be as 15.
7.	**end** Example: `Device(config-if)#end`	Exits in configu mode a to priv EXEC

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.

DETAILED STEPS

Step	Command
1.	**enable** Example: `Device>enable`
2.	**configure terminal** Example: `Device#configure terminal`
3.	**ip routing** Example: `Device(config)#ip routing`

Step	Command
4.	**interface type number** Example: `Device(config)#interface gigabitethernet 0/0/0`
5.	**no switchport** Example: `Device(config-if)#no switchport`
6.	**ip address ip-address mask** Example: `Device(config-if)#ip address 10.1.1.1 255.255.255.0`
7.	**exit** Example: `Device(config-if)#exit`
8.	**interface type number** Example: `Device(config)#interface virtualportgroup 0`
9.	**ip address ip-address mask** Example: `Device(config-if)#ip address 192.168.0.1 255.255.255.0`
10.	**end** Example: `Device(config-if)#end`
11.	**configure terminal** Enter configuration commands, one per line. End with CNTL/Z. Example: `Device#configure terminal`

Step	Command
12.	**app-hosting appid app1** Example: `Device(config)#app-hosting appid app1`
13.	**app-vnic gateway0 virtualportgroup 0 guest-interface 0** Example: `Device(config-app-hosting)#app-vnic gateway0 virtualportgroup 0 guest-interface 0`
14.	**guest-ipaddress 192.168.0.2 netmask 255.255.255.0** Example: `Device(config-app-hosting-gateway0)#guest-ipaddress 192.168.0.2 netmask 255.255.255.0`
15.	**app-default-gateway 192.168.0.1 guest-interface 0** Example: `Device(config-app-hosting-gateway0)#app-default-gateway 192.168.0.1 guest-interface 0`
16.	**end** Example: `Device#end`

Native docker support

Native Docker Support has been added to the 17.2.1 release. This feature enables users to deploy the docker applications on the IR1101. 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

Release 17.2.1 provides support for IOx container applications to be able to access the digital IO. There is a new CLI that has been added to the alarm contact command.

```
Router(config)# alarm contact ?
  <0-4>          Alarm contact number (0: Alarm port, 1-4: Digital I/O)
  attach-to-iox  Enable 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 IR1101. 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 IR1101 router. To install a signed application, signed verification has to be enabled on the device. Signed verification can be enabled using the **app-hosting verification enable** command.



Note Third-party application signing and the installation of third-party signed apps are currently not supported or available on this platform.

```
Router#app-hosting verification enable
```

After enabling the signed verification, you can install the application.

Installing and Uninstalling Apps

DETAILED STEPS

Step	Command
1.	**enable** Example: `Device>enable`
2.	**app-hosting install appid** *application-name* **package** *package-path* Example: `Device#app-hosting install appid lxc_app package flash:my_iox_app.tar`
3.	**app-hosting activate appid** *application-name* Example: `Device#app-hosting activate appid app1`
4.	**app-hosting start appid** *application-name* Example: `Device#app-hosting start appid app1`
5.	**app-hosting stop appid** *application-name* Example: `Device#app-hosting stop appid app1`

Step	Command
6.	**app-hosting deactivate appid** *application-name* Example: `Device#app-hosting deactivate appid app1`
7.	**app-hosting uninstall appid** *application-name* Example: `Device#app-hosting uninstall appid app1`

Overriding the App Resource Configuration

Resource changes will take effect only after the app-hosting activate command is configured.

DETAILED STEPS

Step	Command
1.	**enable** Example: `Device>enable`
2.	**configure terminal** Example: `Device#configure terminal`
3.	**app-hosting appid** *name* Example: `Device(config)#app-hosting appid app1`

Step	Command
4.	**app-resource profile** *name* Example: `Device(config-app-hosting)#app-resource profile custom`
5.	**cpu** *unit* Example: `Device(config-app-resource-profile-custom)# cpu 800`
6.	**memory** *memory* Example: `Device(config-app-resource-profile-custom)# memory 512`
7.	**vcpu** *number* Example: `Device(config-app-resource-profile-custom)# vcpu 2`
8.	**end** Example: `Device(config-app-resource-profile-custom)# end`

Verifying the Application Hosting Configuration

DETAILED STEPS

1. **enable**

Enables privileged EXEC mode. Enter your password if prompted.

Example:

```
Device>enable
```

2. show iox-service

Displays the status of all IOx services

Example:

```
Device# 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#
```

3. show app-hosting detail

Displays detailed information about the application.

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
  serial/syslog    iox_syslog          serial2
  serial/trace     iox_trace           serial3

Network interfaces
   ---------------------------------------
eth0:
   MAC address         : 52:54:dd:fa:25:ee
```

4. show app-hosting list

Displays the list of applications and their status.

Example:

```
Device#show app-hosting list
App id                           State
---------------------------------------------------------
app1                             RUNNING
```

Digital IO Enhancement

Support has been added to allow some digital I/O ports to be managed by IOSd, and some other digital IO ports to be managed by IOx container apps. An updated CLI has been added and the YANG model for Digital IO Enhancement has been updated.

The 17.5.1 version of the CLI is:

```
Router(config)# alarm contact attach-to-iox
```



Note With release 17.5.1, **alarm contact attach-to-iox** gave IOX control for ALL digital IO ports (1 thru 4).

The 17.6.1 version of the CLI is:

```
Router(config)#alarm contact 1 ?
application Set the alarm application
attach-port-to-iox Enable selected Digital IO Ports access from IOX
description Set alarm description
enable Enable the alarm/digital IO port
output Set mode as output
severity Set the severity level reported
threshold Set the digital IO threshold
trigger Set the alarm trigger

Router(config)#alarm contact 1 attach-port-to-iox

Router#show alarm
Alarm contact 0:
Not enabled.
Digital I/O 1:
Attached to IOX.
Digital I/O 2:
Not enabled.
Digital I/O 3:
Not enabled.
Digital I/O 4:
Not enabled.
```

In the updated CLI, <1-4> are the number of digital I/O ports to assign to IOx for container apps.



Note With release 17.6.1, each digital IO port can be assigned to IOX individually.

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)# no switchport
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
```

IOx Access to USB Storage

Customers have requested the ability to mount the host a USB thumb drive within the Docker container running on IOx. The bootflash has a limited number of read/write cycles, and a container continuously writing on the eMMC would prematurely wear out the unit. Using the USB thumb drive will allow Docker containers to write in a continuous manner without compromising the integrity of the bootflash.

Feature Requirements and Limitations

The following apply to this feature:

- The filesystem types supported for USB thumb drives on the IR1101 are: VFAT, EXT2 and EXT3. However, IOx only supports mounting of USB thumb drives with EXT2 and EXT3 filesystem. Cisco recommends EXT3 for the following reasons:
  - EXT3 is a journaling filesystem, which means there are not fragmentation issues.
  - Read/Writes are significantly faster with EXT3 filesystems
  - VFAT has a 4 GB maximum file-size limitation, which is a problem with container continuously writing large files.
- If the USB thumb drive is removed while a write operation by IOx apps is in progress, all the files included in the copy operation will be lost.
- If the USB thumb drive is removed while IOX and the app are using it, IOX will still be in running state. The functionality of the app using USB thumb drive as storage will be severely impacted, since it will not be able to read and/or write on the USB thumb drive.

Making the USB Thumb Drive Available to the IOx App

In order to make the USB thumb drive available to the IOx app, you need to issue a run option. See the following example:

```
Router(config-app-hosting-docker)#run-opts 1 "-v /mnt/usb0:/usbflash0"
```

This command will mount the USB thumb drive file system within the IOx application filesystem, and it will be available in the /usbflash0 folder, as showed by the following log from an IOx application:

```
/ # ls -al usbflash0/
total 705424
drwxrwxrwx    4 root     root          4096 Nov 10 22:42 .
drwxr-xr-x    1 root     root          4096 Nov 15 17:22 ..
-rw-r--r--    1 65534    65534    720025859 Nov 10 22:46 ir1101-universalk9.SSA.bin
drwx------    2 65534    65534        16384 Nov  8 16:32 lost+found
#
```

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

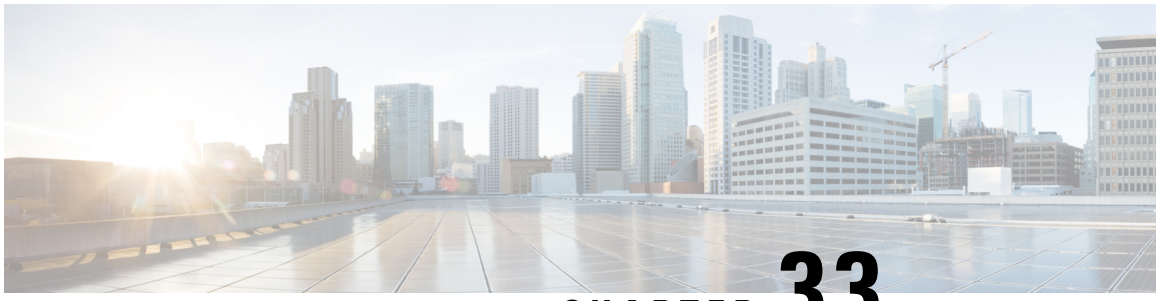
The Cisco ThousandEyes Enterprise Agent enables you to view end-to-end paths across networks and services affecting your business. It actively monitors network traffic paths across internal, external, and internet networks, helping analyze network performance and application availability.

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 33

Cisco SD-WAN Support

This chapter contains the following sections:

- Cisco SD-WAN Overview, on page 357
- SD-WAN, on page 358
- Related Documentation, on page 359

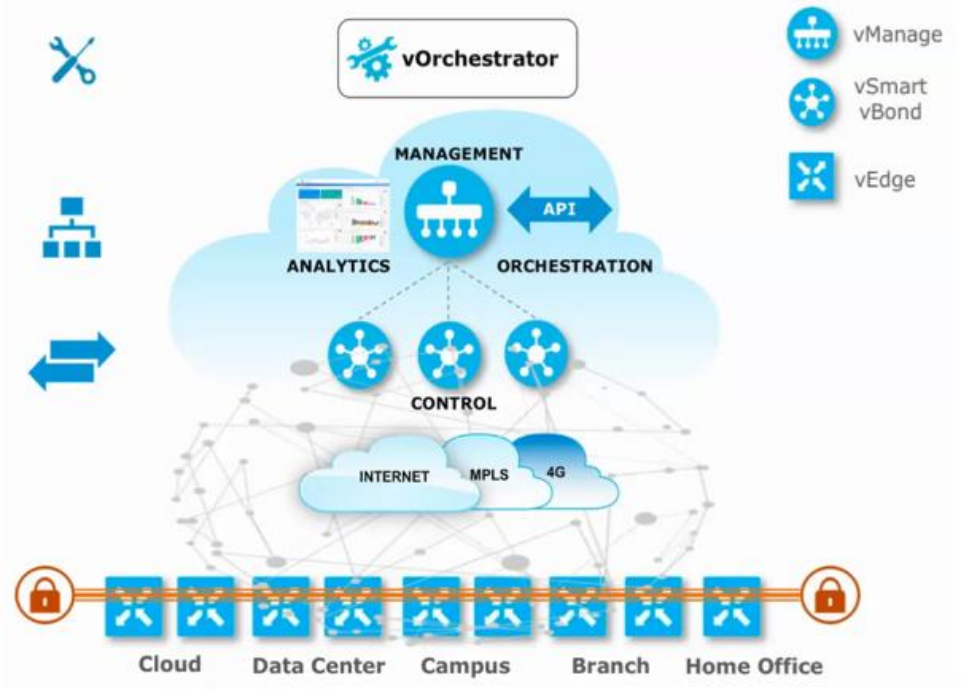
Cisco SD-WAN Overview

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 IR1101 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 the Gi0/0/0 or Cellular interface.

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

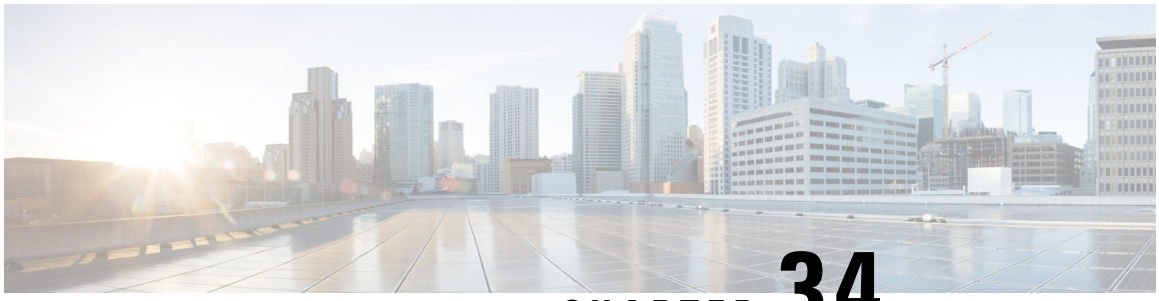
Cisco SDWAN documentation is available from the following sources:

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



CHAPTER 34

Serial Relay Service

This chapter contains the following sections:

- Serial Relay Service Overview, on page 361
- Data Paths, on page 361
- Configuration Commands, on page 363

Serial Relay Service Overview

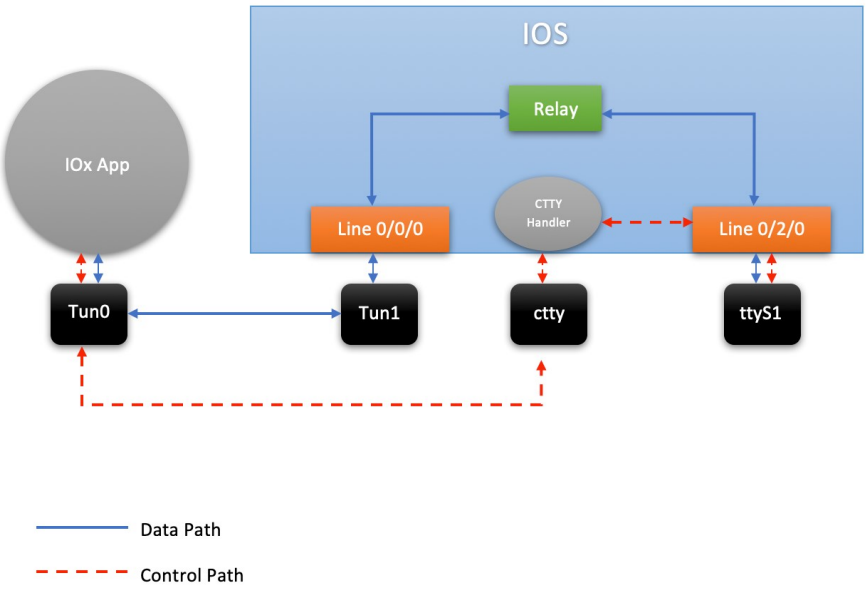
Serial Relay service on the IR1101 enables IOx apps to communicate with the Async Serial port (/dev/ttyS1 under IOS-XE). The configuration of Serial Relay service is similar to that of the IR800.

Data Paths

On the IR1101, 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 58: 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 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) based on buffer availability.
5. The reverse path functions the same with the roles of /dev/ttyS1 and /dev/tun0 reversed.

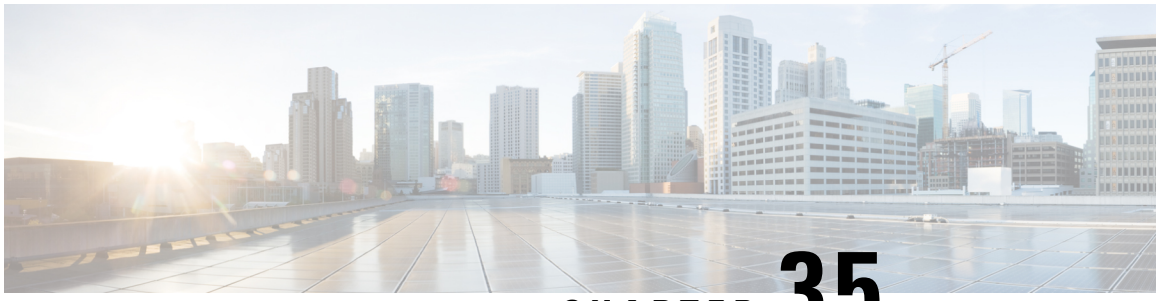
Control Path:

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 driver.
7. Line driver of Line 0/2/0 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

```
IR1101#configure terminal
IR1101(config)#interface async 0/2/0
IR1101(config-if)#encapsulation relay-line
IR1101(config-if)#exit
IR1101(config)#relay line 0/2/0 0/0/0
IR1101(config)#exit
IR1101#
```



CHAPTER 35

ROM Monitor

This chapter contains the following sections:

- ROM Monitor Overview, on page 365
- Access ROM Monitor Mode, on page 366
- Displaying the Configuration Register Setting, on page 368
- Environment Variable Settings, on page 369
- Exiting ROM Monitor Mode, on page 370
- Async serial port for console , on page 371

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.

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 IR1101 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 show romvar.

```
Router#show rom-monitor r0
System Bootstrap, Version 1.2, RELEASE SOFTWARE
Copyright (c) 1994-2018  by cisco Systems, Inc.

Router# show romvar
ROMMON variables:
 PS1 = rommon ! >
 MCP_STARTUP_TRACEFLAGS = 00000000:00000000
 LICENSE_SUITE =
 RET_2_RTS =
 Diagnostic = 1
 THRPUT =
 USER_BOOT_PARAM =  DEBUG_CONF=/bootflash/debug.conf
 EULA_ACCEPTED = TRUE
```

```
 BOOT_WDOG = DISABLE
 LICENSE_BOOT_LEVEL =
 BOOT = bootflash:ir1101_crashkernel.bin,1;
 CRASHINFO = bootflash:crashinfo_RP_00_00_20180619-204307-UTC
 RET_2_RCALTS =
 BSI = 0
 RANDOM_NUM = 1662155698

Router# reload
```

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 ! >
MCP_STARTUP_TRACEFLAGS = 00000000:00000000
 LICENSE_SUITE =
 RET_2_RTS =
 Diagnostic = 1
 THRPUT =
 USER_BOOT_PARAM =  DEBUG_CONF=/bootflash/debug.conf
 EULA_ACCEPTED = TRUE
 BOOT_WDOG = DISABLE
 LICENSE_BOOT_LEVEL =
 BOOT = bootflash:ir1101_crashkernel.bin,1;
 CRASHINFO = bootflash:crashinfo_RP_00_00_20180619-204307-UTC
 RET_2_RCALTS =
 BSI = 0
 RANDOM_NUM = 1662155698
```

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 27: 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.

Rommon Command 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="IR1101 rommon ! > "
IR1101 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 “IR1101 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 IR1101 boot loader does not support netboot, so any setting like environment variables IP_ADDRESS, IP_SUBNET_MASK, DEFAULT_GATEWAY, TFTP_SERVER, TFTP_FILE are not used.

Table 28: 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 **showmon** command :

```
rommon 1 > showmon
System Bootstrap, Version 1.3(REL), RELEASE SOFTWARE
Copyright (c) 1994-2018 by cisco Systems, Inc.

IR1101-K9 platform with 4188160 Kbytes of main memory

MCU Version - Bootloader: 4, App: 4
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 = IR1101-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 IR1101-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

	Command or Action	Purpose
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 366 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.	Enters the privileged EXEC mode after the boot is complete.
Step 4	Run the **show rom-monitor** *slot* command.	Verifies whether the ROMmon has been upgraded.

Async serial port for console

The IR1101 console port is a USB port. Some installations require that the console port be an RS232 port. This release provides a workaround that allows the Async 0/2/0 port to be used as a console port.

This change requires to ROMMON variables as well as IOS XE. You will need to setup both Mini-USB console and Async 0/2/0 with the same baudrate and 8-N-1.

To change the ROMMON variable:

1. Access ROMMON by following the procedure in the IR1101 Software configuration Guide.
2. Set the ROMMON variable CONSOLE_SERIAL with value as 1 using the command in ROMMON: set CONSOLE_SERIAL=1
3. sync

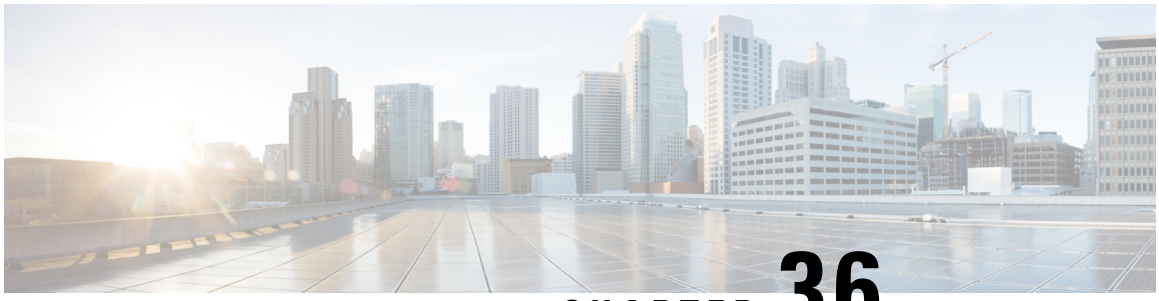
When ROMMON detects CONSOLE_SERIAL=1, it should start to use the new variable. It will also pass console=ttyS1 as boot parameter instead of console=ttyS0.

After setting the ROMMON variable, then boot up the Cisco IOS XE 17.11.1a image. It will read the new variable and use console=ttyS1 as boot parameter instead of console=ttyS0. Cisco IOS XE 17.11.1a should update the new ROMMON image. Then, reboot the device again and setup auto boot if needed.

Note

- Async 0/2/0 pinout is EIA-TIA-561 DTE. When CONSOLE_SERIAL=1 is setup, Async 0/2/0 won't exist. Do NOT perform a factory reset or downgrade the software below 17.11.

  For more information, see Connecting and Troubleshooting IoT Devices Using Asynchronous Serial Interfaces White Paper



CHAPTER 36

Digital Subscriber Line

This chapter contains the following sections:

- Overview, on page 373
- DSL Feature Specifications, on page 375
- Installing the DSL SFP, on page 377
- Support for DSL Annex B, on page 381
- LED Indications on the SFP, on page 381
- DSL SFP Firmware Upgrade, on page 382
- Support DSL SFP Firmware Signing and Signature Validation, on page 383
- ADSL2/2+ Overview, on page 384
- VDSL2 Overview, on page 393
- Yang Model for DSL, on page 396
- Support Four ADSL MIB Objects, on page 397
- Support ADSL MIB Objects, on page 397
- Support VDSL MIB Objects, on page 397
- DSL Troubleshooting, on page 398

Overview

The router adds DSL capability by using a Small Form-factor Pluggable (SFP) network interface module. The DSL solution supports the following Annex:

ADSL2 (A), ADSL2+(A,J, where J only supported by the 17.5.1 release). VDSL2 supports Annex A,B. All in compliance with TR100, TR105, TR114, TR115.

IOS-XE release 17.5.1 adds in support for Annex-J configuration in the controller interface.

Note ADSL2+ J is supported, ADSL2 J is not yet supported in 17.5.1.

To enable Annex-J, perform the following:

```
router#config term
router(conf)#controller vdsl 0/0/0
router(conf-if)#capability annex-j
```

```
router#(conf-if)#exit
router#
```

To remove Annex-J, perform the following:

```
router#config term
router(conf)#controller vdsl 0/0/0
router(conf-if)#no capability annex-j
router#(conf-if)#exit
router#
```

17.5.1 adds in a new command **rx-padding**. This command is used for packets with an MTU less than 64 bytes.



Note If frames less than 64mtu are expected downstream from the service provider, the Vlan configuration must be vlan 96. If frames less than 64mtu are expected downstream from the service provider, only a Single VLAN is supported in a single-PVC, i.e.Vlan96. In future releases, there is plan to extend the range of vlan support to range from Vlan44 to 1024, single-vlan in single-pvc option.

The command example is as follows:

```
router#config term
router#controller vdsl 0/0/0
router(conf-if)#rx-padding
router(conf-if)#end
router#write mem
```

Feature Caveats

This section provides a list of what features are supported and unsupported.

- The DSL SFP operates only when inserted in the IR1101 base unit. It is NOT supported in the IRM-1100 expansion unit. The IR1101 can support only a single DSL SFP on GI0/0/0
- VDSL2 only supports profiles 8a through 17a, 30a is not supported.
- The SFP currently does not have Yang support. This will be provided in a future release.
- Supports Radius and AAA when authenticating and configuring DSL users.
- The DSL interface requires a minimum configuration dependent of the DSL services, therefore Plug and Play (PnP) features are not available on the DSL interface.
- Zero-Touch-Deployment (ZTD) is only supported through IIoT Field Network Director. From FND, use cgna wsma based ZTD only, PnP based ZTD is not supported over the DSL interface. For ZTD, stage with basic minimum configuration and parameters depending on the service provider requirements.

  The IR1101 must be on IOS-XE 17.4.1 or above for DSL support.
- The show controller vdsl 0/0/0 command is used to display all DSL [VDSL2/ADSL2/ADSL2+] controller information, similar to the c111x platforms. Although the controller command is vdsl, is actually means dsl and is used for adsl and vdsl alike.
- For ADSL2/2+ configurations, there is no ATM interface as with c111x platforms. All configurations are on the DSL SFP WAN g0/0/0 interface, its sub-interface options, and controller vdsl0/0/0 itself. ATM packets are handled by the DSL SFP and re-assembled as Ethernet packets. Annex A, L is supported.

- Using the WebUI, interface g0/0/0 can be configured/monitored as normal. No specific options to monitor/configuration option for Controller vdsl 0/0/0 on release 17.4.1.
- VDSL2 and ADSL2+ various MIBS support only trickle in 17.5.1 and beyond releases. MIB information is available later in this section.
- For ADSL2/2+ ATM configuration, if your scenario expects frames <64 byte MTU downstream from Service Provider, please ensure following steps:
  1. rx-padding cli is enabled
  2. Vlan96 value is used in interface configuration
  3. There is no multi-VLAN support in single-PVC in this specific scenario

DSL Feature Specifications

Table 29: DSL Feature Specifications

Multimode DSL (VDSL2 and ADSL2/2+)	• Provided through a DSL SFP • SFP has a single RJ-45 interface • Support for double-ended line testing (DELT) diagnostics mode (VDSL2 Only)

Table 30: VDSL2 Feature Specifications

VDSL2	• VDSL2 993.2 Annex A and Annex B • 997 and 998 band plans • G.994.1 ITU G.hs • VDSL2 profiles: 8a, 8b, 8c, 8d, 12a, 12b, and 17a • Vectoring • U0 band support (25 to 276 kHz) • Ethernet packet transfer mode (PTM) based only on IEEE 802.3ah 64/65 octet encapsulation • Dying gasp

Table 31: ADSL2/2+ Feature Specifications

ADSL2/2+	• Annex A and L for ADSL2 • Annex A for ADSL2+ • Annex J for ADSL2+ (available in 17.5.1) • G.994.1 ITU G.hs • Reach-extended ADSL2 (G.922.3) Annex L for increased performance on loop lengths greater than 16,000 feet from central office • T1.413 ANSI ADSL2/2+ DMT issue 2 compliance • DSL Forum TR-067, and TR-100 conformity • Impulse noise protection (INP) and extended INP • Downstream power backoff (DPBO) • Dying gasp

Dying gasp is when the the router is using some residual power on capacity to send outage messages to the DSLAM. You can verify your router is ready to send out dying gasp messages by using the **show controller vdsl 0/0/0 local** command:

```
Router#show controllers vdsl 0/0/0 local
SFP Vendor PID: SFPV5311TR
SFP Vendor SN: V021932028C
Firmware embedded in IOS-XE: 1_62_8463
Running Firmware Version: 1_62_8455
Management Link: up
DSL Status: showtime
Dumping internal info: idle
Dying Gasp: armed
Dumping DELT info:  idle
```

Note If Dying Gasp is disabled, the output will show **Dying gasp: disarmed**.

There is no configuration for Dying Gasp. The Software takes care internally for the implementation. Once an SFP shut/no shut has been triggered, 1-2 notifications are sent within 50ns.

Dying-Gasp SMS Notification for EM74XX Modems

Prerequisites:

- Hardware Peripheral: P-LTEA-EA, P-LTEA-LA
- Initial Release: IOS-XE 17.5.1
- License: Cisco Network-advantage

Pluggable Interface Modules (PIMs) using the EM7430 or EM7455 modem have extra capacitors to supply power to the modem in case of loss of power to the module. This allows a graceful power off of the modem. When loss of power is detected, the modem is expected to send out dying gasp SMS when configured.

The following is an example of configuring dying gasp with a phone number and SMS message:

```
#controller Cellular 0/1/0
#lte dyinggasp sms send 9119110911 "Losing Power"
Warning: Enabling Dying Gasp SMS configuration completed successfully.
Please reset Modem for the changes to take effect
```

Configuration Steps

Step	Command	Purpose
1	configure terminal	Enters the global configuration mode.
2	controller Cellular <slot>	Enters the interface command mode for the cellular module controller slot.
3	lte dyinggasp detach enable	Enable dying-gasp feature with send detach request
4	lte dyinggasp sms send <phone number> <SMS message>	Configure the phone number to receive SMS text message and the content of text message to be sent by the modem when platform or module powered down.
5	exit	Exit configuration
6	write mem	Save changes to the router configuration

Configuration Example

The following example shows how to enable dying-gasp feature on cellular module in slot 0/1/0, specify phone number receiving the SMS, and the specific SMS text message to be sent by modem upon power failure.

```
router# configure terminal

router(config)# controller cellular 0/1/0
router (config-controller)# lte dyinggasp detach enable
router (config-controller)# lte dyinggasp sms send 4081112222
 IR1101-#999_EM7455_powered_off!
```

Installing the DSL SFP

Instructions for inserting the DSL SFP are found in your products Hardware Installation Guide.

Warning It is critical that the installer read these instructions and be familiar with the correct method of inserting and removing the SFP. Failure to do so may result in damage to the SFP.

The minimum IOS-XE release for DSL SFP support is 17.4.1 on the IR1101.

Basic Configuration

Once the SFP is installed, it requires a basic configuration to bring it up. Follow these steps:

```
configure t
Router(conf)#interface g0/0/0
Router(conf-if)#media-type sfp
Router(conf-if)#no shut
Router(conf-if)#exit
```

At this point, SFP insertion SYSLOG messages will appear.

SFP Verification

After safely installing the SFP, you can check its status with the **show inventory** command:

```
Router#show inventory

++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
INFO: Please use "show license UDI" to get serial number for licensing.
++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++

NAME: "Chassis", DESCR: "IR1101 Base Chassis"
PID: IR1101-K9 , VID: V03 , SN: FCW23500H5X

NAME: "Module 0 - Mother Board", DESCR: "Cisco IR1101 motherboard"
PID: IR1101-K9 , VID: V03 , SN: FOC23473SRK

NAME: "module subslot 0/0", DESCR: "IR1101-ES-5"
PID: IR1101-ES-5 , VID: V01 , SN:

NAME: "subslot 0/0 transceiver 0", DESCR: "GE T"
PID: SFP-VADSL2+-I , VID: V01 , SN: MET2023000A
Ignore the description, it will always reflect GE T for all IR1101 SFPs
PID and S/N are what matter
```

In the below output, ignore the Description and bitrate. The PID/Serial number information are true to the SFP.

```
Router#show interfaces transceiver detail
IDPROM for transceiver Gigabitethernet0/0/0:
Description = SFP or SFP+ optics (type 3)
Transceiver Type: = GE T (26)
Product Identifier (PID) = SFP-VADSL2+-I
Vendor Revision = V5.1
Serial Number (SN) = MET2023000A
Vendor Name = CISCO-METANOIA
Vendor OUI (IEEE company ID) = 00.00.00 (0)
CLEI code =
Cisco part number = 74-124941
Device State = Enabled.
Date code (yy/mm/dd) = 20/23/
Connector type = .
Encoding = 8B10B (1)
Nominal bitrate = GE (1300 Mbits/s)
Minimum bit rate as % of nominal bit rate = not specified
Maximum bit rate as % of nominal bit rate = not specified

Socket Verification


SFP IDPROM Page 0xA0:
000: 03 04 22 08 00 00 00 00 00 00
010: 00 01 0D 00 00 00 00 00 FF 00
```

```
020: 43 49 53 43 4F 2D 4D 45 54 41
030: 4E 4F 49 41 20 20 00 00 00 00
040: 53 46 50 56 35 33 31 31 54 52
050: 35 31 43 53 20 20 56 35 2E 31
060: 00 00 00 3F 08 00 00 00 4D 45
070: 54 32 30 32 33 30 30 30 41 20
080: 20 20 20 20 32 30 32 33 20 20
090: 20 20 00 00 00 6D 63 00 30 60
100: FE 53 E4 C1 54 F1 F1 C1 FA 1A
110: 98 EC 6B E0 7F 00 00 00 00 00
120: 00 00 00 00 8C D0 5C F7 00 00
130: 00 00 00 00 00 00 00 00 37 34
140: 2D 31 32 34 39 34 31 20 56 30
150: 31 20 CF EC 55 00 00 00 00 D4
160: 00 00 00 00 00 00 00 00 00 00
170: 00 00 00 00 00 00 00 00 00 00
180: 00 00 00 00 00 00 00 00 00 00
190: 00 00 53 46 50 2D 56 41 44 53
200: 4C 32 2B 2D 49 20 20 20 20 20
210: 20 20 00 00 17 00 00 00 00 00
220: 00 00 00 5A

SFP IDPROM Page 0xA2:
000: 00 00 00 00 00 00 00 00 00 00
010: 00 00 00 00 00 00 00 00 00 00
020: 00 00 00 00 00 00 00 00 00 00
030: 00 00 00 00 00 00 00 00 00 00
040: 00 00 00 00 00 00 00 00 00 00
050: 00 00 00 00 00 00 00 00 00 00
060: 00 00 00 00 00 00 00 00 00 00
070: 00 00 00 00 00 00 00 00 00 00
080: 00 00 00 00 00 00 00 00 00 00
090: 00 00 00 00 00 00 00 00 00 00
100: 00 00 00 00 00 00 00 00 00 00
110: 00 00 00 00 00 00 00 00 00 00
120: 00 00 00 00 00 00 00 00 00 00
130: 00 00 00 00 00 00 00 00 00 00
140: 00 00 00 00 00 00 00 00 00 00
150: 00 00 00 00 00 00 00 00 00 00
160: 00 00 00 00 00 00 00 00 00 00
170: 00 00 00 00 00 00 00 00 00 00
180: 00 00 00 00 00 00 00 00 00 00
190: 00 00 00 00 00 00 00 00 00 00
200: 00 00 00 00 00 00 00 00 00 00
210: 00 00 00 00 00 00 00 00 00 00
220: 00 00 00 00 00 00 00 00 00 00
230: 00 00 00 00 00 00 00 00 00 00
240: 00 00 00 00 00 00 00 00 00 00
250: 00 00 00 00 00 00
Link reach for 9u fiber (km) = SX(550/270m) (0)
1xFC-MM(500/300m) (0)
2xFC-MM(300/150m) (0)
ESCON-MM(2km) (0)
Link reach for 9u fiber (m) = SX(550/270m) (0)
1xFC-MM(500/300m) (0)
2xFC-MM(300/150m) (0)
ESCON-MM(2km) (0)
Link reach for 50u fiber (m) = SR(2km) (0)
IR-1(15km) (0)
IR-2(40km) (0)
LR-1(40km) (0)
LR-2(80km) (0)
LR-3(80km) (0)
DX(40KM) (0)
```

```
HX(40km) (0)
ZX(80km) (0)
VX(100km) (0)
1xFC, 2xFC-SM(10km) (0)
ESCON-SM(20km) (0)
Link reach for 62.5u fiber (m) = SR(2km) (0)
IR-1(15km) (0)
IR-2(40km) (0)
LR-1(40km) (0)
LR-2(80km) (0)
LR-3(80km) (0)
DX(40KM) (0)
HX(40km) (0)
ZX(80km) (0)
VX(100km) (0)
1xFC, 2xFC-SM(10km) (0)
ESCON-SM(20km) (0)
Nominal laser wavelength = 0 nm.
DWDM wavelength fraction = 0.0 nm.

No transceiver present
```

DSL SFP Annex J support

IOS-XE release 17.5.1 adds in support for Annex-J configuration in the controller interface.

Note ADSL2+ J is supported, ADSL2 J is not yet supported in 17.5.1.

To enable Annex-J, perform the following:

```
router#config term
router(conf)#controller vdsl 0/0/0
router(conf-if)#capability annex-j
router#(conf-if)#exit
router#
```

To remove Annex-J, perform the following:

```
To remove Annex-J:
router#config term
router(conf)#controller vdsl 0/0/0
router(conf-if)#no capability annex-j
router#(conf-if)#exit
router#
```

17.5.1 adds in a new command **rx-padding**. This command is used for packets with an MTU less than 64 bytes.

Important If frames less than 64mtu are expected downstream from the service provider, the Vlan configuration must be vlan 96.

The command example is as follows:

```
router#config term
router#controller vdsl 0/0/0
router(conf-if)#rx-padding
router(conf-if)#end
```

Execute **write mem** to save the configuration.

DSL SFP Annex M support

Support is the same as it was for Annex-J in 17.5.1

Support for DSL Annex B

For the 17.8.1 release, ADSL2+ Annex B will be supported.

Annex B is not configured by default. To enable Annex B, the following command will be used.

```
controller VDSL 0/0/0
  capability annex-b
```

LED Indications on the SFP

The DSL SFP has two LED indicators built into it. This LED operates independent of any LED that is on the panel of the Router.

Note There is no **show platform led** support for the SFP LED. Use the **show controller vdsl 0/0/0 local** command for DSL link status.

LED Indications

The following table describes the SFP LED indications:

Indicator LED	LED Color	State	Description
LED 1	Orange	On	CPE side (expected to be ON when used on an IR router)
LED 1	Orange	Off	Central office side (not supported
xDSL Status LED	Green	Slow Flash	Idle
xDSL Status LED	Green	Fast Flash	Training
xDSL Status LED	Green	Steady	Showtime
xDSL Status LED	Green	Extremely Rapid Flash	Packet Transmit

SFP LED Workflow

The following table describes the SFP LED indications during a bootup:

Before SFP is inserted	Off
During SFP bootup	Slow Green Flash
After auto-negotiation has completed	Solid Green
SFP shut triggered from the CLI	Off
SFP no shut triggered from the CLI	Flashing, then Solid Green
SFP Traffic	Flashing Green

Auto-Negotiation

You can tell the status of auto-negotiation based on the LED on the SFP. On shut/no shut or during auto-negotiation, the following sequence should be observed:

Slow Flashing Green	Idle
Fast Flashing Green	Training
Solid Green	Handshake success, Showtime

If the SFP LED is toggling between slow flashing green and fast flashing green, it usually means it is in auto-negotiation mode. If this continues for a long time, the DSLAM and Router DSL SFP parameters need to be rechecked. The following chapters cover more details on Router xDSL configuration.

DSL SFP Firmware Upgrade

The DSL SFP has firmware loaded on it. You should check the version loaded on the SFP and compare it to what is available in the router image. The customer should make their decision to upgrade according to their own agreement with their ISP.

The SFP must have a minimum configuration in order to upgrade it:

```
configure t
Router(conf)#interface g0/0/0
Router(conf-if)#media-type sfp
Router(conf-if)#no shut
Router(conf-if)#exit
```

Check your firmware levels by executing **show controller vdsl 0/0/0 local** command.

```
Router#show controllers vdsl 0/0/0 local
SFP Vendor PID: SFPV5311TR
SFP Vendor SN: V021932028C
Firmware embedded in IOS-XE: 1_62_8463
Running Firmware Version: 1_62_8455
Management Link: up
DSL Status: showtime
Dumping internal info: idle
Dying Gasp: armed
Dumping DELT info:  idle
```

Use the following command to upgrade the SFP:

```
Router#upgrade hw-module subslot 0/0 sfp 0
Upgrade SFP firmware on interface GigabitEthernet0/0/0 from 1_62_8455 to 1_62_8463
Connection will be disrupted, Continue(Y/N)?y
Start ebm upgrade!!
..................................................
..................................................
................
firmware update success!!
```

The command loads the new firmware, and then performs a shut/no shut on the interface to reset the SFP.

Note In 17.5.1 and beyond, the capability exists to upgrade standalone SFP Firmware, in addition to the SFP Firmware bundled with IOS image. For example:

```
Router#upgrade hw-module subslot 0/0 sfp 0 {flash|usbflash0|msata}:sfp_fw_image
```

MTU Limitation

As per the SFP Data sheet specification, the following are MTU limitations:

- For VDSL, the MTU range on the DSL SFP interface is between 64 - 1800 Bytes
- For ADSL2/2+, the MTU range on the DSL SFP interface is between 64 - 1700 Bytes

Support DSL SFP Firmware Signing and Signature Validation

An optional IOS filepath has been added to the end of the existing upgrade command. The file must be signed with SFP-VADSL2-I key. The file could be in bootflash:/flash:, usbflash0 or msata:. It cannot be from any remote file system.

Command Line Interface

The command line interface for upgrading the module follows:

router# **upgrade hw-module subslot 0/0 sfp 0** <*IOS filepath*>

Options to the command are:

```
Router#upgrade hw-module subslot 0/0 sfp 0 ?
  bootflash:  Firmware filename on local driver
  crashinfo:  Firmware filename on local driver
  flash:      Firmware filename on local driver
  usbflash0:  Firmware filename on local driver
```

The following is an example of the command usage:

```
Router#upgrade hw-module subslot 0/0 sfp 0 bootflash:sfp8455_rel.bin
Digital signature successfully verified in file bootflash:sfp8455_rel.bin
Upgrade SFP firmware on interface GigabitEthernet0/0/0 from 1_62_8463 to 1_62_8455
Connection will be disrupted, Continue(Y/N)?y
Start ebm upgrade!!
..........................................................................
..........................................................................
```

```
.....................................................................................
.....................................................................................

firmware update success!!
```

ADSL2/2+ Overview

This section provides an overview for ADSL2/2+

Important The Router SFP based DSL support differs in configuration and troubleshooting in comparison to other ISR DSL platforms. There is no ATM interface, ethernet to ATM packet translation is handled internally via Adaption Layer5 (AAL5). All configurations are on the controller vdsl 0/0/0 and g0/0/0 interface/sub-interface. UBR is recommended over AAL5.

All details are listed in the chapters that follow.

ADSL2/2+ works in auto mode (configuration on DSLAM auto-negotiation automatically with the DSL controller). Annex A is supported on ADSL2+. Annex A and reach-extended Annex L mode-1 is supported on ADSL2. This is in compliance with TR-100/TR-105

- For Auto-negotiation handshake procedure, the SFP is compliant with ITU-T G.994.1 DSL TRx and for Physical Layer Management compliant with ITU-T G.997.1 for DSL TRx.
- The DSL SFP complies with ITU-T G.99x standard with supporting AVD2 CPE mode only.
- Supports LLC/SNAP and VCMux ethernet bridged encapsulation option.
- All PPPoX encapsulation is configured via PPPoE only. Internally, packet translation is handled via ATM. There is no PPPoA configuration like there is with the c111x ISR.
- ADSL-PVC is configurable in the Controller VDSL 0/0/0: Each SFP supports 8 PVCs.
- Each PVC supports mapping to/from 802.1q Vlan tagging.
- VPI range is 0-255, VCI range is 32-65535.

The 'mode' reflected in **show controller vdsl 0/0/0** will always be PTM (Packet transfer mode). Internally packet translation to ATM is handled (AAL5).

Configuring ADSL2/2+

The Router supports Asymmetric Digital Subscriber Line (ADSL) 2/2+.

Procedure

	Command or Action	Purpose
Step 1	**enable** **Example:** `router> enable`	Enables privileged EXEC mode.

	Command or Action	Purpose
Step 2	**configure terminal** **Example:** `router# configure terminal`	Enters global configuration mode.
Step 3	**controller vdsl <port>** **Example:** `router(config)# controller vdsl 0/0/0`	Enters configuration mode for the ADSL2/2+ controller.
Step 4	**adsl-pvc <vpi/vci>** **Example:** `router(config-controller)#adsl-pvc 0/35`	Configures the PVC's VPI and VCI parameters. Refer to ADSL2/2+ PVC Sub Mode, on page 386 for detailed sub-commands.
Step 5	**bridge-dot1q <1-4094>** **Example:** `router(config-controller-adsl-pvc)#bridge-dot1q 2`	Configures the PVC's bridge-dot1q parameter.
Step 6	**encapsulation llcsnap|vcmux** **Example:** `router(config-controller-adsl-pvc)#encapsulation llcsnap`	Disabled by default. Can be either llcsnap or vcmux. This example shows the PVC encapsulation as LLCSNAP.
Step 7	**exit** **Example:** router(config-controller-adsl-pvc)# **exit**	Enables new configuration to take effect.
Step 8	**end** **Example:** `router(config-controller)# end`	Exits the configuration mode.

ADSL2/2+ Controller Configuration Commands

This section describes some of the CLI commands specific to controller configuration.

Brief	Format	Command Default	Description	Differences From Other IOS-XE ISRs
adsl-pvc	**adsl-pvc** *[name] {<vpi>/<vci>}* **adsl-pvc 0/35** **adsl-pvc PVC1 0/35**	None	ADSL2/2+ PVC Submode VPI/VCI value 0-255 VCI Value 32-65535 For additional details on the ADSL2/2+ submode, refer to ADSL2/2+ PVC Sub Mode, on page 386	VPI: 0-31 VCI: 1-1023

Brief	Format	Command Default	Description	Differences From Other IOS-XE ISRs
bitswap	—	Default is Enabled	Bitswap	—
carrier-set	**carrier-set** *[a43 a43c b43]*	a43 a43c b43	DSL SFP Carrier Set	c111x defines these tones under the modem vdsl option. For example, v43 has to be disabled via cli. In the Router, tone v43 is disabled by default.
default	—	—	Set a command to its defaults	—
description	—	—	Controller specific description	—
exit	—	—	Exit from controller configuration mode. This is mandatory in order to make the configuration take effect.	—
help	—	—	Description of the interactive help system	—
mac-address	**mac-address** <*MAC address*>	The default is the MAC is preconfigured.	DSL SFP MAC Address. There is no need to configure anything to get the controller working.	—
modem vdsl	—	—	Not applicable to the Router. Inherited from the c111x.	Applicable only in the c111x.
mpls	—	—	Not applicable to the Router. Inherited from the c111x.	Applicable only in the c111x.
no	—	—	Negate a command or set its defaults	—
shutdown	—	—	Shutdown vdsl controller	—
sra	—	Default is Enabled	Seamless Rate Adaption	—

ADSL2/2+ PVC Sub Mode

The following table lists related commands.

Brief	Format	Default	Description	Differences From Other IOS-XE ISRs
adsl-pvc	adsl-pvc vpi/vci	None	A maximum of 8 PVCs can be supported on a DSL interface. vci range 32 - 65535 vpi range 0-255	VPI/VCI value 0-31 VCI value 1-1023
bridge-dot1q	**bridge-dot1q** <*1-4094*>	None	802.1Q VLAN ID to PVC mapping	—
cbr	**cbr** <*peak cell rate*> cbr pcr range is 0 to 5500	No	Configure Constant Bit Rate (CBR) Service UBR is recommended over AAL5.	48-1408 in Kbps.
default-pvc	**default-pvc**	First PVC Created	Set PVC as default PVC The default-pvc command under adsl-pvc is an option available with the DSL SFP. It selects which PVC the DSL SFP will treat as the default when there are 2 or more active PVCs.	—
encapsulation	**encapsulation** <*llcsnap|vcmux*>	None	Configure ADSL2/2+ PVC Encapsulation	—
exit	—	—	Exit adsl-pvc sub commands	—
ubr	**ubr** <*peak cell rate*> ubr peak cell rate range is 0 to 5500	Yes	Configure Unspecified Bit Rate (UBR) Service	48-1408 in Kbps.
vbr-nrt	**vbr-nrt** <*peak cell rate*> <*sustainable cell rate*> pcr range is 0 to 5500 scr range is 0 to 5500	No	Configure Non Real-time Variable Bit Rate Service UBR is recommended over AAL5.	48-1408 in Kbps.
vbr-rt	**vbr-rt** <*peak cell rate*> <*sustainable cell rate*> pcr range is 0 to 5500 scr range is 0 to 5500	No	Configure Real-time Variable Bit Rate Service UBR is recommended over AAL5.	48-1408 in Kbps.

Brief	Format	Default	Description	Differences From Other IOS-XE ISRs
vlanid-rx	**vlanid-rx** *<1-4094>*	Depends on bridge-dot1q	Configure the DSL SFP to set the VLAN ID of the Ethernet packet received by the DSL SFP to be sent to the router. Used in conjunction with the DSL SFP VLAN operation vlanop-rx to either remove or replace the VLAN ID from the Ethernet packet.	Only on IoT Routers
vlanid-tx	**vlanid-tx** *<1-4094>*	Depends on bridge-dot1q	Configure the DSL SFP to set VLAN ID of the Ethernet packet for transmission to the network. Used in conjunction with the DSL SFP VLAN operation vlanop-tx to either remove or replace the VLAN ID from the Ethernet packet before transmitting the packet to the network.	Only on IoT Routers
vlanop-rx	**vlanop-rx** *<pass-through|remove|replace>*	Remove	Configure the VLAN ID operation of the DSL SFP to the Ethernet packet received by the DSL SFP to be sent to the router. Remove or replace VLAN operations are used in conjunction with the vlanid-rx. Pass-through option preserves the existing VLAN ID of the Ethernet packet.	Only on IoT Routers

Brief	Format	Default	Description	Differences From Other IOS-XE ISRs
vlanop-tx	**vlanop-tx** *<pass-through|remove|replace>*	Replace	Configure the VLAN ID operation of the DSL SFP to the Ethernet packet for transmission to the network. Remove or replace VLAN operation are used in conjunction with the vlanid-tx. Pass-through option preserves the existing VLAN ID of the Ethernet packet.	Only on IoT Routers

ADSL2+ Example

The following example is from an ADSL2+ configuration:

Note For an explanation of some of the key output messages, see Controller Status Messages, on page 407.

```
Router#show controller vdsl 0/0/0
Controller VDSL 0/0/0 is UP

Daemon Status: UP

XTU-R (DS) XTU-C (US)
Chip Vendor ID: 'META' 'BDCM'.
Chip Vendor Specific: 0x0000 0x0762
Chip Vendor Country: 0xB500 0xB500
Modem Vendor ID: 'META' ' '
Modem Vendor Specific: 0x0000 0x0000
Modem Vendor Country: 0xB500 0x0000
Serial Number Near: MET2023000A V5311TR 1_62_8463
Serial Number Far:
Modem Version Near: 1_62_8463 MT5311.
Modem Version Far: <value>

Modem Status: TC Sync (Showtime!)
DSL Config Mode: AUTO
Trained Mode: G.992.3 (ADSL2) Annex A

TC Mode: PTM
Selftest Result: 0x00
DELT configuration: disabled
DELT state: not running

Failed full inits: 0
Short inits: 0
Failed short inits: 0
```

```
Modem FW Version:
Modem PHY Version:
Modem PHY Source: System

Line 0:

XTU-R (DS) XTU-C (US)
Trellis: ON ON
SRA: enabled enabled.
SRA count: 0 0.
Bit swap: enabled enabled.
Bit swap count: 0 0
Line Attenuation: 2.4 dB dB
Signal Attenuation: 5.0 dB 0.0 dB
Noise Margin: 8.2 dB 6.5 dB
Attainable Rate: 12491 kbits/s 1153 kbits/s
Actual Power: 0.0 dBm 10.2 dBm
Total FECC: 0 0
Total ES: 0 399
Total SES: 0 188
Total LOSS: 0 177
Total UAS: 103 6325
Total LPRS: 0 0
Total LOFS: 0 0
Total LOLS: 0 0


DS Channel1 DS Channel0 US Channel1 US Channel0
Speed (kbps): NA 12491 NA 1093
SRA Previous Speed: NA 0 NA 0
Previous Speed: NA 12583 NA 1097
Reed-Solomon EC: NA 0 NA 0
CRC Errors: NA 209 NA 0
Header Errors: NA 0 NA 0
Interleave (ms): NA 1.00 NA 1.00
Actual INP: NA 0.00 NA 0.00
```

ADSL2 Annex A Example

The following example is from an ADSL2 Annex A configuration:



Note For an explanation of some of the key output messages, see Controller Status Messages, on page 407.

```
show controller vdsl 0/0/0
Controller VDSL 0/0/0 is UP
Daemon Status: UP
XTU-R (DS) XTU-C (US)

Chip Vendor ID: 'META' 'BDCM'
Chip Vendor Specific: 0x0000 0x0762
Chip Vendor Country: 0xB500 0xB500
Modem Vendor ID: 'META' ' '
Modem Vendor Specific: 0x0000 0x0000
Modem Vendor Country: 0xB500 0x0000
Serial Number Near: MET2023000A V5311TR 1_62_8463
Serial Number Far:

Modem Version Near: 1_62_8463 MT5311
Modem Version Far:
```

```
Modem Status: TC Sync (Showtime!)
DSL Config Mode: AUTO
Trained Mode: G.992.5 (ADSL2+) Annex A
TC Mode: PTM
Selftest Result: 0x00
DELT configuration: disabled
DELT state: not running

Failed full inits: 0
Short inits: 0
Failed short inits: 0

Modem FW Version:
Modem PHY Version:
Modem PHY Source: System

Line 0:
XTU-R (DS) XTU-C (US)

Trellis: ON ON
SRA: enabled enabled
SRA count: 0 0
Bit swap: enabled enabled
Bit swap count: 0 0
Line Attenuation: 1.4 dB dB
Signal Attenuation: 2.4 dB 0.0 dB
Noise Margin: 9.5 dB 6.3 dB
Attainable Rate: 23550 kbits/s 1105 kbits/s
Actual Power: 0.0 dBm 12.2 dBm
Total FECC: 1 0
Total ES: 1 396
Total SES: 0 317
Total LOSS: 0 287
Total UAS: 57 3344
Total LPRS: 0 0
Total LOFS: 0 0
Total LOLS: 0 0
DS Channel1 DS Channel0 US Channel1 US Channel0

Speed (kbps): NA 23550 NA 1105
SRA Previous Speed: NA 0 NA 0
Previous Speed: NA 23580 NA 1109
Reed-Solomon EC: NA 0 NA 0
CRC Errors: NA 95 NA 4
Header Errors: NA 0 NA 0
Interleave (ms): NA 1.00 NA 1.00
Actual INP: NA 0.00 NA 0.00
Training Log : Stopped
Training Log Filename : flash:vdsllog.bin
```

ADSL2 Annex L Example

The following example is from an ADSL2 Annex L configuration:



Note For an explanation of some of the key output messages, see Controller Status Messages, on page 407.

```
show controller vdsl 0/0/0
Controller VDSL 0/0/0 is UP
Daemon Status: UP
XTU-R (DS) XTU-C (US)
```

```
Chip Vendor ID: 'META' 'BDCM'
Chip Vendor Specific: 0x0000 0x0762
Chip Vendor Country: 0xB500 0xB500
Modem Vendor ID: 'META' ' '
Modem Vendor Specific: 0x0000 0x0000
Modem Vendor Country: 0xB500 0x0000
Serial Number Near: V0219320270 V5311TR 1_62_8463
Serial Number Far:

Modem Version Near: 1_62_8463 MT5311
Modem Version Far:
Modem Status: TC Sync (Showtime!)
DSL Config Mode: AUTO
Trained Mode: G.992.3 (ADSL2) Annex L
TC Mode: PTM
Selftest Result: 0x00
DELT configuration: disabled
DELT state: not running
Failed full inits: 0
Short inits: 0
Failed short inits: 0
Modem FW Version:
Modem PHY Version:
Modem PHY Source: System
Line 0:
XTU-R (DS) XTU-C (US)

Trellis: ON ON
SRA: enabled enabled
SRA count: 0 0
Bit swap: enabled enabled
Bit swap count: 0 0
Line Attenuation: 2.5 dB dB
Signal Attenuation: 5.7 dB 0.0 dB
Noise Margin: 7.0 dB 6.2 dB
Attainable Rate: 10164 kbits/s 288 kbits/s
Actual Power: 0.0 dBm 8.4 dBm
Total FECC: 0 0
Total ES: 6 0
Total SES: 6 0
Total LOSS: 6 0
Total UAS: 54 31
Total LPRS: 0 0
Total LOFS: 6 0
Total LOLS: 0 0
DS Channel1 DS Channel0 US Channel1 US Channel0

Speed (kbps): NA 10164 NA 243
SRA Previous Speed: NA 0 NA 0
Previous Speed: NA 12495 NA 1089
Reed-Solomon EC: NA 0 NA 0
CRC Errors: NA 0 NA 0
Header Errors: NA 0 NA 0
Interleave (ms): NA 1.00 NA 1.00
Actual INP: NA 0.00 NA 0.00
Training Log : Stopped
Training Log Filename : flash:vdsllog.bin
```

VDSL2 Overview

This section provides an overview for VDSL2,

The Router DSL SFP-VADSL2+-I provides VDSL2 Annex A, B support in conformance to ITU-T standards G.993.2 (VDSL2). This xDSL SFP is also in compliance with TR-114 (VDSL2 Annex A and B performance) and TR-115 (VDSL2 Feature validation tests by University of New Hampshire). The SFP complies with ITU-T G.99x standard with supporting AVD2 CPEmode only.

- Configurable Band Plan, conforms to North America Annex A (G.998) and Europe Annex B (G.997, 998) Band Plans subject to the 3072/4096 and 8-band/4-passband constraints.
- Supports all VDSL2 profiles (8a/b/c/d, 12a/b, 17a, 30a).
- Supports EU type Upstream Band 0 (US0).
- Complies with ITU-T G.994.1 Handshake Procedure for DSL TRx.
- Complies with ITU-T G.997.1 Physical Layer Management for DSL TRx
- Complies with ITU-T G.993.5 Self-FEXT Cancellation (Vectoring) for CPE mode
- Supports Robust Overhead Channel (ROC)
- Supports Online Reconfiguration (OLR) including Seamless Rate Adaptation (SRA) with D/L change and Bit Swapping
- Supports Upstream /Downstream Power Back Off (UPBO/DPBO)
- Supports DELT
- Supported maximum MTU size on VDSL2 is 1800 Bytes
- Standard compliance VDSL2 mode is PTM (Packet transfer mode)
- Supports VDSL2 Vectoring

For configuration and display commands, see the detailed sections below. The **show controller vdsl 0/0/0** is the fundamental command for validation.

Configuring VDSL2

The Router supports Very-high-bit-rate Digital Subscriber Line (VDSL2).

Procedure

	Command or Action	Purpose
Step 1	**enable** **Example:** `router> enable`	Enables privileged EXEC mode.
Step 2	**configure terminal** **Example:**	Enters global configuration mode.

	Command or Action	Purpose
	`router# configure terminal`	
Step 3	**controller vdsl 0/0/0** **Example:** `router(config-controller)# controller vdsl 0/0/0`	Enters configuration mode for the VDSL2 controller.
Step 4	**carrier-set** *a43 a43c b43* **Example:** `router(config-controller)# carrier-set a43|a43c|b43`	Configures the carrier set. Multiple choice. Default is a43 a43c b43. v43 is disabled by default.
Step 5	**end** **Example:** `router(config-controller)# end`	Exits controller configuration mode.

VDSL2 Controller Configuration Commands

This section describes some of the CLI commands specific to controller configuration.

Brief	Format	Command Default	Description
bitswap	—	Default is Enabled	Bitswap
capability	**capability** *[annex-j]*	None	Set the DSL SFP Capability
carrier-set	**carrier-set** *[a43 b43 a43c]*	a43 b43 a43c	DSL SFP Carrier Set
default	—	—	Set a command to its defaults
description	—	—	Controller specific description
exit	—	—	Exit from controller configuration mode
help	—	—	Description of the interactive help system
mac-address	**mac-address** <*MAC address*>	The default is the MAC is preconfigured.	DSL SFP MAC Address. There is no need to configure anything to get the controller working.
modem vdsl	—	N/A	Modem Configuration
mpls	—	—	Not applicable to the IoT Router. Inherited from the c111x.
no	—	—	Negate a command or set its defaults
shutdown	—	—	Shutdown vdsl controller

Brief	Format	Command Default	Description
sra	—	Default is Enabled	Seamless Rate Adaption

VDSL Example

The following example is from a VDSL configuration:

```
show controllers vdsl 0/0/0
Controller VDSL 0/0/0 is UP
Daemon Status: UP
XTU-R (DS) XTU-C (US)

Chip Vendor ID: 'META' 'IKNS'
Chip Vendor Specific: 0x0000 0x0101
Chip Vendor Country: 0xB500 0xB500
Modem Vendor ID: 'META' ' '
Modem Vendor Specific: 0x0000 0x2AB0
Modem Vendor Country: 0xB500 0x37A0
Serial Number Near: E80462D1B001 SFP-V5311-T-R 8431
Serial Number Far: ^A5u
Modem Version Near: 1_62_8431 MT5311
Modem Version Far: 6.7.0.15IK005010

Modem Status: TC Sync (Showtime!)
DSL Config Mode: AUTO
Trained Mode: G.993.2 (VDSL2) Profile 17a

TC Mode: PTM
Selftest Result: 0x00
DELT configuration: disabled
DELT state: not running

Failed full inits: 0
Short inits: 0
Failed short inits: 0

Modem FW Version:
Modem PHY Version:
Modem PHY Source: System

Line 0:
XTU-R (DS) XTU-C (US)

Trellis: ON ON
SRA: enabled enabled
SRA count: 0 0
Bit swap: enabled enabled
Bit swap count: 0 0
Line Attenuation: 2.7 dB dB
Signal Attenuation: 3.9 dB dB
Noise Margin: 7.2 dB 24.8 dB
Attainable Rate: 113289 kbits/s 86904 kbits/s
Actual Power: 9.3 dBm 8.1 dBm
Per Band Status: D1 D2 D3 U0 U1 U2 U3
Line Attenuation(dB): 0.0 1.5 2.5 N/A 0.2 0.2 0.6
Signal Attenuation(dB): 0.0 2.0 4.0 N/A 0.0 0.0 0.0
Noise Margin(dB): 0.0 7.2 7.2 0.0 24.7 24.8 24.8
Total FECC: 0 2203
Total ES: 1 2280
Total SES: 0 2199
Total LOSS: 0 2199
```

```
Total UAS: 81 2199
Total LPRS: 0 0
Total LOFS: 0 0
Total LOLS: 0 0
DS Channel1 DS Channel0 US Channel1 US Channel0

Speed (kbps): NA 103985 NA 50219
SRA Previous Speed: NA 0 NA 0
Previous Speed: NA 103985 NA 50219
Reed-Solomon EC: NA 0 NA 0
CRC Errors: NA 117 NA 1
Header Errors: NA 0 NA 0
Interleave (ms): NA 0.00 NA 0.02
Actual INP: NA 0.00 NA 0.00
Training Log : Stopped
Training Log Filename : flash:vdsllog.bin
```

For an explanation of some of the key output messages, see Controller Status Messages, on page 407.

Yang Model for DSL

YANG is a popular data modeling language to represent data sent over network management protocols such as NETCONF and RESTCONF.

The **Cisco-IOS-XE-controller-vdsl-oper** has been introduced to edit the Controller vdsl configurations which gives the yang support for the DSL.

An example of a typical yang response for edit config of the dsl controller follows:

```
<native xmlns="http://cisco.com/ns/yang/Cisco-IOS-XE-native">
          <controller>
            <VDSL xmlns="http://cisco.com/ns/yang/Cisco-IOS-XE-controller">
              <name>0/0/0</name>
              <adsl-pvc xmlns="http://cisco.com/ns/yang/Cisco-IOS-XE-adsl">
                <vpi-vci>255/65535</vpi-vci>
                <bridge-dot1q>21</bridge-dot1q>
                <encapsulation>vcmux</encapsulation>
              </adsl-pvc>
            </VDSL>
          </controller>
        </native>
      </nc:config></nc:edit-config></nc:rpc>
```

Note The Controller configurations can be retrieved using **get** and **get-config** operations with the Cisco-IOS-XE-native yang model.

Cisco IOS-XE Yang Data Models are found here:

https://github.com/YangModels/yang/tree/master/vendor/cisco/xe

Each release has a directory, and the 17.5.1 release is found under 1751.

Support Four ADSL MIB Objects

MIB support has been added to obtain the DSL line speed and attainable rate on the IR1101.

The new MIBS are shown below:

1.3.6.1.2.1.10.94.1.1.4.1.2 ADSL-LINE MIB:adslAtucChanCurrTxRate

1.3.6.1.2.1.10.94.1.1.5.1.2 ADSL-LINE MIB:adslAturChanCurrTxRate

1.3.6.1.2.1.10.94.1.1.2.1.8 ADSL-LINE MIB:adslAtucCurrAttainableRate

1.3.6.1.2.1.10.94.1.1.3.1.8 ADSL-LINE MIB:adslAturCurrAttainableRate

Command Line Interface

On a router with a DSL SFP connected to ADSL DSLAM, the following existing SNMP CLIs can be used to verify support for the above OIDs:

```
!configure SNMP Server
!---------------------
snmp-server community public RO
snmp-server manager
!
!verify MIB OIDs
!---------------
snmp get-next v2c 33.33.33.102 public oid 1.3.6.1.2.1.10.94.1.1.4.1.2
!
```

The following command can also be used to gather the MIB values from another SNMP Client (for example, a linux device):

```
$ snmpwalk -v 2c -c public 33.33.33.102 1.3.6.1.2.1.10.94.1.1.4.1.2
```

Support ADSL MIB Objects

The following ADSL MIB OID will be supported on the IR1101:

```
1.3.6.1.2.1.10.94.1.1.6.1.15 ADSL-LINE-MIB adslAtucPerfCurr15MinInits
1.3.6.1.2.1.10.94.1.1.6.1.22 ADSL-LINE-MIB adslAtucPerfCurr1DayInits
```

Support VDSL MIB Objects

The following VDSL MIB OID will be supported on the IR1101:

```
1.3.6.1.2.1.10.251.1.4.1.2.1.3    VDSL2-LINE-MIB   xdsl2PMLInitCurr15MTimeElapsed
1.3.6.1.2.1.10.251.1.4.1.2.1.4    VDSL2-LINE-MIB   xdsl2PMLInitCurr15MFullInits
1.3.6.1.2.1.10.251.1.4.1.2.1.5    VDSL2-LINE-MIB   xdsl2PMLInitCurr15MFailedFullInits
1.3.6.1.2.1.10.251.1.4.1.2.1.6    VDSL2-LINE-MIB   xdsl2PMLInitCurr15MShortInits
1.3.6.1.2.1.10.251.1.4.1.2.1.7    VDSL2-LINE-MIB   xdsl2PMLInitCurr15MFailedShortInits
1.3.6.1.2.1.10.251.1.4.1.2.1.10   VDSL2-LINE-MIB   xdsl2PMLInitCurr1DayTimeElapsed
1.3.6.1.2.1.10.251.1.4.1.2.1.11   VDSL2-LINE-MIB   xdsl2PMLInitCurr1DayFullInits
```

```
1.3.6.1.2.1.10.251.1.4.1.2.1.12   VDSL2-LINE-MIB   xdsl2PMLInitCurr1DayFailedFullInits
1.3.6.1.2.1.10.251.1.4.1.2.1.13   VDSL2-LINE-MIB   xdsl2PMLInitCurr1DayShortInits
1.3.6.1.2.1.10.251.1.4.1.2.1.14   VDSL2-LINE-MIB   xdsl2PMLInitCurr1DayFailedShortInits
1.3.6.1.2.1.10.251.1.4.1.1.1.2    VDSL2-LINE-MIB   xdsl2PMLCurr15MValidIntervals
1.3.6.1.2.1.10.251.1.4.1.1.1.3    VDSL2-LINE-MIB   xdsl2PMLCurr15MInvalidIntervals
1.3.6.1.2.1.10.251.1.4.1.1.1.4    VDSL2-LINE-MIB   xdsl2PMLCurr15MTimeElapsed
1.3.6.1.2.1.10.251.1.4.1.1.1.5    VDSL2-LINE-MIB   xdsl2PMLCurr15MFecs
1.3.6.1.2.1.10.251.1.4.1.1.1.6    VDSL2-LINE-MIB   xdsl2PMLCurr15MEs
1.3.6.1.2.1.10.251.1.4.1.1.1.7    VDSL2-LINE-MIB   xdsl2PMLCurr15MSes
1.3.6.1.2.1.10.251.1.4.1.1.1.8    VDSL2-LINE-MIB   xdsl2PMLCurr15MLoss
1.3.6.1.2.1.10.251.1.4.1.1.1.9    VDSL2-LINE-MIB   xdsl2PMLCurr15MUas
1.3.6.1.2.1.10.251.1.4.1.1.1.10   VDSL2-LINE-MIB   xdsl2PMLCurr1DayValidIntervals
1.3.6.1.2.1.10.251.1.4.1.1.1.11   VDSL2-LINE-MIB   xdsl2PMLCurr1DayInvalidIntervals
1.3.6.1.2.1.10.251.1.4.1.1.1.12   VDSL2-LINE-MIB   xdsl2PMLCurr1DayTimeElapsed
1.3.6.1.2.1.10.251.1.4.1.1.1.13   VDSL2-LINE-MIB   xdsl2PMLCurr1DayFecs
1.3.6.1.2.1.10.251.1.4.1.1.1.14   VDSL2-LINE-MIB   xdsl2PMLCurr1DayEs
1.3.6.1.2.1.10.251.1.4.1.1.1.15   VDSL2-LINE-MIB   xdsl2PMLCurr1DaySes
1.3.6.1.2.1.10.251.1.4.1.1.1.16   VDSL2-LINE-MIB   xdsl2PMLCurr1DayLoss
1.3.6.1.2.1.10.251.1.4.1.1.1.17   VDSL2-LINE-MIB   xdsl2PMLCurr1DayUas
1.3.6.1.2.1.10.251.1.4.1.3.1.3    VDSL2-LINE-MIB   xdsl2PMLHist15MMonitoredTime
1.3.6.1.2.1.10.251.1.4.1.3.1.4    VDSL2-LINE-MIB   xdsl2PMLHist15MFecs
1.3.6.1.2.1.10.251.1.4.1.3.1.5    VDSL2-LINE-MIB   xdsl2PMLHist15MEs
1.3.6.1.2.1.10.251.1.4.1.3.1.6    VDSL2-LINE-MIB   xdsl2PMLHist15MSes
1.3.6.1.2.1.10.251.1.4.1.3.1.7    VDSL2-LINE-MIB   xdsl2PMLHist15MLoss
1.3.6.1.2.1.10.251.1.4.1.3.1.8    VDSL2-LINE-MIB   xdsl2PMLHist15MUas
1.3.6.1.2.1.10.251.1.4.1.3.1.9    VDSL2-LINE-MIB   xdsl2PMLHist15MValidInterval
1.3.6.1.2.1.10.251.1.4.1.4.1.3    VDSL2-LINE-MIB   xdsl2PMLHist1DMonitoredTime
1.3.6.1.2.1.10.251.1.4.1.4.1.4    VDSL2-LINE-MIB   xdsl2PMLHist1DFecs
1.3.6.1.2.1.10.251.1.4.1.4.1.5    VDSL2-LINE-MIB   xdsl2PMLHist1DEs
1.3.6.1.2.1.10.251.1.4.1.4.1.6    VDSL2-LINE-MIB   xdsl2PMLHist1DSes
1.3.6.1.2.1.10.251.1.4.1.4.1.7    VDSL2-LINE-MIB   xdsl2PMLHist1DLoss
1.3.6.1.2.1.10.251.1.4.1.4.1.8    VDSL2-LINE-MIB   xdsl2PMLHist1DUas
1.3.6.1.2.1.10.251.1.4.1.4.1.9    VDSL2-LINE-MIB   xdsl2PMLHist1DValidInterval
1.3.6.1.2.1.10.251.1.4.2.1.1.2    VDSL2-LINE-MIB   xdsl2PMChCurr15MValidIntervals
1.3.6.1.2.1.10.251.1.4.2.1.1.3    VDSL2-LINE-MIB   xdsl2PMChCurr15MInvalidIntervals
1.3.6.1.2.1.10.251.1.4.2.1.1.4    VDSL2-LINE-MIB   xdsl2PMChCurr15MTimeElapsed
1.3.6.1.2.1.10.251.1.4.2.1.1.5    VDSL2-LINE-MIB   xdsl2PMChCurr15MCodingViolations
1.3.6.1.2.1.10.251.1.4.2.1.1.6    VDSL2-LINE-MIB   xdsl2PMChCurr15MCorrectedBlocks
1.3.6.1.2.1.10.251.1.4.2.1.1.7    VDSL2-LINE-MIB   xdsl2PMChCurr1DayValidIntervals
1.3.6.1.2.1.10.251.1.4.2.1.1.8    VDSL2-LINE-MIB   xdsl2PMChCurr1DayInvalidIntervals
1.3.6.1.2.1.10.251.1.4.2.1.1.9    VDSL2-LINE-MIB   xdsl2PMChCurr1DayTimeElapsed
1.3.6.1.2.1.10.251.1.4.2.1.1.10   VDSL2-LINE-MIB   xdsl2PMChCurr1DayCodingViolations
1.3.6.1.2.1.10.251.1.4.2.1.1.11   VDSL2-LINE-MIB   xdsl2PMChCurr1DayCorrectedBlocks
1.3.6.1.2.1.10.251.1.4.2.2.1.3    VDSL2-LINE-MIB   xdsl2PMChHist15MMonitoredTime
1.3.6.1.2.1.10.251.1.4.2.2.1.4    VDSL2-LINE-MIB   xdsl2PMChHist15MCodingViolations
1.3.6.1.2.1.10.251.1.4.2.2.1.5    VDSL2-LINE-MIB   xdsl2PMChHist15MCorrectedBlocks
1.3.6.1.2.1.10.251.1.4.2.2.1.6    VDSL2-LINE-MIB   xdsl2PMChHist15MValidInterval
1.3.6.1.2.1.10.251.1.4.2.3.1.3    VDSL2-LINE-MIB   xdsl2PMChHist1DMonitoredTime
1.3.6.1.2.1.10.251.1.4.2.3.1.4    VDSL2-LINE-MIB   xdsl2PMChHist1DCodingViolations
1.3.6.1.2.1.10.251.1.4.2.3.1.5    VDSL2-LINE-MIB   xdsl2PMChHist1DCorrectedBlocks
1.3.6.1.2.1.10.251.1.4.2.3.1.6    VDSL2-LINE-MIB   xdsl2PMChHist1DValidInterval
```

DSL Troubleshooting

This section provides information for troubleshooting and debugging if the DSL control and/or datapath is not up.

Problem: If WAN interface g0/0/0 is DOWN:

Solution: Try the following:

- Check L1 cabling, networking, and with different SFP

- Capture output for **show int g0/0/0**, **show run all**, and **show version**
- Check if g0/0/0 has **media-type sfp** configuration set and the interface is unshut.
- Try another SFP to see if that is detected.
- Check SFP's LED status. Refer to LED Indications on the SFP, on page 381

Problem: If controller state is DOWN:

For example:

```
Router#show controllers vdsl 0/0/0
Controller VDSL 0/0/0 is DOWN
```

Solution: Try the following:

- Check L1 cabling.
- Try inserting RJ11 cable into an RJ11 male to RJ45 female connector to see if it helps align.
- Ensure Running FW is the same as System FW. If not, upgrade the SFP FW. Refer to DSL SFP Firmware Upgrade, on page 382.
- Gather output for all L1 Training logs. Ensure L1 debug logs in folder are sent to Cisco TAC, as well as the output of service internal command **test vdsl option 0x0 6**, and the output from **show controller 0/0/0 local**. Refer to L1 Training Logs, on page 408.
- Possible workaround: After gathering the above logs, try to reboot the router to see if it recovers. If it still does not work, try to hot remove/insert the SFP again.

Problem: If the controller is UP, but **show controller vdsl 0/0/0** shows the DSL Link Idle.

Solution: Try the following:

- Ensure **show controller vdsl 0/0/0 local** shows Running FW = System FW. If not, upgrade FW and shut/no shut g0/0/0. Refer to DSL SFP Firmware Upgrade, on page 382
- Ensure carrier-set match (in controller vdsl 0/0/0) configuration with DSLAM
- Restart DSLAM interface if any config changes have been made
- Fine-tune the Power Spectrum Density, Freq Bandplan, profile, operating mode, vlan, etc... on the DSLAM end. On the Router DSL controller end, auto mode is the default and no configuration is required except possibly carrier-set. For example: If DSLAM only supports POTS, recommended to set carrier-set to a43. By default, Cisco allows a43, a43c, b43.
- Ensure the DSLAM profile ONLY includes supported Profiles, bands, etc as per VDSL2/ADSL2/2+ Refer to the tables in DSL Feature Specifications, on page 375.
- When using the service internal command **test vdsl rawcli "basic show summary 1"** consecutively, do you see the status move from Idle/Handshake/Training back to Idle, or stuck in Idle? If former case, recheck DSLAM profile configs. If latter, share L1 debug logs.

- If the DSLAM has the same configuration that used to work, and then after an image upgrade, or new SFP change the controller is UP but no negotiation , then please provide following to Cisco:
  - SFP LED status
  - Capture **show version**, **show running-config**, **show run all | sec controller**, **show interface gigabitethernet 0/0/0**, and **show controller vdsl 0/0/0 local**.
- Possible workaround: After providing logs to Cisco, attempt to write erase and reload the router. Also, shut/no shut the DSLAM interface tied to this device, and unplug/plug SFP and cables again.

Problem: If the controller is Up, but the daemon is Down.

Solution: Try the following:

- Enable debug vdsl for debug, share with Cisco TAC
- Provide last known working configs and software version
- Possible workaround: After providing logs to Cisco, attempt to write erase and reload the router. Also, shut/no shut the DSLAM interface tied to this device, and unplug/plug SFP and cables again.
- Check if the appropriate datak9, securityk9, and network-advantage licenses are enabled on both Peer and Client.

Problem: If Controller is up, profile with DSLAM up in **show controller vdsl 0/0/0**, but Dialer did not acquire IP

Solution: Try the following:

- Check routes
- Check the output of **debug dialer** to see if it offers any information. If dialer idle time is resetting too soon, modify dialer idle-timeout (default is 120s , which ideally should be enough).
  - Ensure there are SW Licenses (datak9, securityk9, and network-advantage) on both PPPoE server and the PPPoE Client/CPE.
  - The following is a basic Dialer configuration that works:

```
interface Dialer1
ip address negotiated
no ip redirects
encapsulation ppp
dialer pool 1
dialer-group 1
no cdp enable
ppp authentication chap callin
ppp chap hostname WORD
ppp chap password 0 WORD
ppp ipcp route default
!
ip route 0.0.0.0 0.0.0.0 Dialer1 (or any route that works in user environment)
```

- Ensure PPPoE Server authentication credentials match PPPoE client
- If using DHCP, ensure the Server has enough addresses to lease out
- Enable debug ppp session and debug ip dhcp server packet detail on the headend/Peer router to debug if we receive any packets. Enable debug ppp session on router.
- If the above steps did not resolve the issue, provide all of the above debug information to Cisco TAC, along with the following:
  - Output of **show version**, **show running**, **show run all | sec controller**, **show controller vdsl 0/0/0** and **show controller vdsl 0/0/0 local**.
  - Output of service internal commands **test vdsl rawcli "basic show summary 1"**, **basic show summary 1**, and **test vdsl option 0x0 6**.
  - Configuration of the DSLAM.
  - L1 training logs. Refer to L1 Training Logs, on page 408.
- Possible workaround: After gathering the above logs in sequence for Cisco, you can try to write erase and reload Peer and Router. Specifically removing the Dialer interface with PPP configurations and reapplying. As a last resort, try to shut/no shut DSLAM interface attached to this Router DSL SFP interface. Additionally, to isolate behavior, validate this SFP on another Router if available. If it works, then validate multiple SFPs on same Router (to narrow down if it is an SFP or Router issue).

Problem: If controller is Up, Dialer is Up, but Dialer did not acquire IP, Authentication works only with PAP and does not work with CHAP.

Solution: Suppose there is a scenario where:

show controller vdsl 0/0/0 shows showtime

show pppoe session shows PPP session established.

Then we see Virtual Access bound with Dialer successfully, but still Dialer didn't acquire an IP with PAP config in dialer all as well, but CHAP would not work On PPPoE Server end, it showed CHAP authentication passed and device ack too, but still IP not acquiring on PPPoE Client/device end.

debug ppp packet showed everything was okay, but still IP not acquiring. In such cases, enable following to monitor: **debug ppp authentication** enabled, we may notice that after successful chap handshake, there was another attempt by our device/client to validate based on local hostname set on Router CLI required to disable, if there is default local hostname set for chap in Router client (or any IOS router):

```
config t
service internal
Int Dialer1
no ppp chap ignoreus
no shut
exit
```

For further information see the Understanding and Configuring PPP CHAP Authentication link:https://www.cisco.com/c/en/us/support/docs/wan/point-to-point-protocol-ppp/25647-understanding-ppp-chap.html

Problem If controller is up, Dialer acquired an IP, but cannot self-ping Dialer or ping PPPoE Server

Solution: Try the following:

- Ensure the appropriate SW licenses (datak9, securityk9, and network-advantage) are enabled on both the PPPoe Server and Client
- Verify if icmp is enabled on PPPoE client session (enable via access list)
- Ensure pap/chap authentication match is seen in **debug pppoe session**.
- show pppoe session should reflect session (virtual-access binding with Dialer)
- For PPPoE session debugging, this section is common to all IOS platforms: https://www.cisco.com/c/en/us/td/docs/routers/ir910/software/release/1_0/configuration/guide/ir910scg/swpppoe.pdf
- Apply Static IP on g0/0/0 DSL interface and check if you can ping the DSLAM and Peer (to isolate DSL SFP issues)
- The following is a Basic PPPoE Server and PPPoE client configuration that works, presuming PPPoE Server is a Cisco IOS device as well:

```
PPPoE Server
ip dhcp excluded-address 41.41.41.1 41.41.41.9
!
ip dhcp pool 41-41-41-pool
network 41.41.41.0 255.255.255.0
default-router 41.41.41.1
 lease 2
!
username dslpeer password 0 dslpeerpass
!!
bba-group pppoe global
virtual-template 1
!
interface GigabitEthernet0/0/0
no ip address
media-type sfp
!
interface GigabitEthernet0/0/0.1
encapsulation dot1Q 1 native
ip address 41.41.41.1 255.255.255.0
pppoe enable group global
!
interface Virtual-Template1
ip unnumbered GigabitEthernet0/0/0.1
peer default ip address dhcp-pool 41-41-41-pool
ppp authentication pap chap
!
 >>>>>> Add routes as relevant, next hop being the IP that Router Dialer acquires
!
ip route 10.0.0.0 255.255.255.0 41.41.41.3 >> dialer ip, change as necessary

PPPoE Client:
controller VDSL 0/0/0
Carrier-set a43 >>> Can set to whichever [a43, b43, a43c, v43 depending on DSLAM support]
interface GigabitEthernet0/0/0
no ip address
media-type sfp
!
interface GigabitEthernet0/0/0.1
encapsulation dot1Q 1 native
```

```
pppoe enable group global
pppoe-client dial-pool-number 1
!
interface Dialer1
ip address negotiated
no ip redirects
encapsulation ppp
dialer pool 1
dialer-group 1
no cdp enable
ppp authentication chap callin
ppp chap hostname dslpeer
ppp chap password 0 dslpeerpass
ppp ipcp route default
!
ip route 0.0.0.0 0.0.0.0 Dialer1
```

Problem: If DSL traffic has been going through for a while, however bandwidth drops in time:

Solution: Try the following:

- Ensure DSLAM profile PSD, Freq band plan configurations are fine-tuned (in such cases, ideally unrelated to Router DSL SFP).
- Ensure ip arp timeout is increased in the Cisco Router DSL interface, Dialer interface - this may specially help in bursty traffic or during congestion.



Note The following commands may be helpful for troubleshooting:

Interface Status:

```
Router#show ip interface brief
Use this command to validate if Dialer acquired an IP address
```

Inventory Status:

```
Router#show inventory

++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
INFO: Please use "show license UDI" to get serial number for licensing.
++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++

NAME: "Chassis", DESCR: "IR1101 Base Chassis"
PID: IR1101-K9 , VID: V03 , SN: FCW23500H5X

NAME: "Module 0 - Mother Board", DESCR: "Cisco IR1101 motherboard"
PID: IR1101-K9 , VID: V03 , SN: FOC23473SRK

NAME: "module subslot 0/0", DESCR: "IR1101-ES-5"
PID: IR1101-ES-5 , VID: V01 , SN:

NAME: "subslot 0/0 transceiver 0", DESCR: "GE T"
```

```
PID: SFP-VADSL2+-I , VID: V01 , SN: MET2023000A
Ignore the description, it will always reflect GE T for all ISR Router SFPs
PID and S/N are what matter
```

Commands to display the running software details:

```
Router#show running-config all
Router#dir flash:
Router#dir nvram:
Router#show version
```

There are some debugging commands that will also reflect the status of auto-negotiation:

```
Router#configure terminal
Router#service internal
Router#exit
The following test command will reflect auto-negotiation status:
Router#test vdsl rawcli "basic show summary 1"
Link time Rate US/DS Mode Status Annex TxPkts/RxPkts
4 1097/12491 ADSL2 Showtime AnnexA 0/0
```

Frequently Asked Questions

This section provides answers to some common questions.

Question: How can I set VDSL2 or ADSL2/2+ to a specific Annex and profile in Controller?

Answer: The Router DSL SFP operates in auto mode only. There are no options to configure on the SFP controller end. You can only make changes on the DSLAM side.

Question: There is no Controller ADSL option to configure.

Answer: Controller vdsl 0/0/0 is common nomenclature across Cisco IOS-XE products. The same cli is valid for ALL DSL protocols - VDSL2, ADSL2, ADSL2+.

Question: There is no ATM interface to configure.

Answer: There is no ATM interface for user configuration. On all configuration options on controller vdsl 0/0/0 and DSL SFP WAN interface g0/0/0 and its sub-interface options, ATM packets are handled by the DSL SFP and re-assembled as Ethernet packets. Annex A, L is supported.

Question: The training log in show controller vdsl 0/0/0 is not working. There is no option to start/stop.

Answer: This option is only specific to the c111x platform and not the Router DSL SFP. For Router platform L1 debugging, refer to: L1 Training Logs, on page 408

Question: Where can I download DSL SFP Firmware?

Answer:

In 17.5.1 and beyond, standalone FW is available to upgrade via Flash:, mSATA and usbflash0: in IOS To upgrade DSL Firmware, refer to: DSL SFP Firmware Upgrade, on page 382

Question: ADSL2 Annex L is not working.

Answer: Ensure the DSLAM profile configuration has the right bit rate allowed. Since the Router DSL SFP is auto-mode, it will negotiate with the profile with the highest bit rate (so this is predominantly determined by DSLAM config fine-tuning).

Question: Annex-L Mode1 is supported, but not Mode2.

Answer: Ensure that unsupported modes/profiles/bands in DSLAM configuration are disabled. Refer to DSL Feature Specifications, on page 375 for supported specifications.

Question: In ADSL2/2+ if burst size (peak cell rate and sustainable cell rate) are configured to the maximum 5500, dialer keeps flapping.

Answer: If Dialer is flapping, could be receiving Peer upstream and was unable to handle high rate of downstream traffic. Either disable **`ip keepalive`** in dialer configuration, or increase default keepalive to the maximum.

Question: How many PVCs are allowed?

Answer: 8

Question: Controller configurations are not taking effect.

Answer: Ensure you exit out of controller configuration mode for the configuration to take effect. As a workaround, shut/no shut the controller interface. Ideally this should reflected the moment you 'exit' out of controller config mode. Check the DSLAM for matching profile criteria, unsupported bands/profiles should be removed as they may delay the Handshake.

Question: In ADSL2/2+ controller configuration, Maximum Burst Size configuration is not taking affect.

Answer: When configuring either nrt-VBR or rt-VBR, only the configuration of Peak Cell Rate (PCR) and Sustainable Cell Rate (SCR) are supported. The optional Maximum Burst Size (MBS) is not supported.

Question: System hangs during L1 Debug Logs capture, taking very long. show commands are not working.

Answer: When **debug vdsl controller 0/0/0 dump** *internal folder_name* is executed, it drains most of the system resources. A warning syslog to that effect is displayed as well. This takes approximately 10 minutes to complete depending on state of controller. Multiple times during the process the controller is shut/no shut, during this activity do NOT intervene. Once complete, you should observe 'DONE' in syslog and prompted to shut/no shut g0/0/0.

Question: Are there any new SNMP MIBS added?

Answer: Release 17.5.1 introduced the following ADSL2+ MIBS:

- 1.3.6.1.2.1.10.94.1.1.4.1.2 ADSL-LINE MIB:adslAtucChanCurrTxRate
- 1.3.6.1.2.1.10.94.1.1.5.1.2 ADSL-LINE MIB:adslAturChanCurrTxRate
- 1.3.6.1.2.1.10.94.1.1.2.1.8 ADSL-LINE MIB:adslAtucCurrAttainableRate
- 1.3.6.1.2.1.10.94.1.1.3.1.8 ADSL-LINE MIB:adslAturCurrAttainableRate

Question: SFP is stuck in the Router.

Answer: This can occur on older models of the IR1101. There was a faceplate rework.

Follow these steps to ensure the SFP Latch is handled cautiously (as with all SFPs). When doing a hot removal of SFP:

- Remove the latch (hear the click) and tilt to ~ 45deg - 90deg, without pressuring it or forcing it to snap.
- Remove the cable.
- Remove the SFP



Caution When inserting the SFP, make sure you hear it lock in. Insert the cable and then close the latch. You should hear the click again. If you force the latch and it breaks, the SFP will be stuck in the Router. Workaround is to remove the faceplate and remove the SFP.

Controller Status Messages

This section explains some of the key output messages from the **show controller vdsl 0/0/0** command.

Refer to the following table:

Output message	Description
Controller VDSL 0/0/0 is UP	State of the controller
Daemon Status: UP	State of internal IOS DSL Daemon
Chip Vendor ID: 'META' 'BDCM'.	SFP Metanoia Chip information
Chip Vendor Specific: 0x0000 0x0762	SFP Metanoia Chip Information burnt in EEPROM programming
Chip Vendor Country: 0xB500 0xB500	SFP Metanoia Chip information
Modem Vendor ID: 'META'	SFP Metanoia Chip information
Modem Vendor Specific: 0x0000 0x0000	SFP Metanoia Chip information
Modem Vendor Country: 0xB500 0x0000	SFP Metanoia Chip information
Serial Number Near: MET2023000A V5311TR 1_62_8463	SFP Metanoia Chip information
Serial Number Far:	SFP Metanoia Chip information, ignore if empty, Serial Number Near is the value required
Modem Version Near: 1_62_8463 MT5311.	Modem Firmware information
Modem Version Far: <value>	Ignore if empty, the above Near version is what is important
Modem Status: TC Sync (Showtime!)	Shows L1 SFP auto-negotiation status. When SFP is shut/no shut, you see following auto-negotiation sequence: Idle , Handshake, Training, Showtime! Showtime implies auto-neg complete
DSL Config Mode: AUTO	Always in AUTO mode, no specific CLI to configure for ADSL2/2+, VDSL2
Trained Mode: G.992.3 (ADSL2) Annex A	Specifies ITU and Annex type
TC Mode: PTM	Always in Packet Transfer Mode, even for ADSL2/+. The SFP is already translating ATM to Ethernet frames.
SRA: enabled enabled.	Default is enabled
Bit swap: enabled enabled.	Default is enabled

L1 Training Logs

To configure the device perform the following:

```
Router#configure terminal
Router#service internal
Router#logging console
Router#exit
```

To configure debug, perform the following:

```
Router#debug vdsl sfp debug | error | event | info | packet For SFP level debuging
```

```
Router#debug vdsl controller 0/0/0 dump internal {dir} For L1 debugging
```

When the L1 debug dump starts you should see the following:

```
%VDSL_SFP_MGR-5-DUMP_START: Dump internal info started on interface GigabitEthernet0/0/0
```



Important **At this point, the device is unusable. Wait approximately 10 minutes until it completes.**

At that point you should see the following:

```
%VDSL_SFP_MGR-4-DUMP_DONE: Dump internal info done, please shut/no shut on interface
GigabitEthernet0/0/0 to recover
```

To recover the device into normal operational mode, preform the following:

```
Router#configure terminal
Router#interface g0/0/0
Router#shut
Router#no shut
Router#exit
```

Provide directory logs saved in bootflash: to Cisco.



Note Cisco recommends that each time you start a new log or debug, save it to a new directory rather than append to the existing information.

To enable Metanoia SFP debug commands, perform the following:

```
Router#configure terminal
Router#service internal
Router#exit
Router#test vdsl rawcli "basic show summary 1" This command shows the L1 auto-negotiation
status
Link time Rate US/DS Mode Status Annex TxPkts/RxPkts
773 1089/23628 ADSL2+ Showtime AnnexA 470/338

Router#test vdsl option 6 0x0 If functional, State = 2 should display. This command shows
basic L1 bringup of DSL SFP and it's states. Provide to Cisco for L1 troubleshooting.
Debug flags: 0x8000
Seq 0: slot=0 slot_port=0 bay=0 port=0 Name:MetaMgr0_0_0
MetanoiaPort=0 SFP type: 1 State: 2 cnt=855
MAC:00:00:00:00:00:00 Choice:0
hw interface:GigabitEthernet0/0/0 sw interface:GigabitEthernet0/0/0
Firmware file: /etc/SFP_V5311-T-R_CSP.b, size=491520, version=1_62_8463
SFP version: 1_62_8463
Notification Seq: 0x1 cnt: 0xB3 Stat Cycle:255
```

```
VDSL State: 5
EBM Tx: 21039 Rx: 21031
EBM Wait Timeout: 8 Rx Loss: 0
G994 vid CO: BDCM CPE: META
Serial No CO: CPE: MET2023000A V5311TR 1_62_8463
Version CO: CPE: 1_62_8463 MT5311
Capability CO: 000000000001000000 CPE: 000000000001000000
Line Attn: UP: 65535 DOWN: 13
```

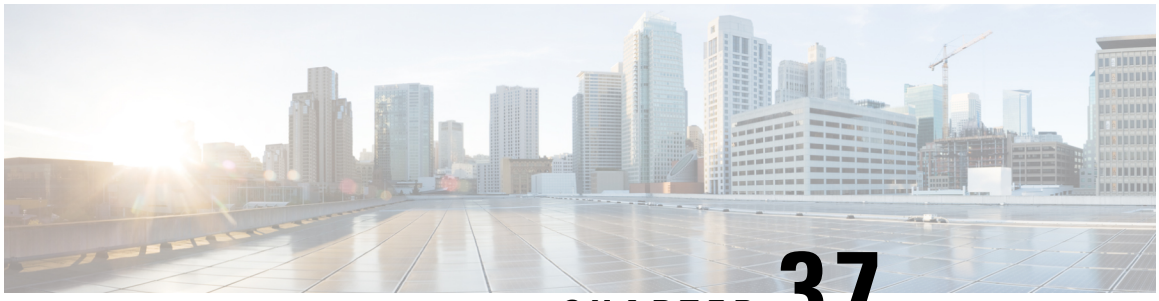
Tips for resetting the SFP:

- Ideally g0/0/0 shut/no shut will work in most cases (for example: after firmware upgrade, hot OIR, etc).

For hard reload of SFP, perform the following:

```
Router#hw-module subslot 0/0 reload
```

This option will force the entire subslot to reload, including the software module. So if connectivity is via telnet/ssh you might lose access for 1-2 minutes, and then all messages/syslogs buffered will print out.



CHAPTER 37

Out Of Band Management

This chapter contains the following sections:

- Out Of Band Management (OOB) Overview, on page 411
- OOB Topology, on page 411
- Feature Caveats, on page 412
- OOB Configuration, on page 412

Out Of Band Management (OOB) Overview

OOB offers a method for connecting two routers together with a USB cable for extra redundancy in case of 4G failure. This allows you to retain out-of-band connectivity by connecting the USB port for Router A to the USB console of Router B, as well as the ability to access Router B console port from Router A.

This feature will need to be implemented with IOS CLI. The user should be able to do a reverse telnet via tty line (/dev/ttyUSB) to another router's USB console.

OOB Topology

The following graphic illustrates the physical connection between two IR1101 routers:

Figure 59: Topology



The blue line above is a USB 2.0 Type A to USB 2.0 mini USB Type B cable. Refer to this topology for the following configuration.

Feature Caveats

Prior to configuring each router, ensure that both routers have a basic serial configuration:

```
line con 0
 stopbits 1
 speed 9600
```

Note Depending on how old the IR1101 is, the default baud rate is 9600 or 115200.

- Plug and Play is not supported. Cable must be installed prior to configuration.
- OOB only works for async0/2/1, which is the USB port. Async0/2/0 is the serial interface on the IR1101
- To exit from the feature, press “Ctrl-Shift-6”, then “x”, then “disconnect”.

OOB Configuration

Refer to the previous figure for examples of Router A and Router B. To access Router B console from Router A:

Power on Router A and configure the following:

```
interface Async0/2/1
 ip address 20.0.0.1 255.0.0.0
 encapsulation relay-line
!
line 0/2/1
 transport input all
 transport output all
```

Make sure that the speed of line 51 is the same speed as the console on Router B:

```
IR1101-A#show line

   Tty Line Typ     Tx/Rx    A Modem  Roty AccO AccI  Uses  Noise Overruns  Int
*     0    0 CTY              -    -      -    -    -     4      0    0/0      -
  0/0/0    2 TTY      0/0     -    -      -    -    -     0      0    0/0      -
  0/2/0   50 TTY   9600/9600  -    -      -    -    -     4      0    0/0      -
  0/2/1   51 TTY   9600/9600  -    -      -    -    -     4      0    0/0      -
     74   74 VTY              -    -      -    -    -     3      0    0/0      -
     75   75 VTY              -    -      -    -    -     1      0    0/0      -
     76   76 VTY              -    -      -    -    -     0      0    0/0      -
     77   77 VTY              -    -      -    -    -     0      0    0/0      -
     78   78 VTY              -    -      -    -    -     0      0    0/0      -
     79   79 VTY              -    -      -    -    -     0      0    0/0      -
     80   80 VTY              -    -      -    -    -     0      0    0/0      -
     81   81 VTY              -    -      -    -    -     0      0    0/0      -
     82   82 VTY              -    -      -    -    -     0      0    0/0      -
     83   83 VTY              -    -      -    -    -     0      0    0/0      -
     84   84 VTY              -    -      -    -    -     0      0    0/0      -
     85   85 VTY              -    -      -    -    -     0      0    0/0      -
     86   86 VTY              -    -      -    -    -     0      0    0/0      -
     87   87 VTY              -    -      -    -    -     0      0    0/0      -
     88   88 VTY              -    -      -    -    -     0      0    0/0      -

 Line(s) not in async mode -or- with no hardware support:
1, 3-49, 52-73, 89-735
```

Configure line 0/2/1 on Router A:

```
IR1101-A#configure term
Enter configuration commands, one per line.  End with CNTL/Z.
IR1101-A(config)#line 0/2/1
IR1101-A(config-line)#speed 9600
IR1101-A(config-line)#
```

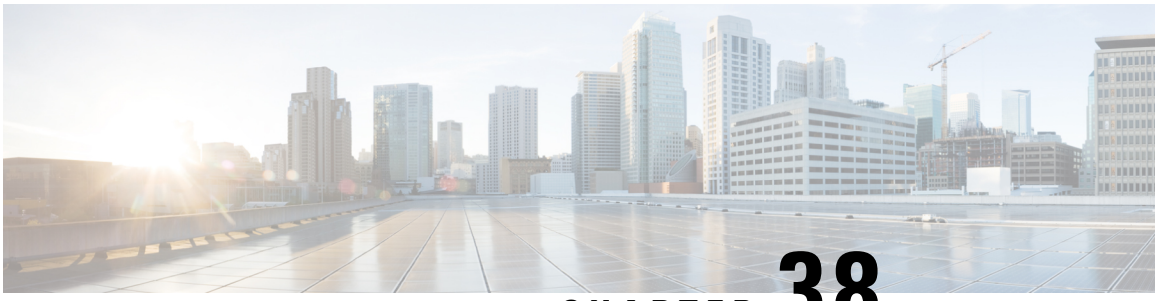
Telnet to Router B via Router A IP, port 2051:

```
IR1101-A#telnet 20.0.0.1 2051
Trying 20.0.0.1, 2051 ... Open


IR1101-B#

IR1101-B#   <== to exit, press “Ctrl-Shift-6”, then “x”, then “disconnect”

IR1101-A#disconnect
Closing connection to 20.0.0.1 [confirm]
```



CHAPTER 38

HDLC Support for SCATS

This chapter contains the following sections:

- HDLC Support for SCATS Overview, on page 415
- Configure IOx Application, on page 416
- Deploy SCATs Application, on page 419
- Troubleshooting, on page 427

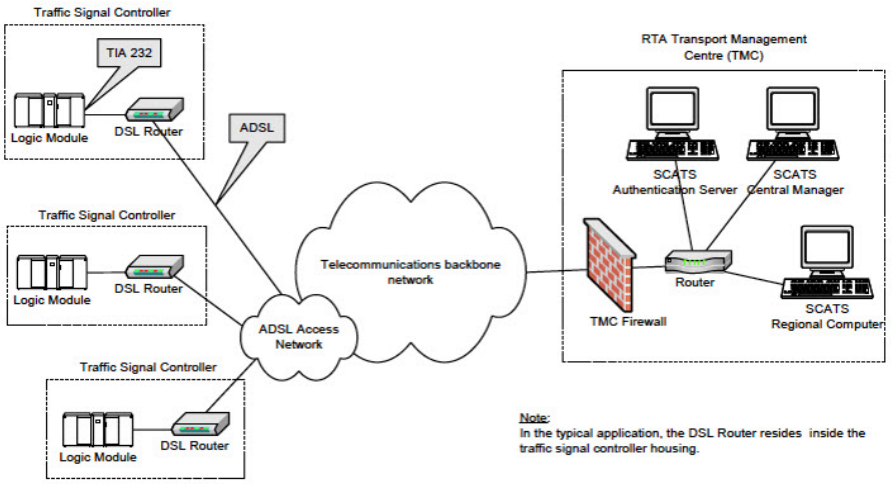
HDLC Support for SCATS Overview

The Sydney Coordinated Adaptive Traffic System (SCATS), is an intelligent transportation system that manages the dynamic (on-line, real-time) timing of signal phases at traffic signals, meaning that it tries to find the best phasing (i.e. cycle times, phase splits and offsets) for a traffic situation (for individual intersections as well as for the whole network). SCATS is based on the automatic plan selection from a library in response to the data derived from loop detectors or other road traffic sensors. SCATS uses sensors at each traffic signal to detect vehicle presence in each lane and pedestrians waiting to cross at the local site. The vehicle sensors are generally inductive loops installed within the road pavement. The pedestrian sensors are usually push buttons. Various other types of sensors can be used for vehicle presence detection, provided that a similar and consistent output is achieved. Information collected from the vehicle sensors allows SCATS to calculate and adapt the timing of traffic signals in the network.

High-Level Data Link Control (HDLC) is a group of data link (Layer 2) protocols used to transmit synchronous data packets between point-to-point nodes. Data is organized into addressable frames. This format has been used for other multipoint-to-multipoint protocols, and inspired the HDLC-like framing protocol described in RFC 1662. HDLC uses a zero-insertion/deletion process (bit stuffing) to ensure that the bit pattern of the delimiter flag does not occur in the fields between flags. The HDLC frame is synchronous and therefore relies on the physical layer (Layer 1) to clock and synchronize frame transmission and reception.

This feature is being developed as an IOx app which integrates with the existing virtualization layers available in IOS XE based IoT routers. The intended application is to have a SCATS controller connected to the router via serial cable. The SCATs protocol the app will follow is documented in specification TSI-SP-068.

The following figure is an example of a typical SCATS traffic control network application:



In the above figure, an IR1101 plays the role of the DSL Router to which the Traffic Signal Controller (TSC) is connected via a serial interface. Upon connection to the TSC, the router obtains a Site ID from the controller, which it will then forward to the SCATs Authentication Server. The authentication servers will be provided to the IOx app through a JSON file including IP and port and there can be up to three authentication servers that the IOx app can cycle through.

Once the Authentication Server has received the Site ID, it will reply to the router with the corresponding SCATs regional computer IP and port that matches that Site ID. All further communication is then done transparently from TSC to Regional Computer.

The router will use two modes to communicate with the TSC (HDLC and non-HDLC). There are four available serial configurations, and the user can select which configurations will be used by enabling or disabling them through a second JSON file provided to the app.

Since this is an IOx app, the feature can be disabled by stopping, deactivating, or uninstalling the app. The application will mainly be deployed using Local Manager. App size is about 50 MB, CPU is 400 units and memory is 128 MB.

Configure IOx Application

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.

Important The SCATs IOX application does not configure and enable a VPN for the connections. If a VPN is necessary for operations, please configure the VPN through IOS XE on the IR1101 outside of the application.

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.

Enable IOx

Perform the following steps:

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:** `username admin privilege 15 password 0 admin`	Establishes a username-based authentication system and privilege level for the user. The username privilege level must be configured as 15.
Step 7	**end** **Example:** `Device(config-if)#end`	Exits interface configuration mode and returns to privileged EXEC mode.

Configure 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.

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** *type number* **Example:** `Device(config)#interface virtualportgroup 0`	Configures an interface and enters interface configuration mode.
Step 4	**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 5	**end** **Example:** `Device(config-if)#end`	Exits interface configuration mode and returns to privileged EXEC mode.

Configure Serial Port for IOx Communication

Use the following steps to configure the serial 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	**interface async** *number* **Example:** `Device(config)#interface async 0/3/0`	Configures an async interface and enters interface configuration mode.

	Command or Action	Purpose
Step 4	**encapsulation relay-line** **Example:** `Device(config-if)#encapsulation relay-line`	Configure the async interface as a relay-line.
Step 5	**end** **Example:** `Device(config-if)#end`	Exits interface configuration mode and returns to privileged EXEC mode.
Step 6	**relay-line** *slot/subslot/port for modems* **Example:** `Device(config)#relay-line 0/0/1 0/3/0`	Configure the relay line between async interface and IOx app.
Step 7	**end** **Example:** `Device(config)#end`	Exits interface configuration mode and returns to privileged EXEC mode.

Deploy SCATs Application

There are two methods to deploy the SCATs application on the IR1101. Either through the Local Manager (Graphical UI) or through IOS-XE (On-device CLI).

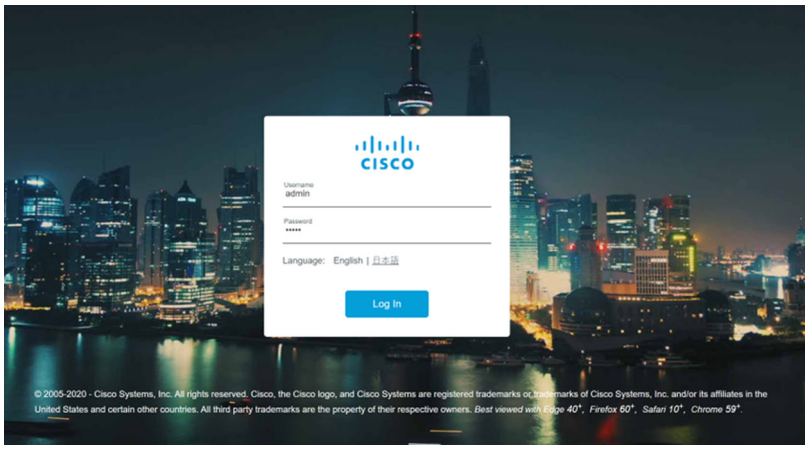
Deploy SCATs Application via Local Manager

If you have gone through the procedure to enable the webserver and to add a user, you should be able to access the IR1101 web interface using the SVI IP-address. using https://<svi ip>/ (eg: https://192.168.0.30/) and then log in using the user created earlier.

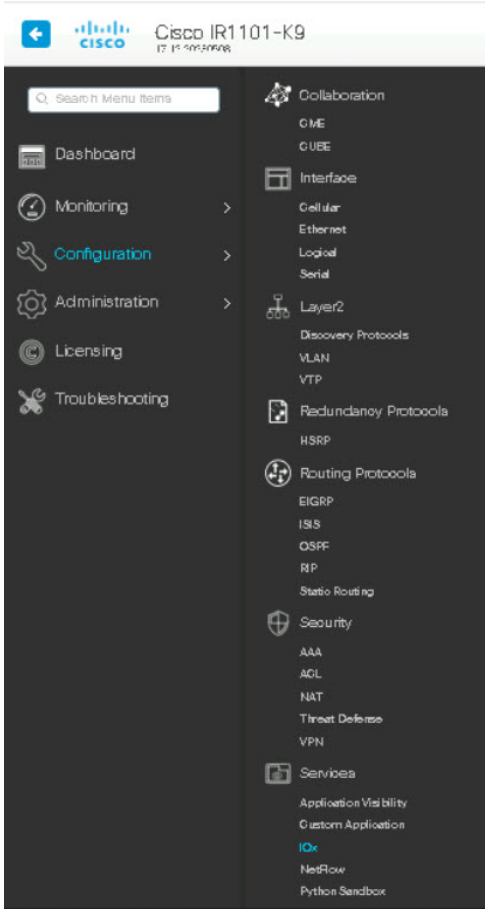
Procedure

Step 1 **https://**<*svi ip*>/

The WebUI login appears:



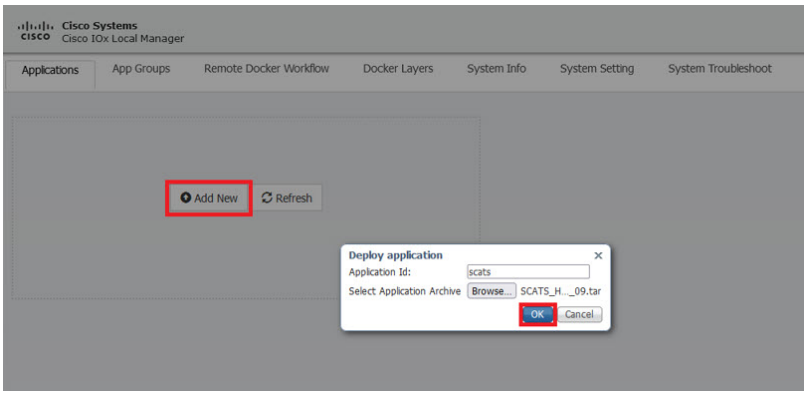
Step 2 Navigate to the IOx page through **Configuration > Services > IOx**. See the following image:



Step 3 Use the same user credentials to enter Cisco IOx Local Manager. (For direct access, use the following URL: https://<svi ip>/iox/login)

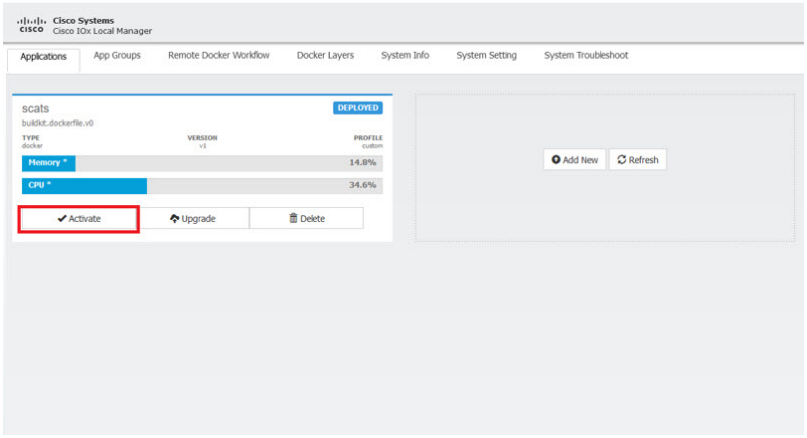


Step 4 Deploy the application by clicking **Add New**. Assign a name to the Application Id, and select the SCATs application package for the Application Archive.

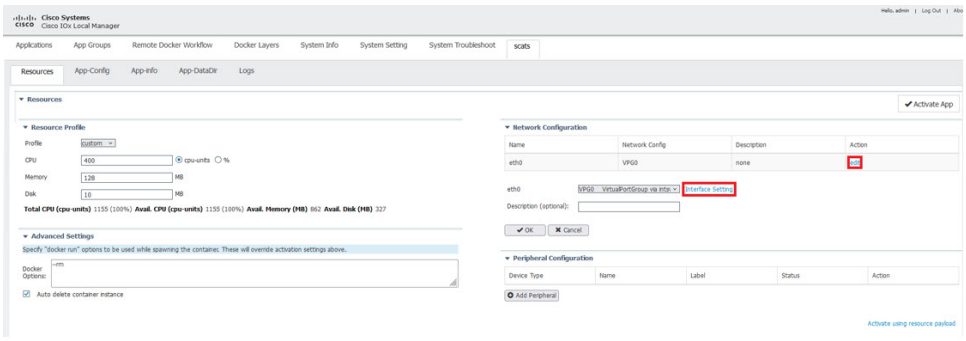


Step 5 After selecting **OK**, the application will be uploaded and installed into the IR1101.

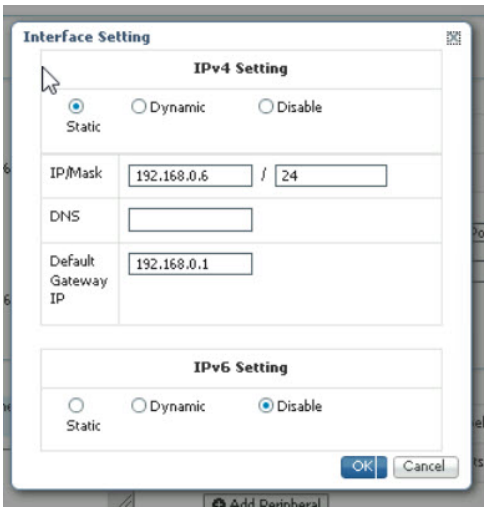
Step 6 Click on **Activate**.



Step 7 Under the **Network Configuration** section, the VirtualPortGroup0 configuration from above can be seen. Click on **edit** and then click **Interface Setting**.

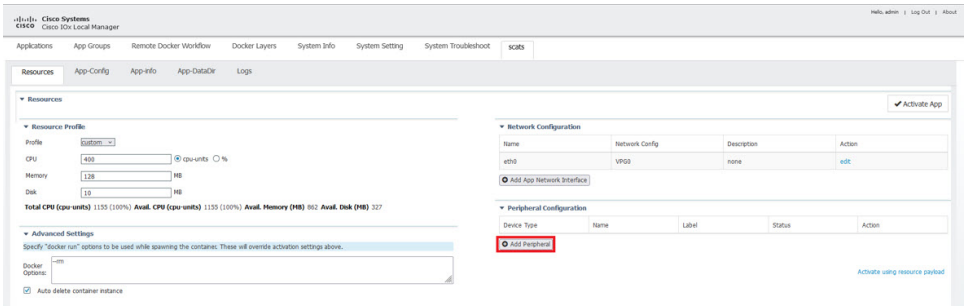


Step 8 Configure the IP addresses for the interface.

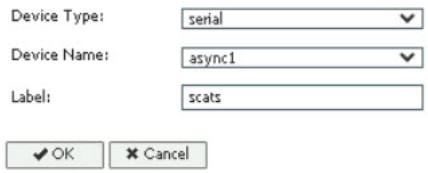


Step 9 Click **OK** on both windows to finalize the network configuration.

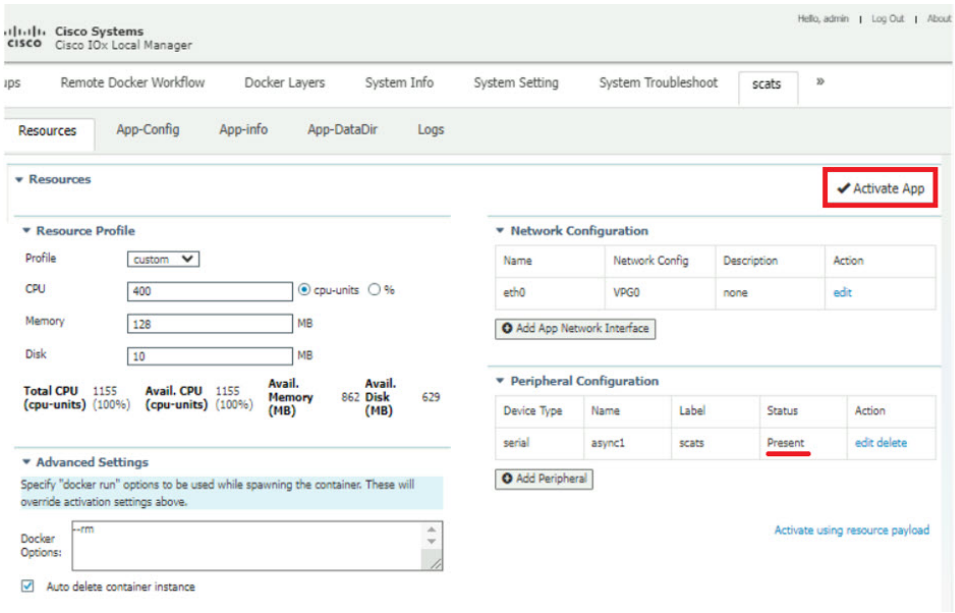
Step 10 Under the **Peripheral Configuration** section, select **serial** for the Device Type.



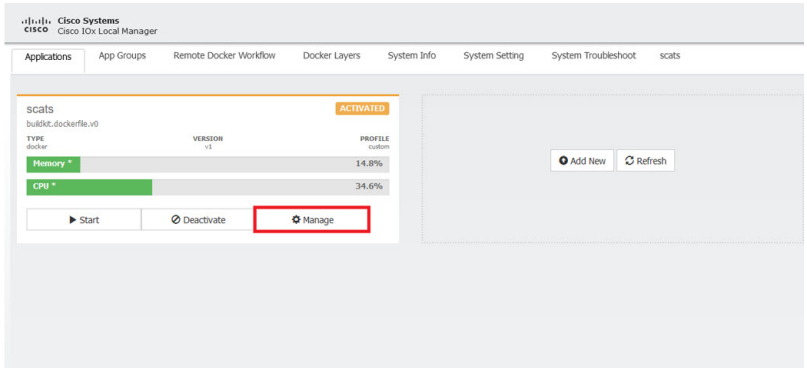
Step 11 For Device Name, select the **async** interface the serial relay line was mapped to. Label the peripheral and click **OK**



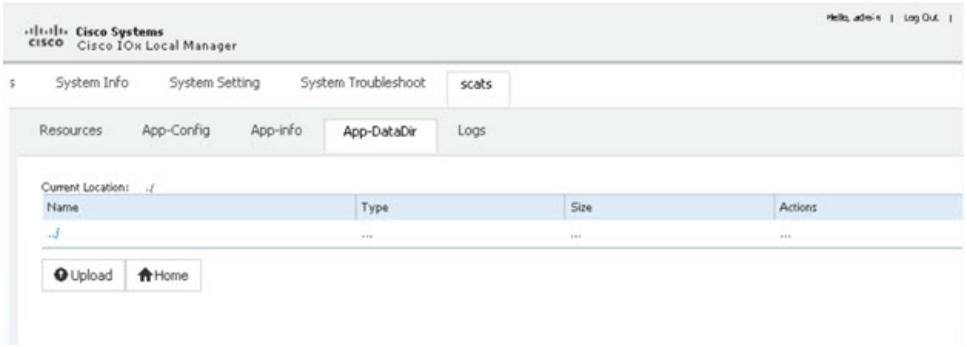
Step 12 The **Status** should say **Present** for the peripheral. Click on **Activate App** in the top right corner, and then select **Applications** in the top bar to return to the main page.



Step 13 The application will now be activated. Click **Manage** to be brought back to the Resources Page.



Step 14 From the **Resources** tab, click **App-DataDir**.



SCATs requires two files to operate, authserver.json and serialconfig.json, to notify the application of the available authentication servers and which of the four serial configurations for SCATs to enable.

```
Example of authserver.json file (1-3 auth servers allowed)
{
    "auth_servers":[
        {"ip":"10.0.1.13", "port":2012},
        {"ip":"10.0.1.1", "port":2012}
    ]
}

Example of serialconfig.json file
{
    "serial_configurations":[
        {"serial_config":"enabled"},
        {"serial_config":"disabled"},
        {"serial_config":"disabled"},
        {"serial_config":"disabled"}
    ]
}
```

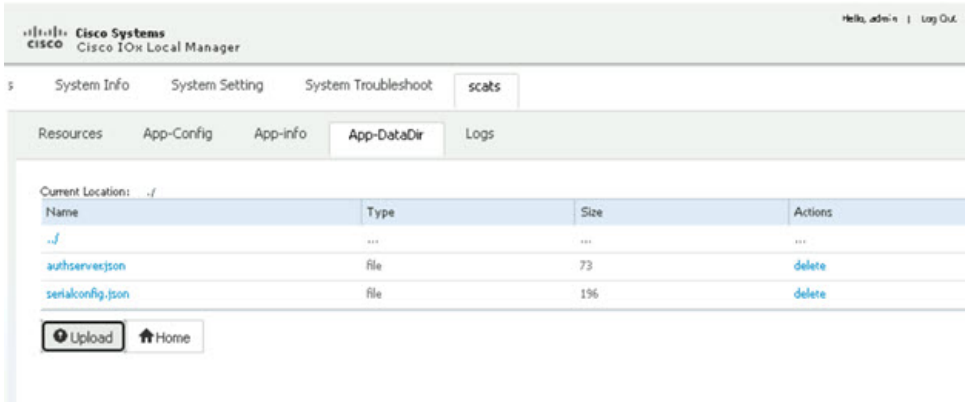
Step 15 Click on **Upload** and choose the files to be uploaded into the App-DataDir.

Note
The paths must be authserver.json and serialconfig.json.

Step 16 Click **OK** to select.

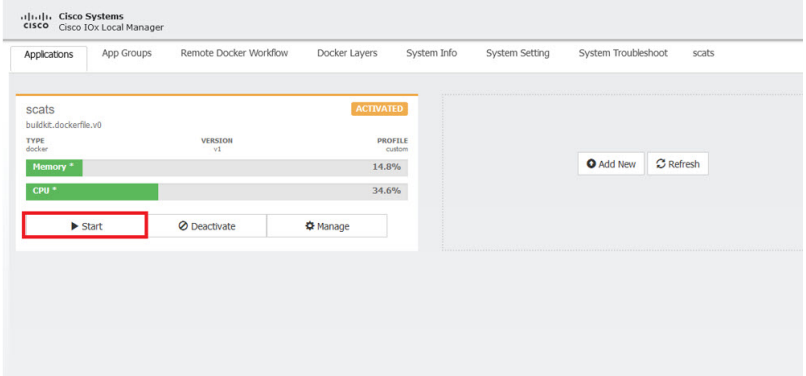


Step 17 Verify the two json files are present and then click **Upload**.



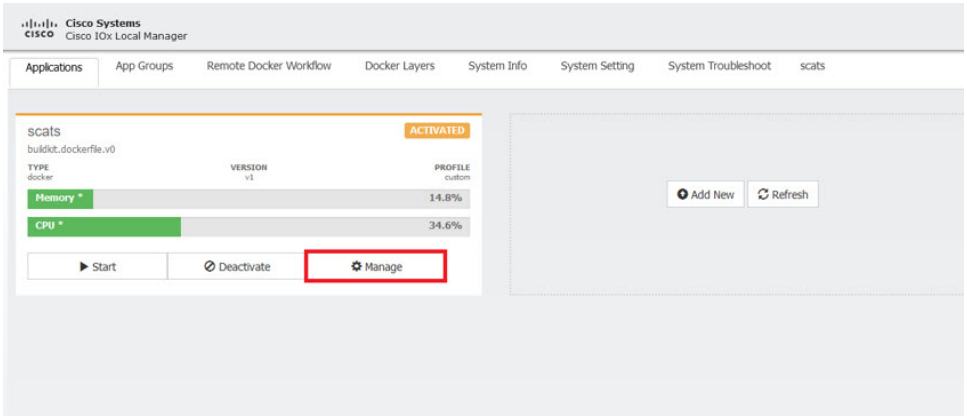
Step 18 Select **Applications** to return to the main application page. Click on **Start** to start the application. It will show as running now.

Step 19 Click on **Start** to start the application.

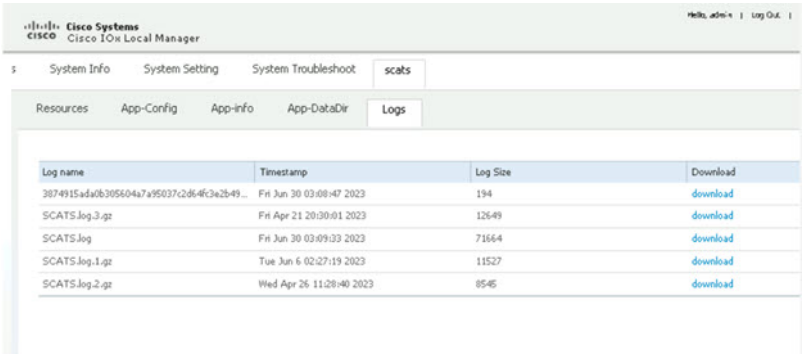


The status shows the application is activated, and you should see Memory and CPU details.

Step 20 To troubleshoot any issues, click on **Manage** and then click on the **Logs** tab.



Step 21 Logs from the application will be stored under SCATS.log* and can be downloaded from the Local Manager.



To stop and delete the app, click **Stop**, then **Deactivate** and **Delete**.

Deploy SCATs Application Using the IOS-XE CLI

Use the following steps:

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** *app-name* **Example:** `Device(config)#app-hosting appid scats`	Configures the SCATS application and enters the application configuration mode.
Step 4	**app-vnic** *gateway-number* **virtualportgroup** *number* **guest-interface** *number* **Example:** `Device(config-app-hosting)#app-vnic gateway0 virtualportgroup 0 guest-interface 0`	Configures the application interface and the gateway of the application.
Step 5	**guest-ipaddress** *ip-address* **netmask** *mask* **Example:** `Device(config-app-hosting-gateway0)#guest-ipaddress 192.168.0.6 netmask 255.255.255.0`	Configures the application Ethernet interface ip address.
Step 6	**app-default-gateway** *ip-address* **guest-interface** *number* **Example:** `Device(config-app-hosting-gateway0)#app-default-gateway 192.168.0.1 guest-interface 0`	Configures the application default gateway ip address.
Step 7	**app-hosting docker** **Example:** `Device(config-app-hosting)#app-hosting docker`	Enter the configuration mode for docker options.
Step 8	**run-opts** *option-number* "*-- device host-serial:container-serial* **Example:** `Device (config-app-hosting-docker)#run-opts 1 "--device /dev/ttySerial1:/dev/ttySerial1"`	Match the async interface to the container interface. **Note** The serial port must match what the relay line was set to. In the example, async 0/3/0 was set to async 1, so the corresponding serial is /dev/ttySerial1.
Step 9	**end** **Example:** `Device(config-app-hosting-docker)#end`	Exits docker options configuration mode and returns to privileged EXEC configuration mode.

	Command or Action	Purpose
Step 10	**app-hosting install appid** *application-name* **package** *package-path* **Example:** `Device#app-hosting install appid scats package flash:SCATS_HDLC_signed_05_09.tar`	Installs the SCATS app from the specified location. The app can be installed from any local storage location such as, flash, bootflash, and usbflash0.
Step 11	**app-hosting activate appid** *application-name* **Example:** `Device#app-hosting activate appid scats`	Activates the SCATS application. Activates the SCATS application. This command validates all application resource requests, and if all resources are available the application is activated; if not, the activation fails.
Step 12	**app-hosting data appid** *application-name* **copy** *authserver.json-path* **Example:** `Device#app-hosting data appid scats copy flash:authserver.json`	Copy the authserver.json file into the IOx App-Data dir.
Step 13	**app-hosting data appid** *application-name* **copy** *serialconfig.json-path* **Example:** `Device# app-hosting data appid scats copy flash:serialconfig.json`	Copy the serialconfig.json file into the IOx App-Data dir.
Step 14	**app-hosting start appid** *application-name* **Example:** `Device#app-hosting start appid scats`	Starts the SCATs application. Application start-up scripts are activated.

Troubleshooting

To troubleshoot the app, perform the following:

Start a session within the IOx app, for example:

app-hosting connect appid *application-name* **session**

For example:

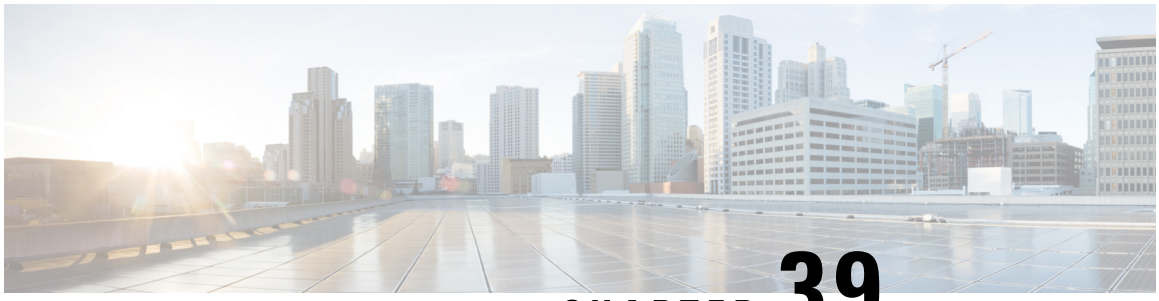
`Device#app-hosting connect appid scats session`

Logs can be viewed in /iox_data/logs under SCATS.log*.

Stop and Delete the Application

To stop and delete the app, do the following steps:

Step	Command	Purpose
1	**app-hosting stop appid** *application-name* `Device#app-hosting stop appid scats`	Stops the application.
2	**app-hosting deactivate appid** *application-name* `Device#app-hosting deactivate appid scats`	Deactivates all resources allocated for the application.
3	**app-hosting uninstall appid** *application-name* `Device#app-hosting uninstall appid scats`	Uninstalls the application. Uninstalls all packaging and images stored. All changes and updates to the application are also removed.



CHAPTER 39

Ethernet VPN Virtual Extensible LAN Over Generic Routing Encapsulation

This chapter contains the following sections:

- Overview, on page 429
- Configuration Examples, on page 430
- Configuration Steps, on page 432
- Troubleshooting, on page 434
- Additional Resources, on page 436

Overview

Ethernet VPN (EVPN) is a standards-based BGP distributed control plane for Network Virtualization Overlay (NVO), that provides Layer 2 (bridging) and Layer 3 (routing) connectivity over IP or IP/MPLS underlay networks.

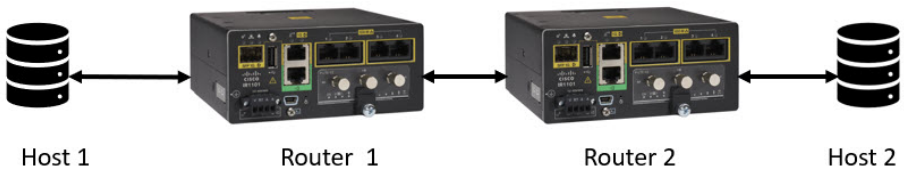
Virtual Extensible LAN (VxLAN) is a solution to support a flexible, large-scale multitenant environment over a shared common physical infrastructure. The transport protocol over the physical data center network is IP plus UDP.

Generic Routing Encapsulation (GRE) provides a virtual point-to-point private connection and encapsulates and forwards packets over an IP-based network.

This solution provides the customer the ability to extend an L2 broadcast domain over an L3 IP network. The GRE tunnel facilitates connection of disjoint L3 network subnets over which VXLAN packets can be transferred.

The following shows an example topology:

Figure 60: Topology



Configuration Examples

The following configuration supports the topology illustrated in the Overview, on page 429.

Host 1

```
interface GigabitEthernet1/7
 switchport access vlan 21
 switchport mode access
!
interface Vlan21
 ip address 21.21.21.1 255.255.255.0
```

Router 1

```
l2vpn evpn
 replication-type ingress
!
l2vpn evpn instance 21 vlan-based
 encapsulation vxlan
 replication-type ingress
 default-gateway advertise enable
!

bridge-domain 21
 member Vlan21 service-instance 21
 member evpn-instance 21 vni 30000
!
!
interface Loopback0
 ip address 100.100.100.1 255.255.255.255
!
interface Tunnel100
 ip address 102.102.102.1 255.255.255.252
 ip pim sparse-mode
 mpls ip
 tunnel source 192.168.5.100
 tunnel destination 192.168.5.200
!
interface GigabitEthernet0/0/0
 ip address 192.168.5.100 255.255.255.0
!
interface FastEthernet0/0/1
 switchport access vlan 21
 switchport mode access
!
interface FastEthernet0/0/2
!
!
interface Vlan21
 no ip address
 service instance 21 ethernet
  encapsulation dot1q 21
 !

!
interface nve1
 no ip address
 source-interface Loopback0
 host-reachability protocol bgp
 member vni 30000 ingress-replication
```

```
!
router ospf 1
 router-id 100.100.100.1
 network 100.100.100.1 0.0.0.0 area 0
 network 102.102.102.0 0.0.0.3 area 0
!
router bgp 1
 bgp router-id 100.100.100.1
 bgp log-neighbor-changes
 neighbor 102.102.102.2 remote-as 1
 !
 address-family ipv4
  redistribute connected
  neighbor 102.102.102.2 activate
  neighbor 102.102.102.2 send-community both
 exit-address-family
 !
 address-family vpnv4
  import l2vpn evpn
  neighbor 102.102.102.2 activate
  neighbor 102.102.102.2 send-community extended
 exit-address-family
 !
 address-family l2vpn evpn
  neighbor 102.102.102.2 activate
  neighbor 102.102.102.2 send-community both
 exit-address-family
!
ip pim rp-address 100.100.100.2
```

Router 2

```
l2vpn evpn
 replication-type ingress
!
l2vpn evpn instance 21 vlan-based
 encapsulation vxlan
 replication-type ingress
 default-gateway advertise enable
!
!

bridge-domain 21
 member Vlan21 service-instance 21
 member evpn-instance 21 vni 30000
!
!
interface Loopback0
 ip address 100.100.100.2 255.255.255.255
!
interface Tunnel100
 ip address 102.102.102.2 255.255.255.252
 ip pim sparse-mode
 mpls ip
 tunnel source 192.168.5.200
 tunnel destination 192.168.5.100
!
interface GigabitEthernet0/0/0
 ip address 192.168.5.200 255.255.255.0
 negotiation auto
!
interface GigabitEthernet0/1/0
 switchport access vlan 21
 switchport mode access
```

```
!
!
interface Vlan21
 no ip address
 service instance 21 ethernet
  encapsulation dot1q 21
 !

!
interface nve1
 no ip address
 source-interface Loopback0
 host-reachability protocol bgp
 member vni 30000 ingress-replication
!
router ospf 1
 router-id 100.100.100.2
 network 100.100.100.2 0.0.0.0 area 0
 network 102.102.102.0 0.0.0.3 area 0
!
router bgp 1
 bgp router-id 100.100.100.2
 bgp log-neighbor-changes
 neighbor 102.102.102.1 remote-as 1
 !
 address-family ipv4
  redistribute connected
  neighbor 102.102.102.1 activate
  neighbor 102.102.102.1 send-community both
 exit-address-family
 !
 address-family vpnv4
  import l2vpn evpn
  neighbor 102.102.102.1 activate
  neighbor 102.102.102.1 send-community extended
 exit-address-family
 !
 address-family l2vpn evpn
  neighbor 102.102.102.1 activate
  neighbor 102.102.102.1 send-community both
 exit-address-family
!
ip forward-protocol nd
ip pim rp-address 100.100.100.2
```

Host 2

```
interface GigabitEthernet1/7
 switchport access vlan 21
 switchport mode access
!
interface Vlan21
 ip address 21.21.21.2 255.255.255.0
```

Configuration Steps

The following steps configure Router 1:

1. Create the EVPN and EVPN instance:

```
l2vpn evpn
replication-type ingress
```

```
!
l2vpn evpn instance 21 vlan-based
encapsulation vxlan
replication-type ingress
default-gateway advertise enable
```

2. Add a port to VLAN 21:

```
interface FastEthernet0/0/1
switchport access vlan 21
switchport mode access
```

3. Configure BDI 21 on Vlan 21:

```
interface Vlan21
no ip address
service instance 21 ethernet
  encapsulation dot1q 21
```

4. Assign IP to Loopback interface:

```
interface Loopback0
ip address 100.100.100.1 255.255.255.255
```

5. Configure IP on WAN interface:

```
interface GigabitEthernet0/0/0
ip address 192.168.5.100 255.255.255.0
```

6. Configure a GRE Tunnel with WAN interface IP:

```
interface Tunnel100
ip address 102.102.102.1 255.255.255.252
ip pim sparse-mode
mpls ip
tunnel source 192.168.5.100
tunnel destination 192.168.5.200
```

7. Configure VXLAN:

```
interface nve1
no ip address
source-interface Loopback0
host-reachability protocol bgp
member vni 30000 ingress-replication
```

8. Apply the EVPN and VxLAN instance on BDI (Bridge domain interface):

```
bridge-domain 21
 member Vlan21 service-instance 21
member evpn-instance 21 vni 30000
```

9. Configure OSPF and BGP as overlay protocol:

```
router ospf 1
router-id 100.100.100.1
network 100.100.100.1 0.0.0.0 area 0
network 102.102.102.0 0.0.0.3 area 0
!
router bgp 1
bgp router-id 100.100.100.1
bgp log-neighbor-changes
neighbor 102.102.102.2 remote-as 1
!
address-family ipv4
  redistribute connected
  neighbor 102.102.102.2 activate
```

```
  neighbor 102.102.102.2 send-community both
 exit-address-family
 !
 address-family vpnv4
  import l2vpn evpn
  neighbor 102.102.102.2 activate
  neighbor 102.102.102.2 send-community extended
 exit-address-family
 !
 address-family l2vpn evpn
  neighbor 102.102.102.2 activate
  neighbor 102.102.102.2 send-community both
 exit-address-family
```

Perform similar steps to configure Router 2 with the appropriate IP addresses. Then configure IP addresses on both of the hosts for reachability.

Troubleshooting

The following show commands can be used to help troubleshoot your setup.

Router 1

```
Router1#show l2vpn evpn peers vxlan

Interface VNI      Peer-IP                                 Num routes eVNI     UP time
--------- -------- --------------------------------------- ---------- -------- --------
nve1      30000    100.100.100.2                           1          30000    00:00:18


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       30000    L2CP 100.100.100.2    1              30000      UP   N/A  00:00:40


Router1#show l2vpn evpn mac
MAC Address    EVI   BD    ESI                       Ether Tag  Next Hop(s)
-------------- ----- ----- ------------------------ ---------- ---------------
0000.24aa.c926 21    21    0000.0000.0000.0000.0000 0          Vl21:21
0000.24aa.c927 21    21    0000.0000.0000.0000.0000 0          100.100.100.2

Router1#show bgp l2vpn evpn all
BGP table version is 7, local router ID is 100.100.100.1
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: 100.100.100.1:21
 *>   [2][100.100.100.1:21][0][48][000024AAC926][0][*]/20
                       0.0.0.0                                32768 ?
 *>i  [2][100.100.100.1:21][0][48][000024AAC927][0][*]/20
                       100.100.100.2            0    100      0 ?
Route Distinguisher: 100.100.100.2:21
```

```
 *>i  [2][100.100.100.2:21][0][48][000024AAC927][0][*]/20
                      100.100.100.2            0    100      0 ?
Route Distinguisher: 100.100.100.1:21
 *>   [3][100.100.100.1:21][0][32][100.100.100.1]/17
                      0.0.0.0                            32768 ?
 *>i  [3][100.100.100.1:21][0][32][100.100.100.2]/17
                      100.100.100.2            0    100      0 ?
     Network          Next Hop            Metric LocPrf Weight Path
Route Distinguisher: 100.100.100.2:21
 *>i  [3][100.100.100.2:21][0][32][100.100.100.2]/17
                      100.100.100.2            0    100      0 ?
Router1#
```

Router 2

```
Router2#show l2vpn evpn peers vxlan

Interface VNI      Peer-IP                                 Num routes eVNI     UP time
--------- -------- --------------------------------------- ---------- -------- --------
nve1      30000    100.100.100.1                           1          30000    00:00:17

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       30000    L2CP 100.100.100.1    1              30000      UP   N/A  00:00:22


Router2#show l2vpn evpn mac
MAC Address    EVI   BD    ESI                          Ether Tag  Next Hop(s)
-------------- ----- ----- ---------------------------- ---------- ---------------
0000.24aa.c926 21    21    0000.0000.0000.0000.0000 0          100.100.100.1
0000.24aa.c927 21    21    0000.0000.0000.0000.0000 0          Vl21:21

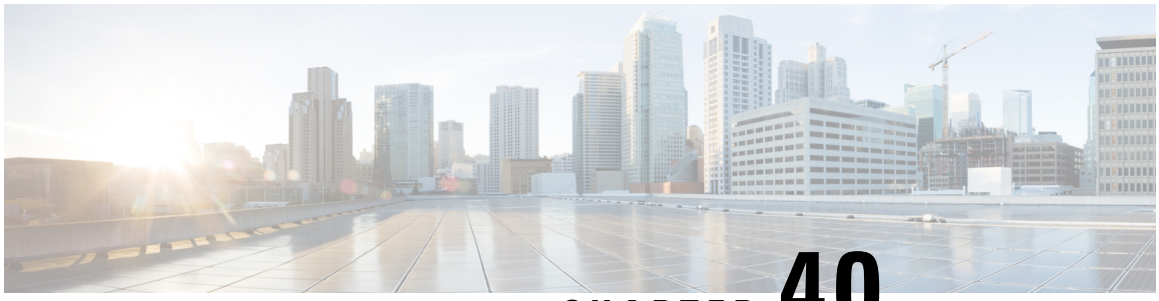
Router2#show bgp l2vpn evpn all
BGP table version is 23, local router ID is 100.100.100.2
Status codes: s suppressed, d damped, h history, * valid, > best, i - internal,
              r RIB-failure, S Stale, m multipath, b backup-path, f RT-Filter,
              x best-external, a additional-path, c 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: 100.100.100.1:21
 *>i  [2][100.100.100.1:21][0][48][000024AAC926][0][*]/20
                      100.100.100.1            0    100      0 ?
Route Distinguisher: 100.100.100.2:21
 *>i  [2][100.100.100.2:21][0][48][000024AAC926][0][*]/20
                      100.100.100.1            0    100      0 ?
 *>   [2][100.100.100.2:21][0][48][000024AAC927][0][*]/20
                      0.0.0.0                            32768 ?
Route Distinguisher: 100.100.100.1:21
 *>i  [3][100.100.100.1:21][0][32][100.100.100.1]/17
                      100.100.100.1            0    100      0 ?
Route Distinguisher: 100.100.100.2:21
 *>i  [3][100.100.100.2:21][0][32][100.100.100.1]/17
                      100.100.100.1            0    100      0 ?
 *>   [3][100.100.100.2:21][0][32][100.100.100.2]/17
                      0.0.0.0                            32768 ?
Router2#
```

Additional Resources

The following are additional sources of information:

- Configure VXLAN
- Configure GRE Tunnels



CHAPTER 40

Cyber Vision Center

- Cyber Vision Support, on page 437
- Installing CVC Sensor using LM GUI, on page 444

Cyber Vision Support

Cisco Cyber Vision Center (CVC) gives more visibility into Industrial IoT networks across Industrial Control Systems (ICS) with real-time monitoring of control and data networks. On IoT IOS-XE platforms beginning with release 17.4, integration of CVC is supported by deploying IOX Cyber Vision sensor. With this sensor deployed on IoT Routers, the platform can forward the traffic from IOX applications to Cyber Vision Center for real-time monitoring and we can forward any captured PCAP files to Vision center from IOX application.

Deployment of Cyber Vision Center (CVC) on IOS-XE platform

Procedure

Step 1 Download Cisco supported Cyber Vision IOX application from the following location:

https://software.cisco.com/download/home/286325414/type/286325316/release/3.1.1?catid=268438162

Select **Cisco Cyber Vision Sensor IOx Application 3.1.1 for IE3400 and IR1101**.

Step 2 Install CVC version 3.1.1 on Virtual Machine or on any Hypervisor. The following location is the download link for different versions of CVC:

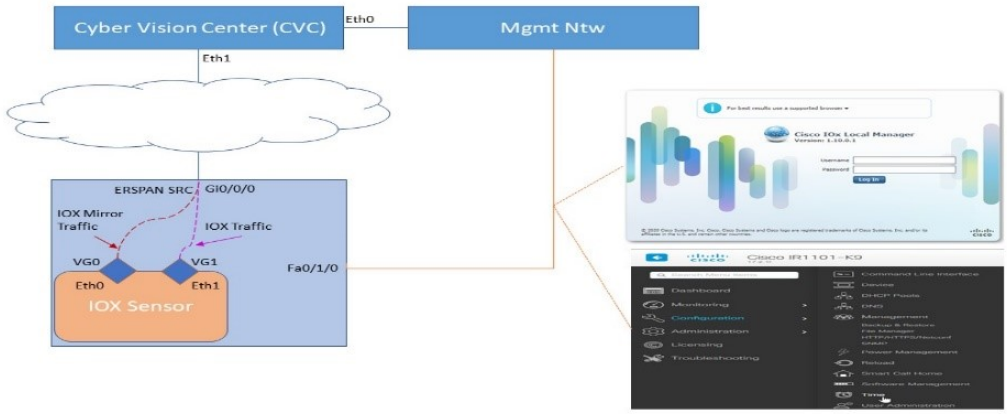
https://software.cisco.com/download/home/286325414/type

Release Notes for Cisco Cyber Vision Release 3.1.1:

https://www.cisco.com/c/dam/en/us/td/docs/security/cyber_vision/Cisco-Cyber-Vision_Release-Note-3-1-1.pdf

Step 3 The CVC sensor requires two VirtualPort Group interfaces. One on the platform where one interface is used for IOX traffic, and the other for mirror traffic which is forwarded to physical, SVI or Tunnel interface which ERSPAN source. Refer to the following illustration:

Figure 61: CVC over L3 interface



Step 4 The CVC Sensor deployment can be installed from either the LMGUI or CLI.

Example Configuration for ERSPAN over L3 configuration along with Virtual Port Groups

Physical and Virtual Port Configuration:

```
interface virtualportgroup 0
ip address 169.254.1.1 255.255.255.252
interface virtualportgroup 1
ip nat inside
ip address 169.254.0.1 255.255.255.252
interface gi0/0/0
ip address 101.0.0.151 255.255.255.0
ip nat outside
no shut
```

ERSPAN Configuration:

```
monitor session 1 type erspan-source
source interface Gi0/0/0
no shutdown
destination
erspan-id 1
mtu 1464
ip address 169.254.1.2
origin ip address 169.254.1.1
```

NAT Configuration with Access-list:

```
ip nat inside source list NAT_ACL interface Gi0/0/0 overload
ip access-list standard NAT_ACL
 10 permit 169.254.0.0 0.0.0.3
```

CLI Installation

To install the app through the CLI, copy the CVC sensor to bootflash, USB, or mSATA. Then install the app using the app-hosting CLI, and provide the docker options before activating the app.

For example:

```
Router(config-if)#iox
Router# app-hosting install app-id <app-id> package {bootflash:/|usbflash0:|msata:}
app-hosting appid  <app-id>
 app-vnic gateway0 virtualportgroup 0 guest-interface 0
  guest-ipaddress 169.254.1.2 netmask 255.255.255.252
 app-vnic gateway1 virtualportgroup 1 guest-interface 1
  guest-ipaddress 169.254.0.2 netmask 255.255.255.252
 app-default-gateway 169.254.0.1 guest-interface 1
 app-resource docker
  run-opts 1 "--rm --tmpfs /tmp:rw,size=128m"
Router# app-hosting {activate|start|stop|deactivate|uninstall} app-id <app-id>
```

LMGUI Installation

Configure the following to reach the LMGUI:

```
iox
ip http server
ip http secure-server
ip http authentication local
Username cisco privilege 15 password cisco
Login URL: http://<Mgmt_IP>/iox/login
```

Additional details can be found in Installing CVC Sensor using LM GUI, on page 444

Register the Router Details

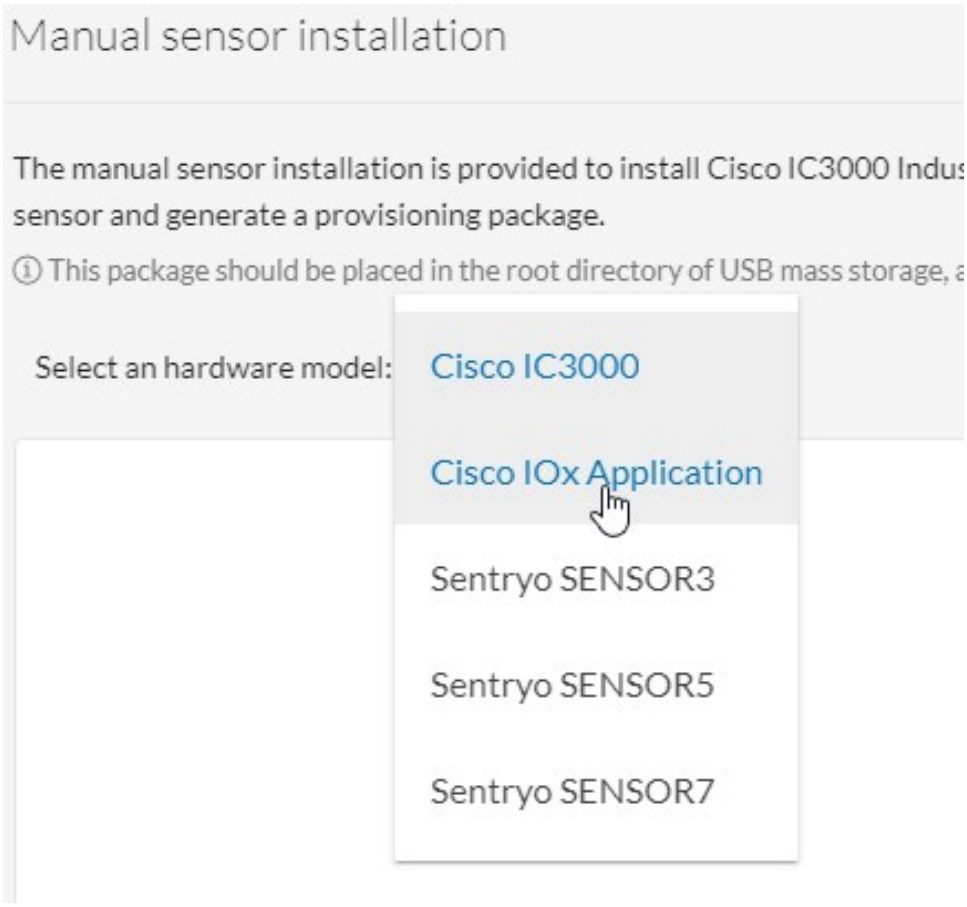
Procedure

Step 1 Register the IOS-XE Router details on CVC by logging in and navigating to:

Admin > Sensors > Install Sensor Manually

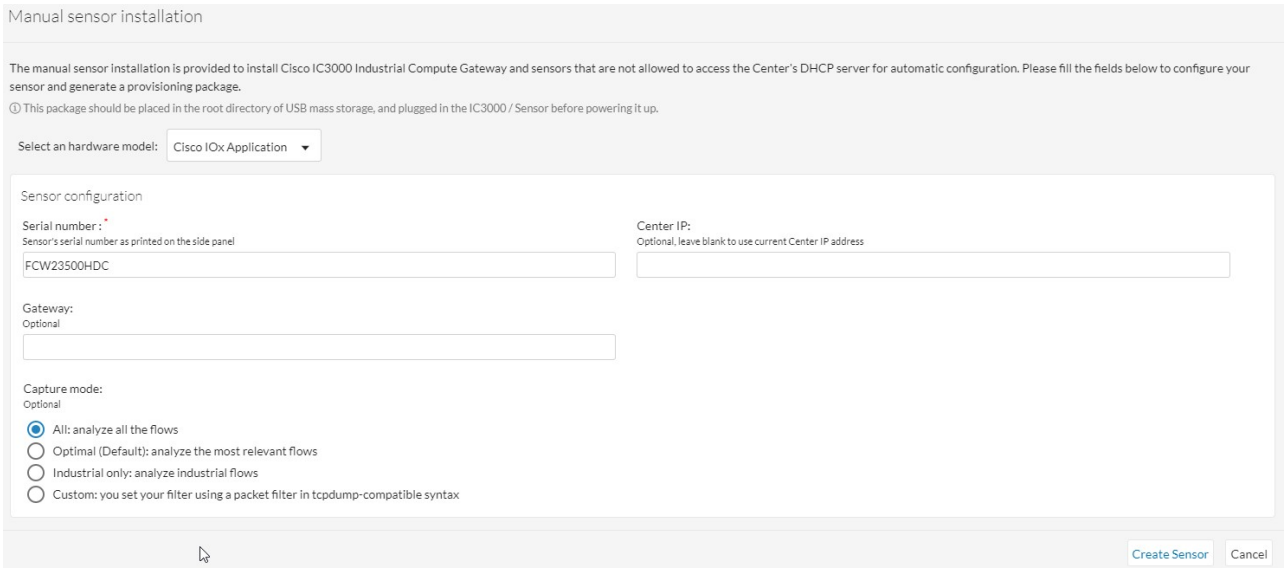
Then click on Cisco IOx Application. Refer to the following:

Figure 62: Sensor install



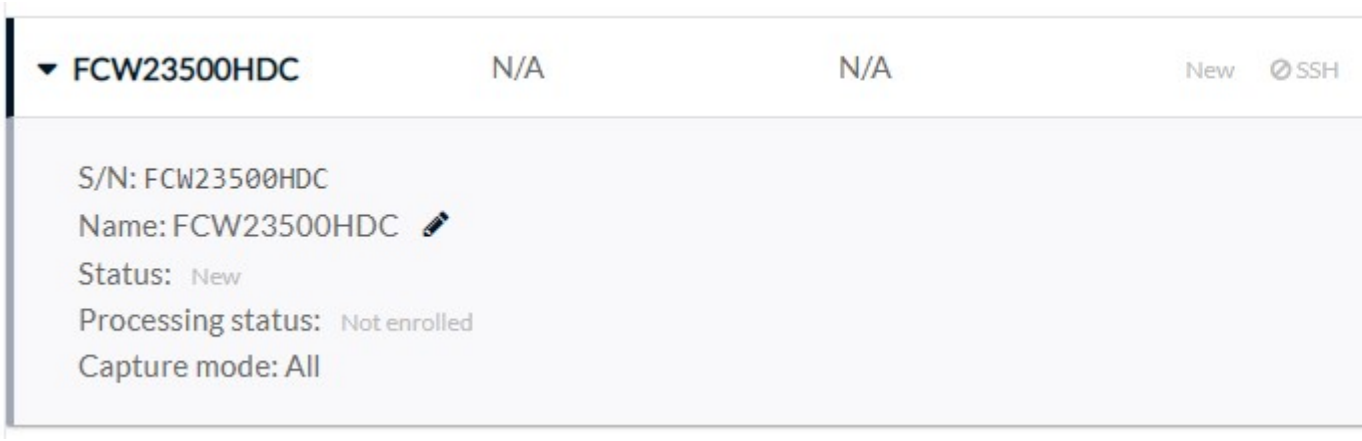
Step 2 Provide the serial number of the Router. It should be an exact match from the output of **show inventory**, and then click on **Create Sensor**. Refer to the following:

Figure 63: Router Serial Number



Step 3 Generate the Provisioning file from CVC by clicking on Get Provisioning File. Refer to the following:

Figure 64: Generate Provisioning File



Step 4 Download the provisioning file to a local directory. The file comes as a zip file with a file name like the following:

Example:

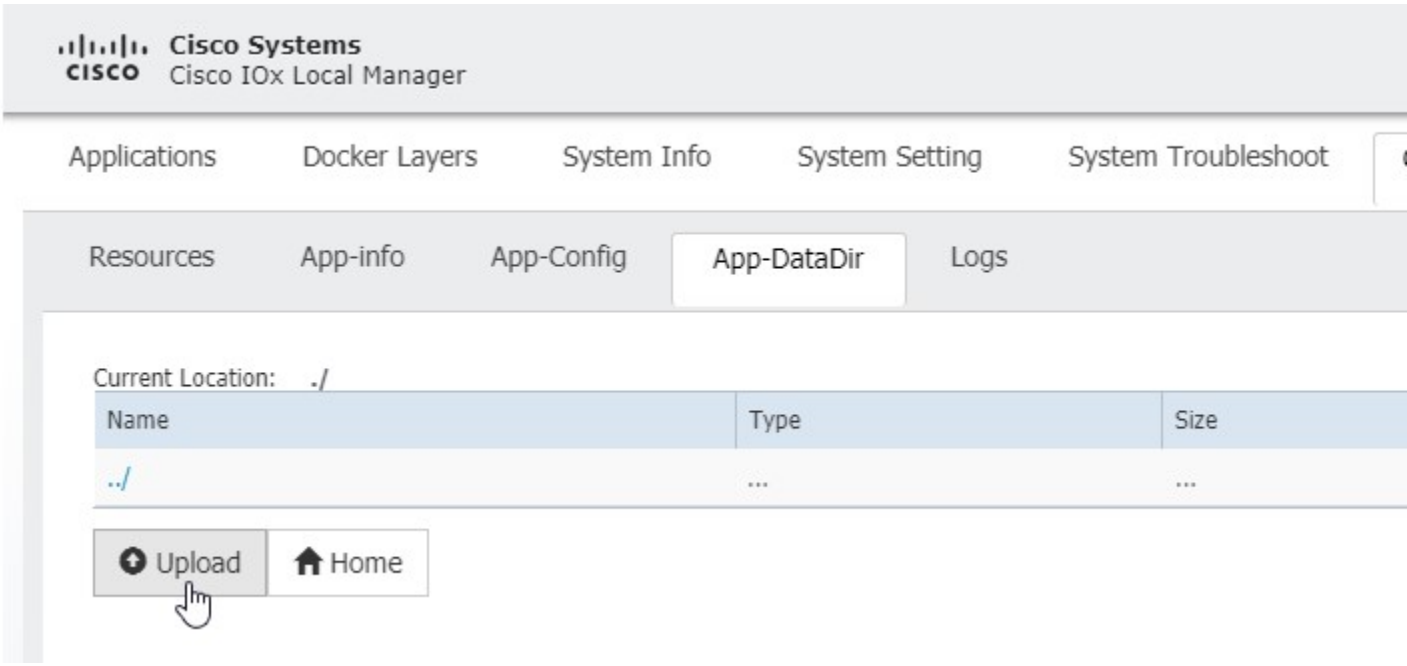
sbs-sensor-config-*<S/N of Router>.zip*

Step 5 Import the Provisioning file to Router through the LM GUI. From the LM GUI Applications, navigate to:

Applications > CVC App (Application Name) > Manage > App-DataDir

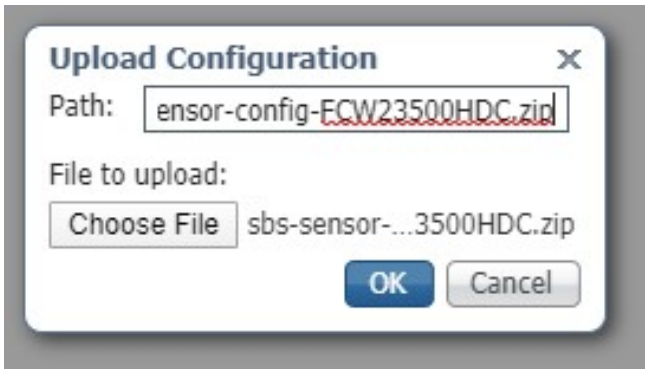
Refer to the following:

Figure 65: Upload Provision File



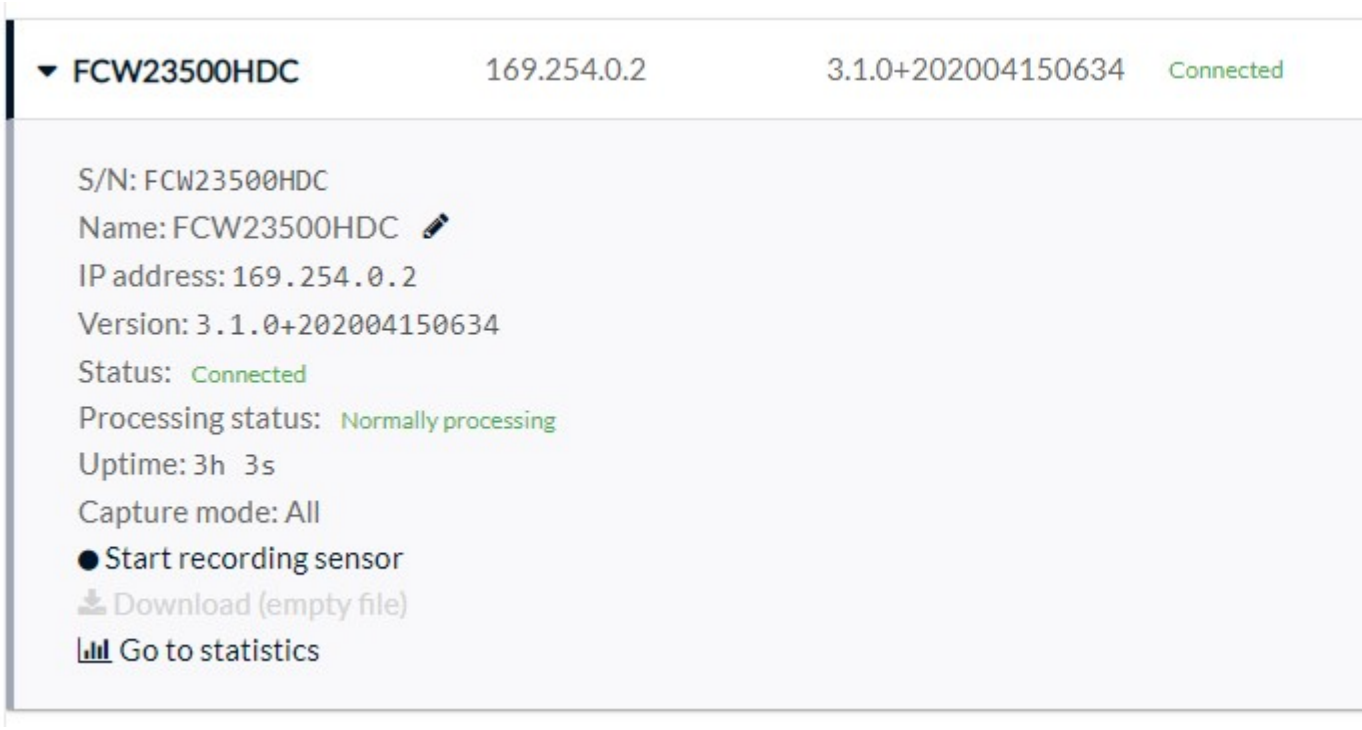
Step 6 Click **Upload**. The Upload Configuration window appears. Upload the downloaded provisioned file from CVC with the same name. Refer to the following:

Figure 66: Upload Configuration



Step 7 Verify the Authentication on CVC. Validate if the installed sensor Status changed to **Connected** or **Waiting for Data**. Refer to the following:

Figure 67: Sensor Status



Capture Live Traffic

Procedure

Step 1 Sync the date and time between CVC and Router. To capture the live traffic there should be exact clock sync between Router and CVC.

Step 2 Simulate IOX Traffic or play captured PCAP files. The CVC Sensor installed on the Router is a docker app. To login to the console of the App, perform the following command:

Example:

```
app-hosting connect app-id <app-name> session
```

Step 3 Upload the PCAP Files to the App from LM-GUI. Navigate to:

Applications > CVC App (Application Name) > Manage >App-Dir

The following commands show how to play the PCAP file:

Example:

```
Router# show app-hosting list
App id    State
```

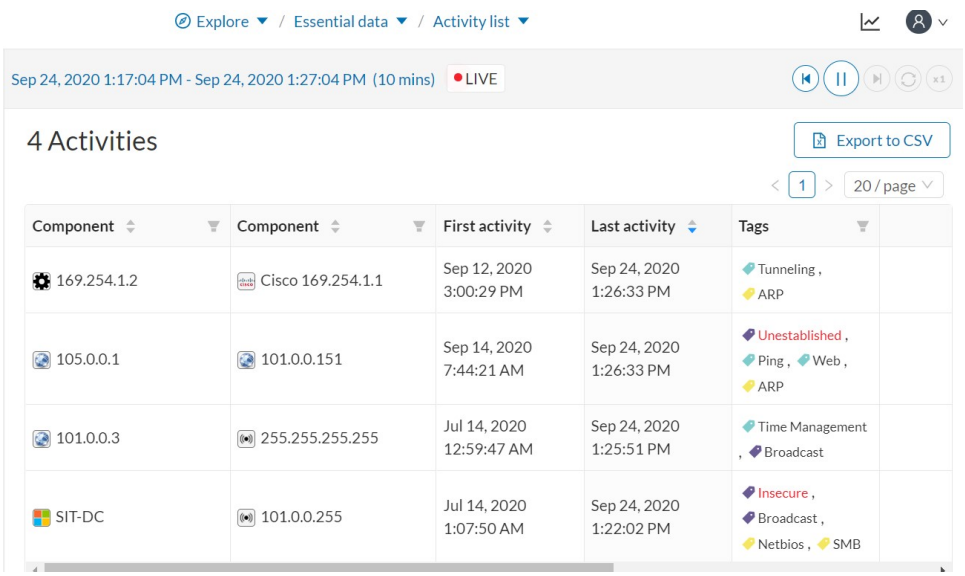
```
------------------------------------------------
CVC Sensor    RUNNING

Router# app-hosting connect appid CVCSensor session
sh-5.0#
*Jul 14 08:45:05.603: %SELINUX-3-MISMATCH: R0/0: audispd: type=AVC msg=audit(15! in/busybox.nosuid"
 dev="overlay" ino=72930 scontext=system_u:system_r: polaris_bexecute_*
sh-5.0# flowctl read-capture-file /iox_data/appdata/tl04
OK
sh-5.0#
```

Step 4 Monitor the traffic on CVC. Navigate to **Explore > Essential Data > Activity List**

Refer to the following:

Figure 68: Activity List

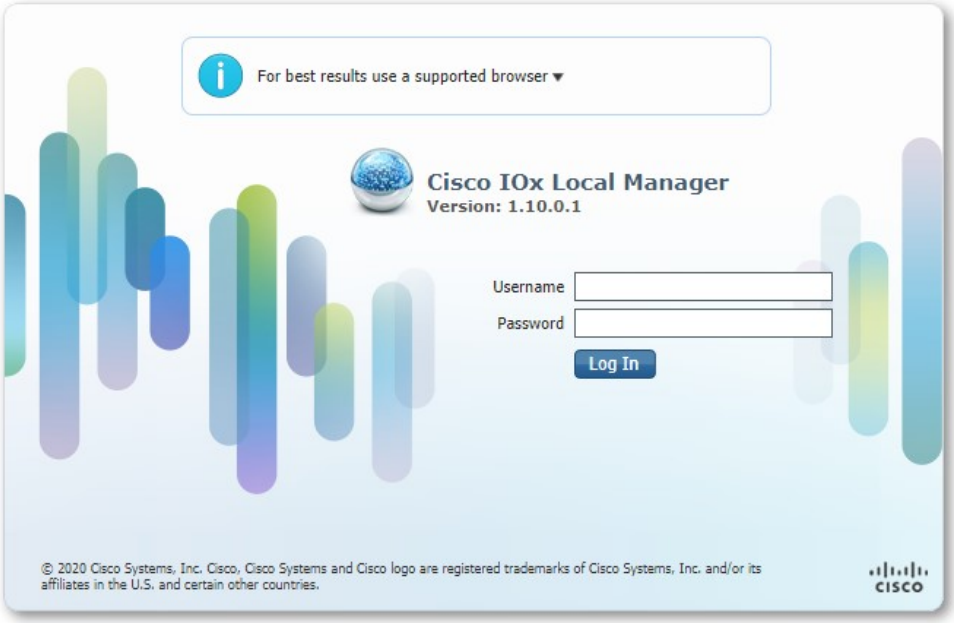


Installing CVC Sensor using LM GUI

Procedure

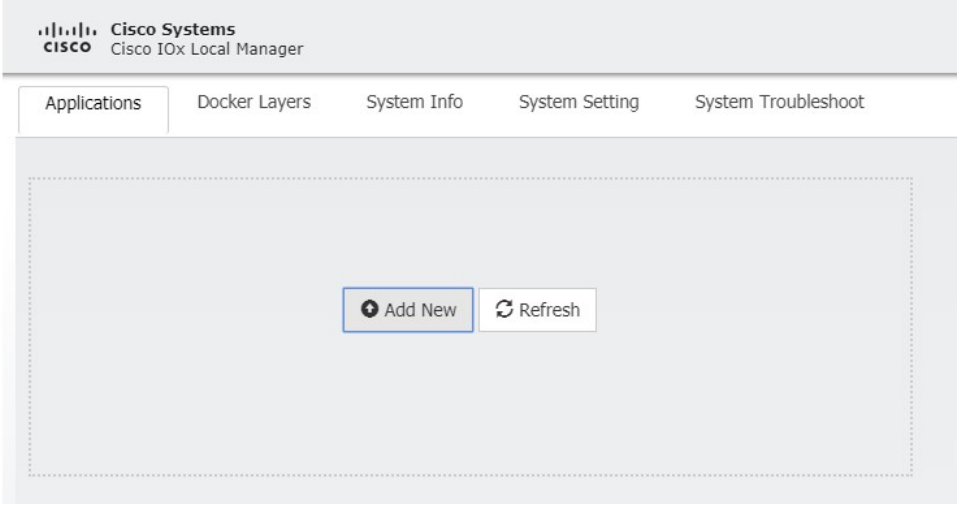
Step 1 Login using user account and password.

Figure 69: Local Manager Login



Step 2 Install the sensor virtual application. Once you are logged in, the following menu will appear:

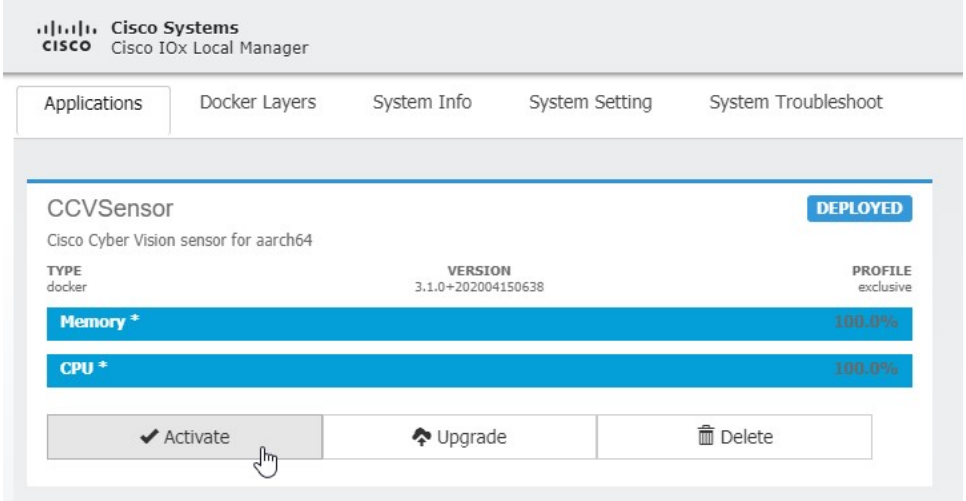
Figure 70: LM GUI Application Install



Step 3 Click on **Add New**. Navigate to the app file, for example, CiscoCyberVision-IOx-aarch64-xxx.tar. Add the name of the app, for example, **CCVSensor**.

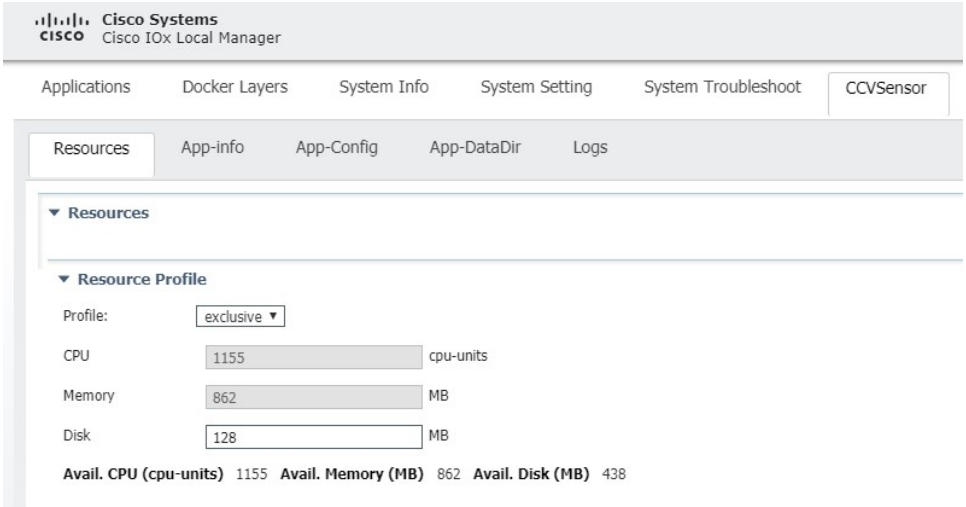
Configure the sensor virtual application. Refer to the following:

Figure 71: CCVSensor Activation



Step 4 Click on **Activate** to launch the configuration of the sensor application. Click on the **CCVSensor** Tab, and click on **Resources**. Refer to the following:

Figure 72: Setup Sensor LM IOXAppDisk



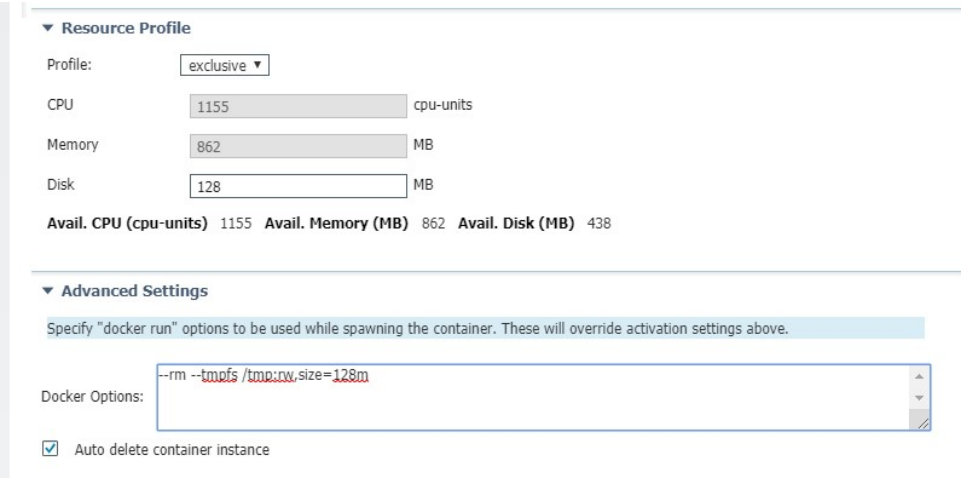
Change the disk size to 128MB.

Note
Do not use more space than that.

Step 5 Navigate to **Advanced Settings**. In advanced options, configure the tmpfs by adding the following in the text area beside Docker Options:

--tmpfs /tmp:rw,size=128m

Figure 73: Advanced Settings



Step 6 Bind interfaces in the container to an interface on the host in the **Network Configuration** section.

What to do next

Move to the next sections Binding eth0 and Binding eth1.

Binding eth0

To configure eth0:

Procedure

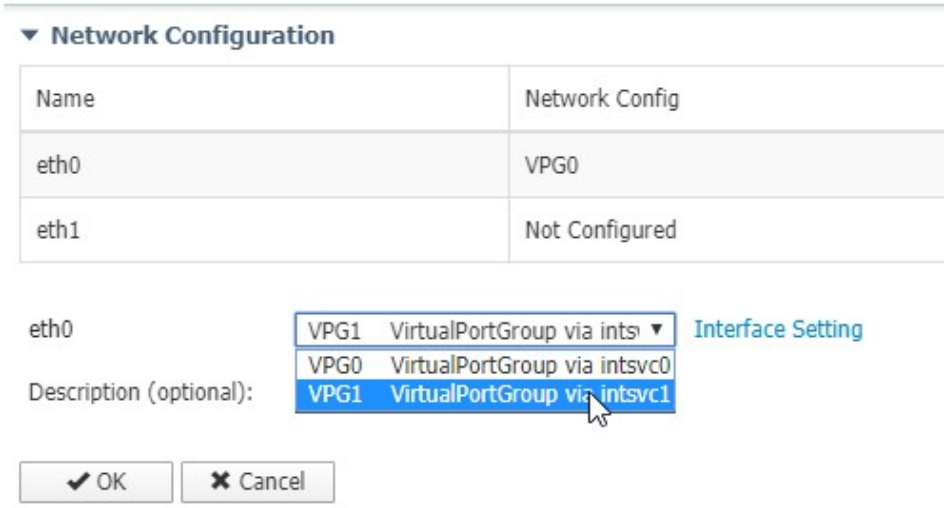
Step 1 Select interface eth0, and then click on **edit**.

Figure 74: eth0



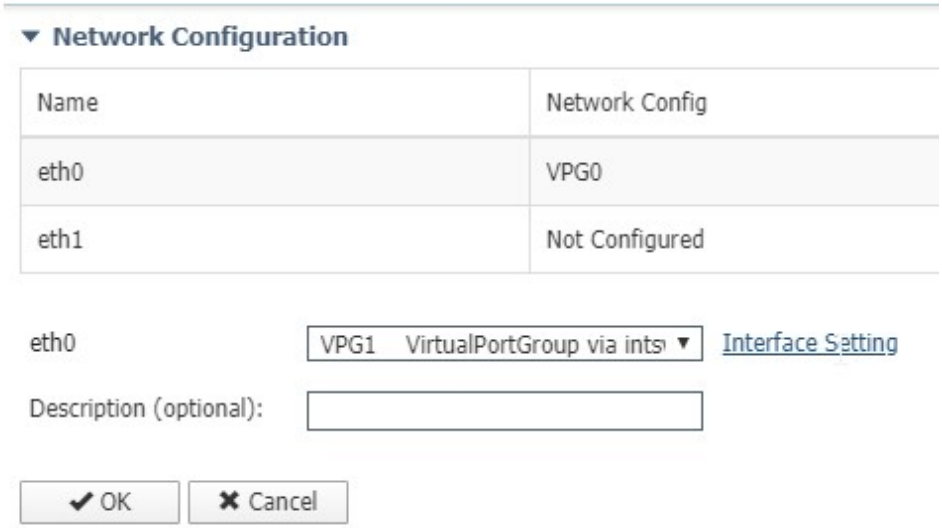
Step 2 Select the Interface **VPG1**.

Figure 75: VPG1



Step 3 Click on **Interface Setting**.

Figure 76: Interface Setting



Step 4 Apply the following configuration:

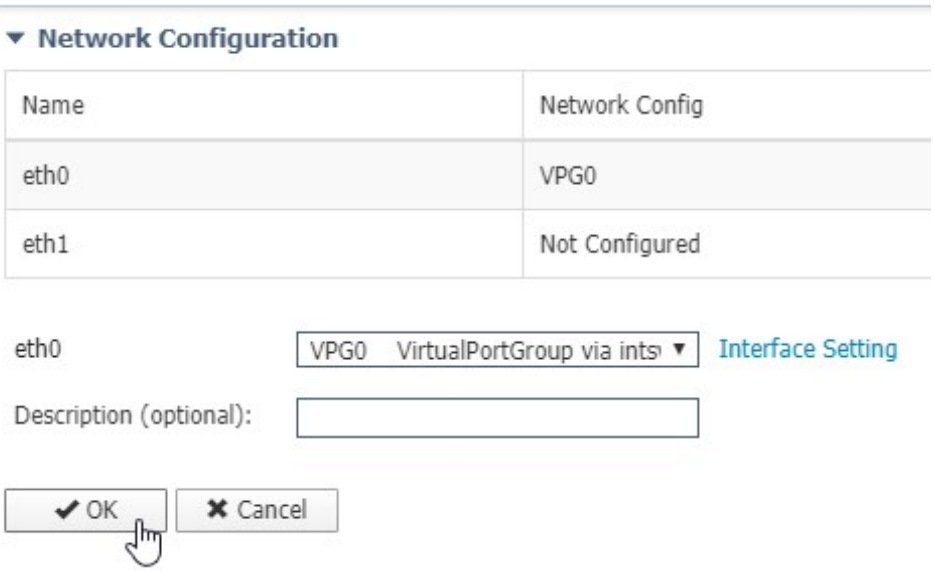
- Choose the **Static** option
- IP/Mask add **169.254.0.2 / 30**
- Default Gateway IP is **169.254.0.1**

Then click on **OK**.

Figure 77: IPv4 Setting

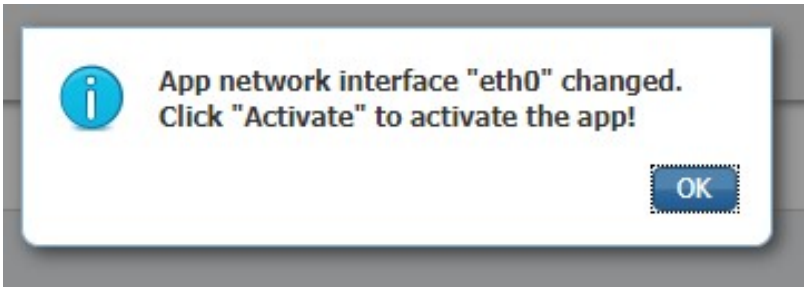


Step 5 Click on **OK** again.



Step 6 The **Activate** window appears. Click on **OK**.

Figure 78: Activate Window



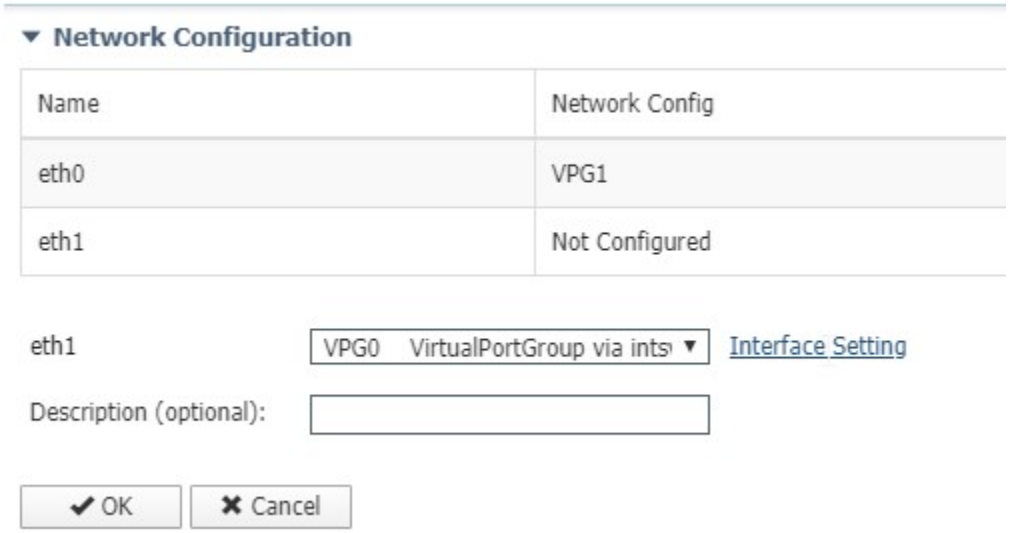
Binding eth1

To configure the eth1 interface:

Procedure

Step 1 Select VPG0.

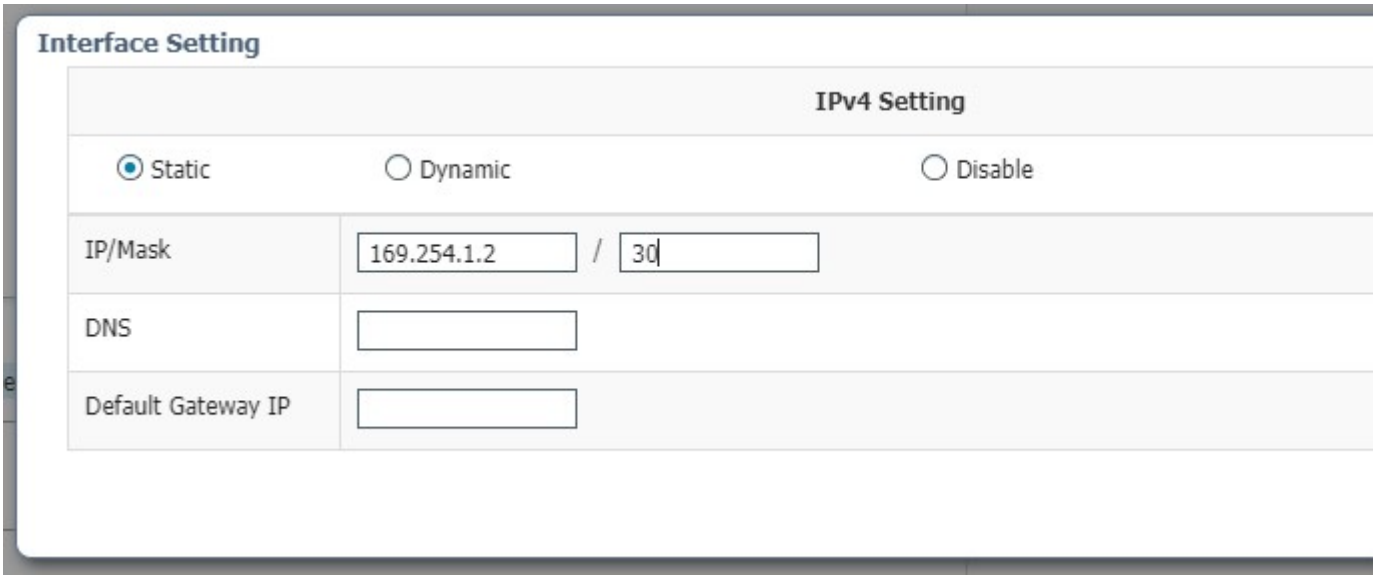
Figure 79: VPG0



Step 2 Click **Interface Setting** and apply the following configuration:

- Choose the **Static** option
- IP/Mask add **169.254.1.2 / 30**

Figure 80: IPV4 Setting



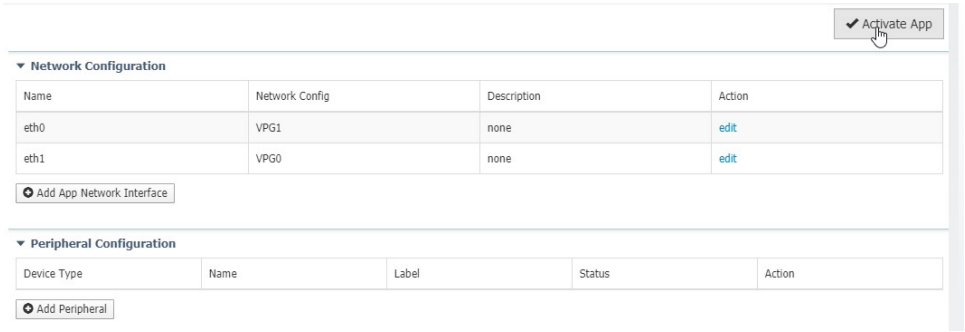
Activate the Application

Now the sensor application should be activated.

Procedure

Step 1 Click on **Activate App**. Refer to the following:

Figure 81: Activate the Application



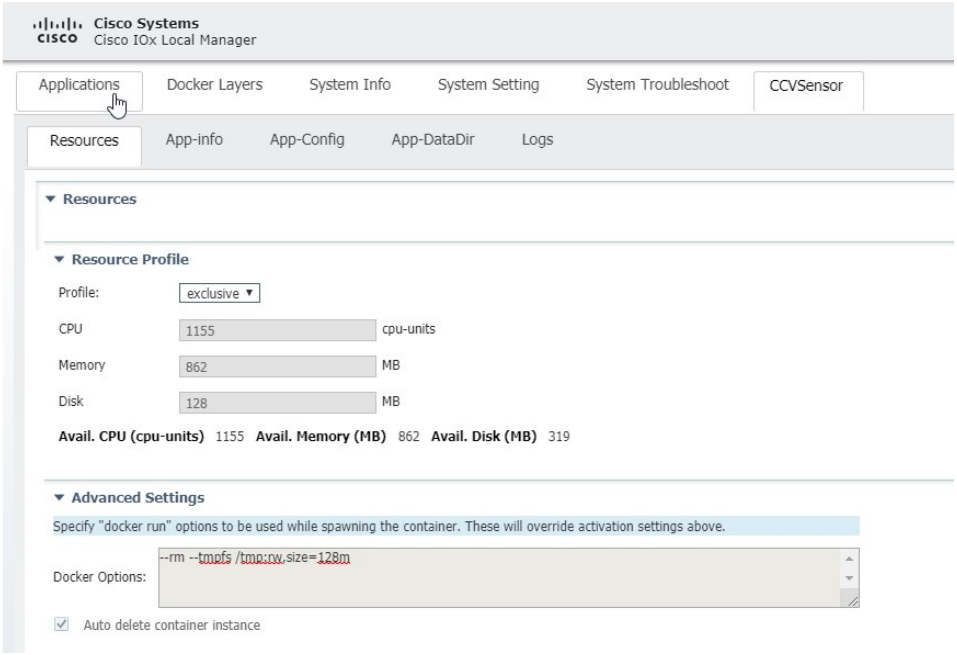
Step 2 The progress window appears. This may take several seconds to finish.

Figure 82: Activation Progress



Step 3 Click on **Applications** to display the app status. Refer to the following:

Figure 83: Applications Resources



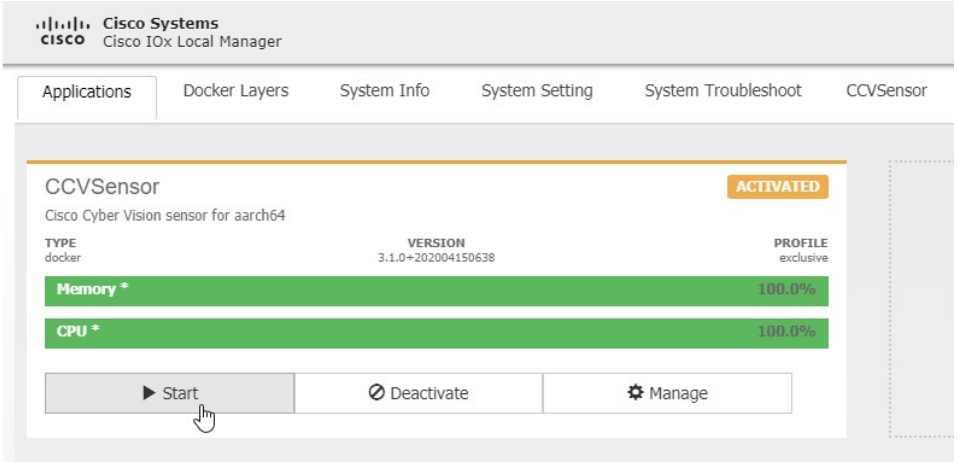
Step 4 The application is activated and needs to be started.

Starting the Application

Procedure

Step 1 Click on **Start**. Refer to the following:

Figure 84: Start Application



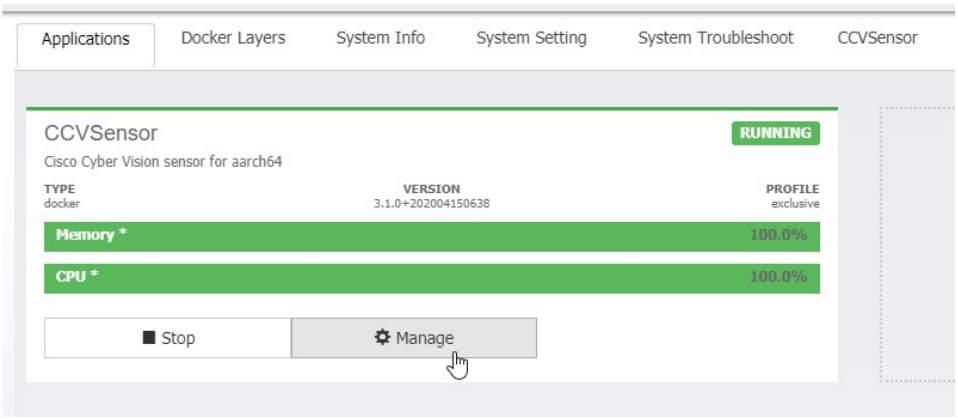
Step 2 The progress window appears. This may take several seconds to finish.

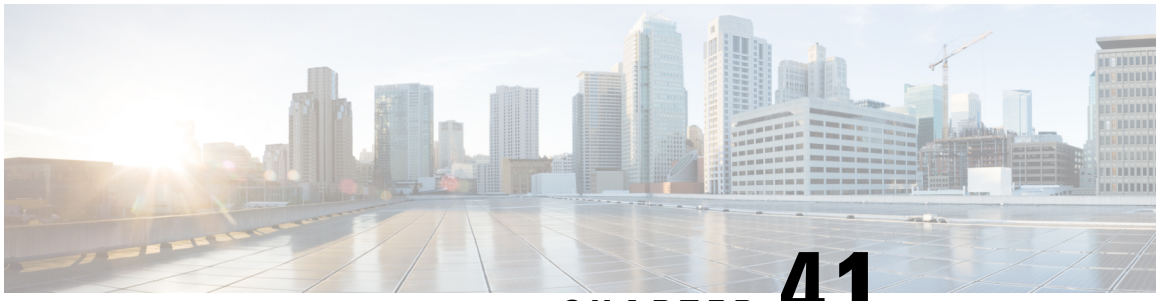
Figure 85: Progress Window



Step 3 After some time, the app status will change to running.

Figure 86: Application Running





CHAPTER 41

Troubleshooting

This chapter contains the following sections:

- Troubleshooting Overview, on page 455
- Understanding Diagnostic Mode, on page 455
- Before Contacting Cisco or Your Reseller, on page 456
- show interfaces Troubleshooting Command, on page 456
- Change the Configuration Register, on page 457
- Recovering a Lost Password, on page 460
- Reset the Password and Save Your Changes, on page 461
- Password Recovery Disable, on page 461
- Reset the Configuration Register Value, on page 462
- Configuring a Console Port Transport Map, on page 463
- Viewing Console Port, SSH, and Telnet Handling Configurations, on page 464
- Using the factory reset Commands, on page 466
- Support for Security-Enhanced Linux (SELinux), on page 466

Troubleshooting Overview

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.

The IR1101 supports the following interfaces:

- GigabitEthernet 0/0/0
- Cellular 0/1/0
- FastEthernet 0/0/1 to 0/0/4
- Async 0/2/0
- Cellular 0/x/x

• LORAWAN0/x/0

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 16.10.01
Cisco IOS Software [Gibraltar], ISR Software (ARMV8EL_LINUX_IOSD-UNIVERSALK9-M), Version 16.10.1,
RELEASE
SOFTWARE (fc1)
Technical Support: http://www.cisco.com/techsupport
Copyright (c) 1986-2018 by Cisco Systems, Inc.
Compiled Fri 09-Nov-18 18:08 by mcpre


Cisco IOS-XE software, Copyright (c) 2005-2018 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: IOS-XE ROMMON

Router uptime is 14 hours, 36 minutes
Uptime for this control processor is 14 hours, 37 minutes
System returned to ROM by reload
System restarted at 08:47:04 GMT Mon Nov 12 2018
System image file is "bootflash:ir1101-universalk9.16.10.01.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-package                                  Technology-package
Current                       Type                     Next reboot
------------------------------------------------------------------------------
network-essentials     Smart License                  network-essentials


Smart Licensing Status: UNREGISTERED/EVAL MODE

cisco IR1101-K9 (ARM64) processor (revision 1.2 GHz) with 711861K/6147K bytes of memory.
Processor board ID FCW222700MY
3 Virtual Ethernet interfaces
4 FastEthernet interfaces
1 Gigabit Ethernet interface
1 Serial interface
1 terminal line
2 Cellular interfaces
32768K bytes of non-volatile configuration memory.
4038072K bytes of physical memory.
3110864K bytes of Bootflash at bootflash:.
0K bytes of WebUI ODM Files at webui:.

Configuration register is 0x1821


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 0x01** 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

	Command or Action	Purpose
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 0x142** to reset the configuration register: **Example:** `rommon 2> confreg 0x142`	

Step 4	Initialize the router by entering the **reset** command:
	Example:
	`rommon 2> reset`
	Example:
	`--- System Configuration Dialog ---`
		The router cycles its power, and the configuration register is set to 0x142. The router uses the boot ROM system image, indicated by the system configuration dialog:

Step 5	Enter **no** in response to the prompts until the following message is displayed: **Example:** `Press RETURN to get started!`	
Step 6	Press **Return**. The following prompt appears: **Example:** `Router>`	

	Command or Action	Purpose

Step 7	Enter the enable command to enter enable mode. Configuration changes can be made only in enable mode:
	Example:
	`Router> enable`
	Example:
	`Router#`
		The prompt changes to the privileged EXEC prompt:

Step 8	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.

```
IR1101#config term
IR1101#license smart reservation
```



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

	Command or Action	Purpose
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`	

Password Recovery Disable

The No Service Password-Recovery is a Cisco IOS Platform independent feature/CLI which is available in Cisco IOS-XE devices. When the No Service Password-Recovery security feature is enabled, it prevents anyone with console access from using a break sequence (Control+C) during bootup to enter into rommon.

Note Ensure a valid Cisco IOS image is present in flash before enabling this feature. Failure to do so will result in the router going into a into boot loop. Hard power reset button is disabled if system has **no service password recovery**.

The following events will cause the router to go into rommon mode as standard IOS-XE behavior:

- config-reg setting is manual boot
- User opts to reset to factory default option

For more information and configuration steps, refer to the following:https://www.cisco.com/c/en/us/td/docs/ios-xml/ios/sec_usr_cfg/configuration/15-sy/sec-usr-cfg-15-sy-book/sec-no-svc-pw-recvry.html

Config register change issue with service password recovery update

When service password recovery is disabled, then the config register cannot be changed and will be stuck at 0x01. This issue was found on the IR1101 Router. For additional information see the tech note Understand Configuration Register Usage on all Routers.

Reset the Configuration Register Value

To reset the configuration register value after you have recovered or reconfigured a password, follow these steps:

Procedure

	Command or Action	Purpose
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** **Example:** `Router> enable`	Enables privileged EXEC mode. Enter your password if prompted.
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

	Command or Action	Purpose
		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. • *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.

```
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
```

Support for Security-Enhanced Linux (SELinux)

Security-Enhanced Linux is a set of patches to the Linux kernel and some utilities to incorporate a strong, flexible mandatory access control (MAC) architecture into the major subsystems of the kernel. SELinux provides an enhanced mechanism to enforce the separation of information based on confidentiality and integrity requirements, which allows threats of tampering and bypassing of application security mechanisms to be addressed and enables the confinement of damage that can be caused by malicious or flawed applications.

SELinux enforces mandatory access control policies that confine user programs and system servers to the minimum amount of privilege they require to do their jobs. This reduces or eliminates the ability of these programs and daemons to cause harm when compromised (for example, via buffer overflows or mis-configurations). This confinement mechanism operates independently of the traditional Linux access control mechanisms.

The are no additional requirements or configuration steps required to enable or operate the SELinux feature. The solution is enabled/operational by default as part of the base IOS-XE software on supported platforms.

The following are enhanced show commands that have been defined for viewing SELinux related audit logs.

show platform software audit all

show platform software audit summary

show platform software audit switch *<<1-8> | active | standby> <FRU identifier from a drop-down list>*

Command Examples

The following is a sample output of the **show software platform software audit summary** command:

```
Device# show platform software audit summary
==================================
AUDIT LOG ON switch 1
----------------------------------
AVC Denial count: 58
==================================
```

The following is a sample output of the **show software platform software audit all** command:

```
Device# show platform software audit all
==================================
AUDIT LOG ON switch 1
----------------------------------
========== START ============
type=AVC msg=audit(1539222292.584:100): avc:  denied  { read } for  pid=14017
comm="mcp_trace_filte" name="crashinfo" dev="rootfs" ino=13667
scontext=system_u:system_r:polaris_trace_filter_t:s0
tcontext=system_u:object_r:polaris_disk_crashinfo_t:s0 tclass=lnk_file permissive=1
type=AVC msg=audit(1539222292.584:100): avc:  denied  { getattr } for  pid=14017
comm="mcp_trace_filte" path="/mnt/sd1" dev="sda1" ino=2
scontext=system_u:system_r:polaris_trace_filter_t:s0
tcontext=system_u:object_r:polaris_disk_crashinfo_t:s0 tclass=dir permissive=1
type=AVC msg=audit(1539222292.586:101): avc:  denied  { getattr } for  pid=14028 comm="ls"
 path="/tmp/ufs/crashinfo" dev="tmpfs" ino=58407
scontext=system_u:system_r:polaris_trace_filter_t:s0
tcontext=system_u:object_r:polaris_ncd_tmp_t:s0 tclass=dir permissive=1
type=AVC msg=audit(1539222292.586:102): avc:  denied  { read } for  pid=14028 comm="ls"
name="crashinfo" dev="tmpfs" ino=58407 scontext=system_u:system_r:polaris_trace_filter_t:s0
 tcontext=system_u:object_r:polaris_ncd_tmp_t:s0 tclass=dir permissive=1
type=AVC msg=audit(1539438600.896:119): avc:  denied  { execute } for  pid=8300 comm="sh"
name="id" dev="loop0" ino=6982 scontext=system_u:system_r:polaris_auto_upgrade_server_rp_t:s0
 tcontext=system_u:object_r:bin_t:s0 tclass=file permissive=1
========== END ============
```

(output omitted for brevity)

The following is a sample output of the **show software platform software audit switch** command:

```
Device# show platform software audit switch active R0
========== START ============
type=AVC msg=audit(1539222292.584:100): avc:  denied  { read } for  pid=14017
comm="mcp_trace_filte" name="crashinfo" dev="rootfs" ino=13667
scontext=system_u:system_r:polaris_trace_filter_t:s0
tcontext=system_u:object_r:polaris_disk_crashinfo_t:s0 tclass=lnk_file permissive=1
type=AVC msg=audit(1539222292.584:100): avc:  denied  { getattr } for  pid=14017
comm="mcp_trace_filte" path="/mnt/sd1" dev="sda1" ino=2
scontext=system_u:system_r:polaris_trace_filter_t:s0
tcontext=system_u:object_r:polaris_disk_crashinfo_t:s0 tclass=dir permissive=1
type=AVC msg=audit(1539222292.586:101): avc:  denied  { getattr } for  pid=14028 comm="ls"
 path="/tmp/ufs/crashinfo" dev="tmpfs" ino=58407
scontext=system_u:system_r:polaris_trace_filter_t:s0
tcontext=system_u:object_r:polaris_ncd_tmp_t:s0 tclass=dir permissive=1
type=AVC msg=audit(1539222292.586:102): avc:  denied  { read } for  pid=14028 comm="ls"
name="crashinfo" dev="tmpfs" ino=58407 scontext=system_u:system_r:polaris_trace_filter_t:s0
 tcontext=system_u:object_r:polaris_ncd_tmp_t:s0 tclass=dir permissive=1
type=AVC msg=audit(1539438624.916:122): avc:  denied  { execute_no_trans } for  pid=8600
comm="auto_upgrade_se" path="/bin/bash" dev="rootfs" ino=7276
scontext=system_u:system_r:polaris_auto_upgrade_server_rp_t:s0
```

```
tcontext=system_u:object_r:shell_exec_t:s0 tclass=file permissive=1
type=AVC msg=audit(1539438648.936:123): avc:  denied  { execute_no_trans } for  pid=9307
comm="auto_upgrade_se" path="/bin/bash" dev="rootfs" ino=7276
scontext=system_u:system_r:polaris_auto_upgrade_server_rp_t:s0
tcontext=system_u:object_r:shell_exec_t:s0 tclass=file permissive=1
type=AVC msg=audit(1539438678.649:124): avc:  denied  { name_connect } for  pid=26421
comm="nginx" dest=8098 scontext=system_u:system_r:polaris_nginx_t:s0
tcontext=system_u:object_r:polaris_caf_api_port_t:s0 tclass=tcp_socket permissive=1
type=AVC msg=audit(1539438696.969:125): avc:  denied  { execute_no_trans } for  pid=10057
comm="auto_upgrade_se" path="/bin/bash" dev="rootfs" ino=7276
scontext=system_u:system_r:polaris_auto_upgrade_server_rp_t:s0
tcontext=system_u:object_r:shell_exec_t:s0 tclass=file permissive=1
type=AVC msg=audit(1539438732.973:126): avc:  denied  { execute_no_trans } for  pid=10858
comm="auto_upgrade_se" path="/bin/bash" dev="rootfs" ino=7276
scontext=system_u:system_r:polaris_auto_upgrade_server_rp_t:s0
tcontext=system_u:object_r:shell_exec_t:s0 tclass=file permissive=1
type=AVC msg=audit(1539438778.008:127): avc:  denied  { execute_no_trans } for  pid=11579
comm="auto_upgrade_se" path="/bin/bash" dev="rootfs" ino=7276
scontext=system_u:system_r:polaris_auto_upgrade_server_rp_t:s0
tcontext=system_u:object_r:shell_exec_t:s0 tclass=file permissive=1
type=AVC msg=audit(1539438800.156:128): avc:  denied  { name_connect } for  pid=26421
comm="nginx" dest=8098 scontext=system_u:system_r:polaris_nginx_t:s0
tcontext=system_u:object_r:polaris_caf_api_port_t:s0 tclass=tcp_socket permissive=1
type=AVC msg=audit(1539438834.099:129): avc:  denied  { execute_no_trans } for  pid=12451
comm="auto_upgrade_se" path="/bin/bash" dev="rootfs" ino=7276
scontext=system_u:system_r:polaris_auto_upgrade_server_rp_t:s0
tcontext=system_u:object_r:shell_exec_t:s0 tclass=file permissive=1
type=AVC msg=audit(1539438860.907:130): avc:  denied  { name_connect } for  pid=26421
comm="nginx" dest=8098 scontext=system_u:system_r:polaris_nginx_t:s0
tcontext=system_u:object_r:polaris_caf_api_port_t:s0 tclass=tcp_socket permissive=1
========== END ============
==================================
```

Syslog Message Reference

Facility-Severity-Mnemonic

- %SELINUX-3-MISMATCH

Severity-Meaning

- ERROR LEVEL Log

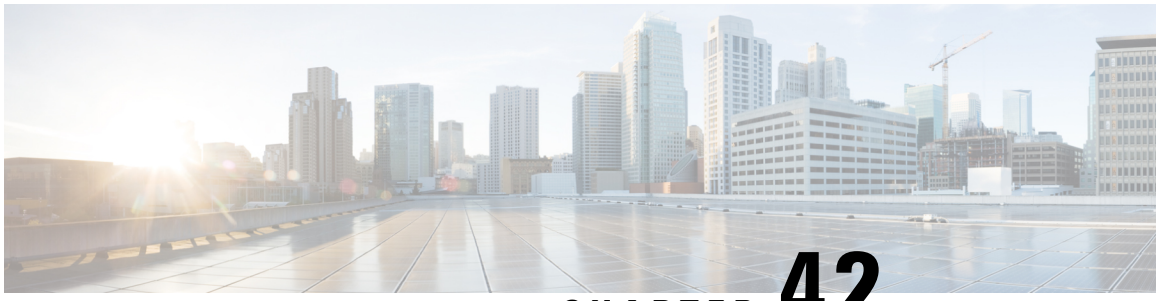
Message Explanation

- A resource access was made by the process for which a resource access policy is not defined. The operation was flagged but not denied.
- The operation continued successfully and was not disrupted. A system log has been generated about the missing policy for resource access by the process as denied operation.

Recommended Action

- Please contact CISCO TAC with the following relevant information as attachments:
  - The message exactly as it appears on the console or in the system log.
  - Output of "show tech-support" (text file)
  - Archive of Btrace files from the box using the following command ("request platform software trace archive target <URL>") For Example:

Device#**request platform software trace archive target flash:selinux_btrace_logs**



CHAPTER 42

System Messages

This chapter contains the following sections:

- Information About Process Management, on page 471
- How to Find Error Message Details, on page 471

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: http://www.cisco.com/cisco/psn/bssprt/bss. 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: http://www.cisco.com/cisco/psn/bssprt/bss. 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: http://www.cisco.com/cisco/psn/bssprt/bss. 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: http://www.cisco.com/cisco/psn/bssprt/bss. 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.

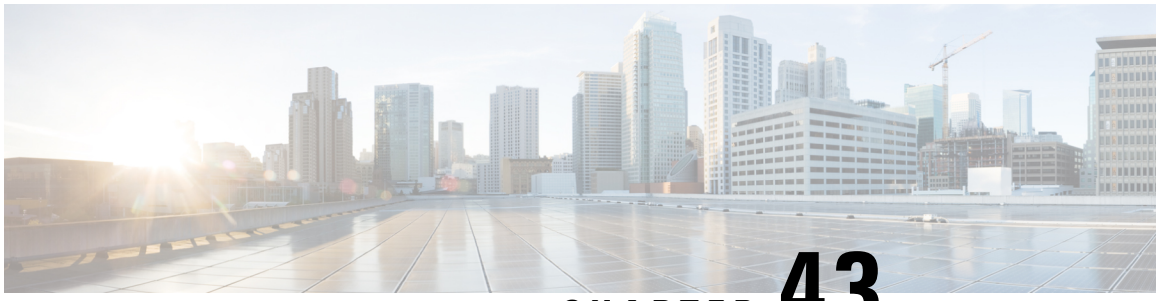
Error Message: `%PMAN-6-PROCSTART : The process [chars] has started`

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 43

Environmental Monitoring

- Environmental Monitoring, on page 477
- Environmental Monitoring and Reporting Functions, on page 477
- Environmental Monitoring Functions, on page 478
- Environmental Reporting Functions, on page 479
- SNMP Polling of Temperature OID, on page 485
- Additional References, on page 486
- Technical Assistance, on page 486

Environmental Monitoring

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 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 478
- Environmental Reporting Functions, on page 479

Environmental Monitoring Functions

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 32: 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: Example

```
Router# show diag all eepromMIDPLANE EEPROM data:

Product Identifier (PID) : IR1101-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) : IR1101-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) : IR1101-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) : IR1101-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) : IR1101-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: Example

```
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: Example

```
Router# show environment all
Sensor List: Environmental Monitoring
```

```
Sensor Location State Reading
Temp: LM75BXXX R0 Normal 48 Celsius
```

show inventory: Example

```
Router# show inventory
++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
INFO: Please use "show license UDI" to get serial number for licensing.
++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
NAME: "Chassis", DESCR: "IR1101 Base Chassis"
PID: IR1101-K9 , VID: V00 , SN: FCW2132TH0Z

NAME: "Module 0 - Mother Board", DESCR: "Cisco IR1101 motherboard"
PID: IR1101-K9 , VID: , SN:

NAME: "module subslot 0/0", DESCR: "IR1101-ES-5"
PID: IR1101-ES-5 , VID: V01 , SN:

NAME: "subslot 0/0 transceiver 0", DESCR: "GE SX"
PID: GLC-SX-MM-RGD , VID: V01 , SN: FNS16370HL4

NAME: "module subslot 0/1", DESCR: "P-LTE-US Module"
PID: P-LTE-US , VID: V01 , SN: FOC21333R92

NAME: "Modem 0 on Cellular0/1/0", DESCR: "Sierra Wireless WP7603"
PID: WP7603 , VID: 10000, SN: 359528080000794
```

show platform: Example

```
Router# show platform
Chassis type: IR1101-K9

Slot Type State Insert time (ago)
--------- ------------------- --------------------- -----------------
0 IR1101-K9 ok 01:52:41
0/0 IR1101-ES-5 ok 01:51:35
R0 IR1101-K9 ok, active 01:52:41
F0 IR1101-K9 init, active 01:52:41
Router#
```

show platform diag: Example

```
Router# show platform diag
Chassis type: IR1101-K9

Slot: 0, IR1101-K9
Running state : ok
Internal state : online
Internal operational state : ok
Physical insert detect time : 00:00:25 (5d02h ago)
Software declared up time : 00:01:07 (5d02h ago)
CPLD version :
Firmware version : 1.3
```

```
Sub-slot: 0/0, IR1101-ES-5
Operational status : ok
Internal state : inserted
Physical insert detect time : 00:02:21 (5d02h ago)
Logical insert detect time : 00:02:21 (5d02h ago)

Sub-slot: 0/1, P-LTE-US
Operational status : ok
Internal state : inserted
Physical insert detect time : 00:02:21 (5d02h ago)
Logical insert detect time : 00:02:21 (5d02h ago)

Slot: R0, IR1101-K9
Running state : ok, active
Internal state : online
Internal operational state : ok
Physical insert detect time : 00:00:25 (5d02h ago)
Software declared up time : 00:00:25 (5d02h ago)
CPLD version : 00000000
Firmware version : 1.2

Slot: F0, IR1101-K9
Running state : init, active
Internal state : online
Internal operational state : ok
Physical insert detect time : 00:00:25 (5d02h ago)
Software declared up time : 00:01:10 (5d02h ago)
Hardware ready signal time : 00:00:00 (never ago)
Packet ready signal time : 00:00:00 (never ago)
CPLD version : 00000000
Firmware version : 1.2

Router#
```

show platform software status control-processor: Example

```
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: Example

```
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) : IR1101-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: Example

```
Router# show version
Cisco IOS XE Software, Version 16.10.01
Cisco IOS Software [Gibraltar], ISR Software (ARMV8EL_LINUX_IOSD-UNIVERSALK9-M), Version
16.10.1prd7, RELEASE SOFTWARE (fc1)
Technical Support: http://www.cisco.com/techsupport
Copyright (c) 1986-2018 by Cisco Systems, Inc.
Compiled Wed 31-Oct-18 23:27 by mcpre
```

```
Cisco IOS-XE software, Copyright (c) 2005-2018 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: IOS-XE ROMMON

Router uptime is 1 hour, 53 minutes
Uptime for this control processor is 1 hour, 54 minutes
System returned to ROM by reload
System image file is "usb0:ir1101-universalk9.16.10.01prd7.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-package Technology-package
Current Type Next reboot
------------------------------------------------------------------------------
network-advantage Smart License network-advantage


Smart Licensing Status: UNREGISTERED/EVAL EXPIRED

cisco IR1101-K9 (ARM64) processor (revision 1.2 GHz) with 711867K/6147K bytes of memory.
Processor board ID FCW2150TH0F
1 Virtual Ethernet interface
4 FastEthernet interfaces
1 Gigabit Ethernet interface
1 Serial interface
1 terminal line
32768K bytes of non-volatile configuration memory.
4038072K bytes of physical memory.
3110864K bytes of Bootflash at bootflash:.
0K bytes of WebUI ODM Files at webui:.
30670832K bytes of USB Flash at usbflash0:.

Configuration register is 0x0 (will be 0x2102 at next reload)
```

```
Router#
```

show power: Example

```
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

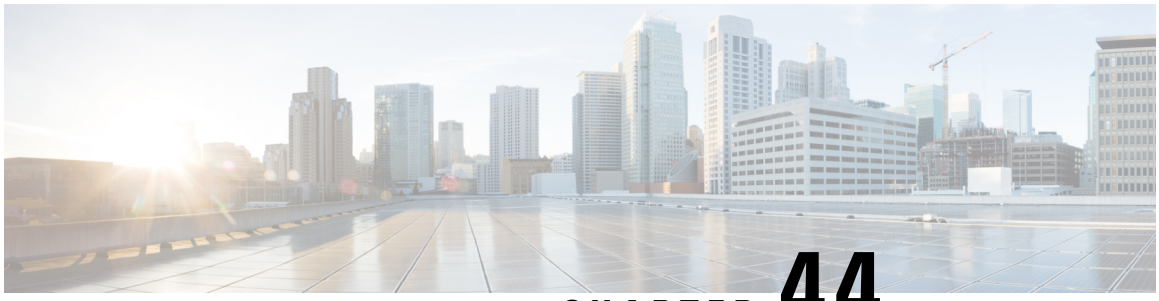
The following sections provide references related to the power efficiency management feature.

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 44

WAN Monitoring

This chapter contains the following sections:

- Information About WANMon, on page 487
- Prerequisites , on page 488
- Guidelines and Limitations, on page 488
- Configuring WANMon, on page 488
- Verifying WANMon Configuration, on page 490
- Configuration Examples, on page 491

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
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, cellular0/1/0 or cellular0/3/0). 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, cellular0/1/0 or cellular0/3/0). <*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 {cellular0/1/0 {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 {cellular0/1/0 {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 {cellular0/1/0 {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   {cell0/1/0 {ipsla 1}}
```

Configuration Examples

The following examples are provided:

WANMon Cellular Interface Configuration Example

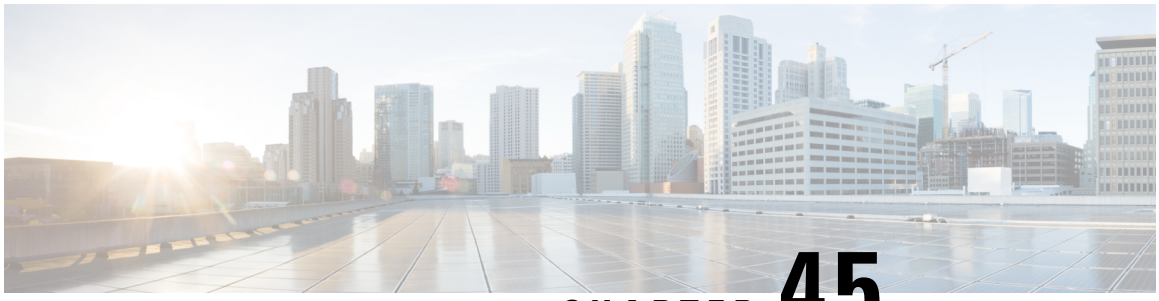
```
track 1 ip sla 1
ip sla 1
 icmp-echo 172.27.166.250
 timeout 6000
 frequency 300
ip sla schedule 1 life forever start-time now
event manager environment wanmon_if_list {cellular0/1/0 {ipsla 1}}
event manager policy tm_wanmon.tcl authorization bypass
```

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 {cellular0/1/0 {ipsla 1} {stub 21}}

event manager policy tm_wanmon.tcl authorization bypass
```



CHAPTER 45

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 493
- Monitoring Hardware Using Alarms, on page 498

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, on page 493
- Cisco IOS Process Resources, on page 493
- Overall Control Plane Resources, on page 497

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#33e0077971693714bd2b0bc347d77489
Address Bytes Prev Next Ref PrevF NextF what Alloc PC

Head Total(b) Used(b) Free(b) Lowest(b) Largest(b)
Processor 7F68ECD010 728952276 281540188 447412088 445683380 234766720
lsmpi_io 7F6852A1A8 6295128 6294304 824 824 412
Dynamic heap limit(MB) 200 Use(MB) 0



Processor memory

Address Bytes Prev Next Ref PrevF NextF what Alloc PC
7F68ECD010 0000000568 00000000 7F68ECD2A0 001 -------- -------- *Init* :400000+60E37C4
7F68ECD2A0 0000032776 7F68ECD010 7F68ED5300 001 -------- -------- Managed Chunk Q
:400000+60D12A8
7F68ED5300 0000000056 7F68ECD2A0 7F68ED5390 001 -------- -------- *Init* :400000+3B0C610
7F68ED5390 0000012808 7F68ED5300 7F68ED85F0 001 -------- -------- *Init* :400000+B8A5D64
Address Bytes Prev Next Ref PrevF NextF what Alloc PC
7F68ED85F0 0000032776 7F68ED5390 7F68EE0650 001 -------- -------- List Elements
:400000+60A4A9C
7F68EE0650 0000032776 7F68ED85F0 7F68EE86B0 001 -------- -------- List Headers :400000+60A4AD8
7F68EE86B0 0000032776 7F68EE0650 7F68EF0710 001 -------- -------- IOSXE Process S
:400000+11924CC
7F68EF0710 0000032776 7F68EE86B0 7F68EF8770 001 -------- -------- IOSXE Queue Pro
:400000+1192510
7F68EF8770 0000065544 7F68EF0710 7F68F087D0 001 -------- -------- IOSXE Queue Bal
:400000+1192554
7F68F087D0 0000000328 7F68EF8770 7F68F08970 001 -------- -------- *Init* :400000+B89E1D8
7F68F08970 0000000328 7F68F087D0 7F68F08B10 001 -------- -------- *Init* :400000+B89E1D8
7F68F08B10 0000000328 7F68F08970 7F68F08CB0 001 -------- -------- *Init* :400000+B89E1D8
7F68F08CB0 0000000360 7F68F08B10 7F68F08E70 001 -------- -------- Process Events
:400000+60F9CD4
7F68F08E70 0000000056 7F68F08CB0 7F68F08F00 001 -------- -------- SDB String :400000+605981C
7F68F08F00 0000000080 7F68F08E70 7F68F08FA8 001 -------- -------- Init :400000+60599E4
Address Bytes Prev Next Ref PrevF NextF what Alloc PC
7F68F08FA8 0000036872 7F68F08F00 7F68F12008 001 -------- -------- *Init* :400000+11891E8
7F68F12008 0000010008 7F68F08FA8 7F68F14778 001 -------- -------- Platform VM Pag
:400000+11AD244
7F68F14778 0000002008 7F68F12008 7F68F14FA8 001 -------- -------- *Init*
iosd_crb_ir1101_unix:7F8EB59000+5CC1C
7F68F14FA8 0000200712 7F68F14778 7F68F46008 001 -------- -------- Interrupt Stack
:400000+11891E8
7F68F46008 0000003008 7F68F14FA8 7F68F46C20 001 -------- -------- Watched Semapho
:400000+60FE448
7F68F46C20 0000000328 7F68F46008 7F68F46DC0 001 -------- -------- *Init* :400000+B89E1D8
7F68F46DC0 0000000096 7F68F46C20 7F68F46E78 001 -------- -------- Init :400000+60599E4
7F68F46E78 0000000216 7F68F46DC0 7F68F46FA8 001 -------- -------- *Init* :400000+60ED228
7F68F46FA8 0000036872 7F68F46E78 7F68F50008 001 -------- -------- *Init* :400000+11891E8
7F68F50008 0000000896 7F68F46FA8 7F68F503E0 001 -------- -------- Watched Message
:400000+60FE4A8
7F68F503E0 0000002008 7F68F50008 7F68F50C10 001 -------- -------- Watcher Message
:400000+60FE4D8
Address Bytes Prev Next Ref PrevF NextF what Alloc PC
7F68F50C10 0000000360 7F68F503E0 7F68F50DD0 001 -------- -------- Process Events
:400000+60F9CD4
7F68F50DD0 0000000184 7F68F50C10 7F68F50EE0 001 -------- -------- *Init* :400000+60ED918
7F68F50EE0 0000000112 7F68F50DD0 7F68F50FA8 001 -------- -------- *Init* :400000+60B57CC
7F68F50FA8 0000036872 7F68F50EE0 7F68F5A008 001 -------- -------- *Init* :400000+11891E8
7F68F5A008 0000002336 7F68F50FA8 7F68F5A980 001 -------- -------- Process Array
:400000+6102A4C
7F68F5A980 0000000184 7F68F5A008 7F68F5AA90 001 -------- -------- *Init* :400000+60ED918
```

```
7F68F5AA90 0000000184 7F68F5A980 7F68F5ABA0 001 -------- -------- *Init* :400000+60ED918
7F68F5ABA0 0000000184 7F68F5AA90 7F68F5ACB0 001 -------- -------- *Init* :400000+60ED918
7F68F5ACB0 0000000184 7F68F5ABA0 7F68F5ADC0 001 -------- -------- *Init* :400000+60ED918
7F68F5ADC0 0000000184 7F68F5ACB0 7F68F5AED0 001 -------- -------- *Init* :400000+60ED918
```

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
PID Runtime(ms) Invoked uSecs 5Sec 1Min 5Min TTY Process
1 0 17 0 0.00% 0.00% 0.00% 0 Chunk Manager
2 552 1205 458 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 36 13 2769 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 4052 920 4404 0.23% 0.09% 0.06% 0 Check heaps
10 12 101 118 0.00% 0.00% 0.00% 0 Pool Manager
11 0 1 0 0.00% 0.00% 0.00% 0 DiscardQ Backgro
PID Runtime(ms) Invoked uSecs 5Sec 1Min 5Min TTY Process
12 0 2 0 0.00% 0.00% 0.00% 0 Timers
13 0 163 0 0.00% 0.00% 0.00% 0 WATCH_AFS
14 0 2 0 0.00% 0.00% 0.00% 0 ATM AutoVC Perio
15 0 2 0 0.00% 0.00% 0.00% 0 ATM VC Auto Crea
16 76 3024 25 0.00% 0.00% 0.00% 0 IOSXE heartbeat
17 0 13 0 0.00% 0.00% 0.00% 0 DB Lock Manager
18 0 1 0 0.00% 0.00% 0.00% 0 DB Notification
19 0 1 0 0.00% 0.00% 0.00% 0 IPC Apps Task
20 0 1 0 0.00% 0.00% 0.00% 0 ifIndex Receive
21 36 1210 29 0.00% 0.00% 0.00% 0 IPC Event Notifi
22 72 5904 12 0.00% 0.00% 0.00% 0 IPC Mcast Pendin
PID Runtime(ms) Invoked uSecs 5Sec 1Min 5Min TTY Process
23 0 1 0 0.00% 0.00% 0.00% 0 Platform appsess
24 0 101 0 0.00% 0.00% 0.00% 0 IPC Dynamic Cach
25 16 1210 13 0.00% 0.00% 0.00% 0 IPC Service NonC
26 0 1 0 0.00% 0.00% 0.00% 0 IPC Zone Manager
27 64 5904 10 0.00% 0.00% 0.00% 0 IPC Periodic Tim
28 76 5904 12 0.00% 0.00% 0.00% 0 IPC Deferred Por
29 0 1 0 0.00% 0.00% 0.00% 0 IPC Process leve
30 0 1 0 0.00% 0.00% 0.00% 0 IPC Seat Manager
31 8 346 23 0.00% 0.00% 0.00% 0 IPC Check Queue
32 0 1 0 0.00% 0.00% 0.00% 0 IPC Seat RX Cont
33 0 1 0 0.00% 0.00% 0.00% 0 IPC Seat TX Cont
PID Runtime(ms) Invoked uSecs 5Sec 1Min 5Min TTY Process
34 48 606 79 0.00% 0.00% 0.00% 0 IPC Keep Alive M
35 28 1210 23 0.00% 0.00% 0.00% 0 IPC Loadometer
36 0 1 0 0.00% 0.00% 0.00% 0 IPC Session Deta
37 0 1 0 0.00% 0.00% 0.00% 0 SENSOR-MGR event
38 4 606 6 0.00% 0.00% 0.00% 0 Compute SRP rate
39 0 1 0 0.00% 0.00% 0.00% 0 MEMLEAK PROCESS
40 0 1 0 0.00% 0.00% 0.00% 0 ARP Input
41 112 6331 17 0.00% 0.00% 0.00% 0 ARP Background
42 0 2 0 0.00% 0.00% 0.00% 0 ATM Idle Timer
43 0 1 0 0.00% 0.00% 0.00% 0 ATM ASYNC PROC
44 0 1 0 0.00% 0.00% 0.00% 0 CEF MIB API
--More--
...
show process cpu platform sorted

CPU utilization for five seconds: 11%, one minute: 12%, five minutes: 12%
```

```
Core 0: CPU utilization for five seconds: 1%, one minute: 3%, five minutes: 3%
Core 1: CPU utilization for five seconds: 1%, one minute: 3%, five minutes: 3%
Core 2: CPU utilization for five seconds: 1%, one minute: 1%, five minutes: 1%
Core 3: CPU utilization for five seconds: 42%, one minute: 42%, five minutes: 42%
Pid PPid 5Sec 1Min 5Min Status Size Name
------------------------------------------------------------------------------
18246 17700 34% 34% 34% S 272500 qfp-ucode-sparr
18297 16477 1% 1% 1% S 165768 fman_fp_image
9992 9121 1% 1% 1% S 743608 linux_iosd-imag
27122 26048 0% 0% 0% S 8460 nginx
26048 25864 0% 0% 0% S 19252 nginx
25928 1 0% 0% 0% S 2960 rotee
25864 1 0% 0% 0% S 3532 pman.sh
24212 2 0% 0% 0% S 0 kworker/u8:0
19648 8282 0% 0% 0% S 220 sleep
19635 10903 0% 0% 0% S 212 sleep
18121 17675 0% 0% 0% S 10968 ngiolite
17979 1 0% 0% 0% S 1660 rotee
17863 2 0% 0% 0% S 0 kworker/1:0
17859 1 0% 0% 0% S 2836 rotee
17737 17095 0% 0% 0% S 56828 iomd
17700 13380 0% 0% 0% S 3556 pman.sh
17675 12798 0% 0% 0% S 3524 pman.sh
17518 16854 0% 0% 0% S 15024 hman
17312 1 0% 0% 0% S 2828 rotee
17095 12798 0% 0% 0% S 3568 pman.sh
17085 1 0% 0% 0% S 2876 rotee
16942 2 0% 0% 0% S 0 kworker/0:1
16892 14768 0% 0% 0% S 108952 cpp_cp_svr
16854 13380 0% 0% 0% S 3568 pman.sh
16716 1 0% 0% 0% S 2996 rotee
16664 15963 0% 0% 0% S 51096 cpp_sp_svr
16477 13380 0% 0% 0% S 3540 pman.sh
16326 15536 0% 0% 0% S 39852 cpp_ha_top_leve
16270 1 0% 0% 0% S 2972 rotee
15963 13380 0% 0% 0% S 3528 pman.sh
15779 15163 0% 0% 0% S 55208 cpp_driver
15730 1 0% 0% 0% S 1640 rotee
15536 13380 0% 0% 0% S 3528 pman.sh
15412 1 0% 0% 0% S 1716 rotee
15274 14681 0% 0% 0% S 15004 hman
15163 13380 0% 0% 0% S 3624 pman.sh
15083 14361 0% 0% 0% S 26792 cman_fp
15057 1 0% 0% 0% S 1660 rotee
14891 1 0% 0% 0% S 2868 rotee
14768 13380 0% 0% 0% S 3568 pman.sh
14722 14127 0% 0% 0% S 27536 cmcc
14717 14108 0% 0% 0% S 15220 btman
14681 12798 0% 0% 0% S 3572 pman.sh
14627 1 0% 0% 0% S 2996 rotee
14361 13380 0% 0% 0% S 3596 pman.sh
14338 1 0% 0% 0% S 2984 rotee
14314 1 0% 0% 0% S 2824 rotee
14155 13577 0% 0% 0% S 15128 btman
14127 12798 0% 0% 0% S 3612 pman.sh
14108 13380 0% 0% 0% S 3572 pman.sh
13813 13380 0% 0% 0% S 252 inotifywait
--More--
```

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 4 seconds ago
Load Average: healthy
1-Min: 0.29, status: healthy, under 5.00
5-Min: 0.51, status: healthy, under 5.00
15-Min: 0.54, status: healthy, under 5.00
Memory (kb): healthy
Total: 4038072
Used: 2872136 (71%), status: healthy
Free: 1165936 (29%)
Committed: 2347228 (58%), under 90%
Per-core Statistics
CPU0: CPU Utilization (percentage of time spent)
User: 1.00, System: 0.70, Nice: 0.00, Idle: 97.88
IRQ: 0.30, SIRQ: 0.10, IOwait: 0.00
CPU1: CPU Utilization (percentage of time spent)
User: 0.70, System: 0.30, Nice: 0.00, Idle: 98.48
IRQ: 0.30, SIRQ: 0.20, IOwait: 0.00
CPU2: CPU Utilization (percentage of time spent)
User: 0.20, System: 1.11, Nice: 0.00, Idle: 98.27
IRQ: 0.40, SIRQ: 0.00, IOwait: 0.00
CPU3: CPU Utilization (percentage of time spent)
User: 8.23, System: 24.37, Nice: 0.00, Idle: 58.00
IRQ: 9.26, SIRQ: 0.11, IOwait: 0.00
```

```
Router# show platform software status control-processor brief
Load Average
Slot Status 1-Min 5-Min 15-Min
RP0 Healthy 0.28 0.46 0.52

Memory (kB)
Slot Status Total Used (Pct) Free (Pct) Committed (Pct)
RP0 Healthy 4038072 2872672 (71%) 1165400 (29%) 2349820 (58%)

CPU Utilization
Slot CPU User System Nice Idle IRQ SIRQ IOwait
RP0 0 0.70 0.20 0.00 98.58 0.30 0.20 0.00
1 1.10 0.90 0.00 97.59 0.30 0.10 0.00
2 0.40 1.31 0.00 97.87 0.40 0.00 0.00
3 8.00 26.55 0.00 56.33 8.99 0.11 0.00
```

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.

Report Alarms Through SNMP

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)

Yang Support for IO Ports

This feature increases the compatibility between the Command Line Interface and the Yang Model. Cisco IOS-XE Yang Data Models are found here:

https://github.com/YangModels/yang/tree/master/vendor/cisco/xe

Each release has a directory, and the 17.3.1 release is found under 1731. The two modules for Digital IO are Cisco-IOS-XE-digital-io-oper and Cisco-IOS-XE-digitalio.

The following are relevant IOS-XE CLI commands available:

Show Commands

- show run
- show alarm
- show led

Configuration Commands

- alarm contact attach-to-iox
- no alarm contact attach-to-iox
- alarm contact 1 enable enable
- no alarm contact <1-4> enable
- alarm contact <1-4> application <wet | dry>
- no alarm contact <1-4> application
- alarm contact <1-4> description <alarm description>
- no alarm contact <1-4> description
- alarm contact <1-4> severity <critical | major | minor | none>
- no alarm contact <1-4> severity
- alarm contact <1-4> threshold <1600-2700>
- no alarm contact <1-4> threshold
- alarm contact <1-4> trigger <closed | open>
- no alarm contact <1-4> trigger
- alarm contact <1-4> output <1 | 0>
- alarm contact <1-4> output relay temperature <critical | major | minor>
- alarm contact <1-4> output relay input-alarm <0-4>

- no alarm contact <1-4> output

SNMP MIB for Digital I/O

Digital I/O is similar to the ALARM IN and ALARM OUT supported in other IR devices. On other devices, ALARM IN is a dedicated input and the ALARM OUT is a dedicated output. With Digital I/O it can be input or output. There are 4 Digital I/O available on the IR1101 with an IRM-1100 Expansion Module.

MIB support will reflect the show alarm output for digital I/O only.

CISCO-DIGITAL-IO-MIB.my will have 4 digital I/O nodes. Each digital I/O node will have corresponding attributes like description, enable, severity, application, output, threshold, trigger leaf nodes for each digital I/O nodes.

SNMP MIB supports the show power CLI

SNMP MIB support for the **show power** CLI is available through a new mib file: CISCO-ENTITY-SENSOR-MIB.my

The following is an example of the **show power** CLI:

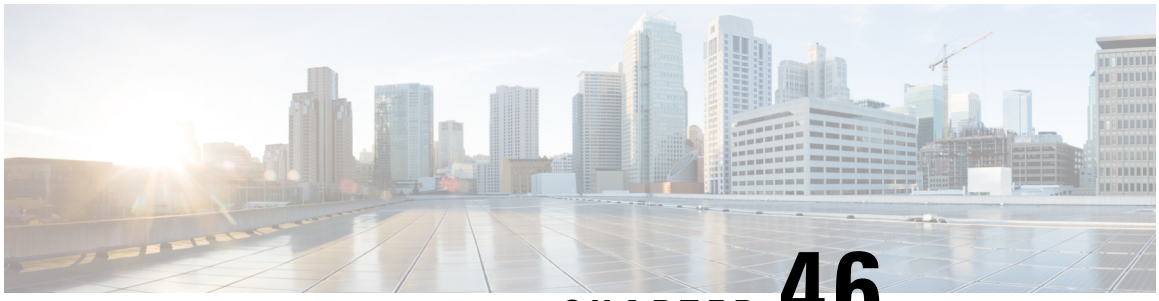
```
#show power
Main PSU :
    Total Power Consumed: 8.77 Watts
    Configured Mode : N/A
    Current runtime state same : N/A
    PowerSupplySource : External PS
```

The following is an example of the CISCO-ENTITY-SENSOR-MIB.my MIB:

```
SensorDataType (INTEGER) watts(6)
SensorDataScale (INTEGER) milli(8)
SensorValue(INTEGER) 8770
```

Use the following commands to configure:

```
Router#config term
Router#(config) snmp-server community public RW
Router#(config) end
```



CHAPTER 46

gRPC Network Operations

- gRPC Network Operations Interface updates, on page 503
- GNMI Broker update, on page 503

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

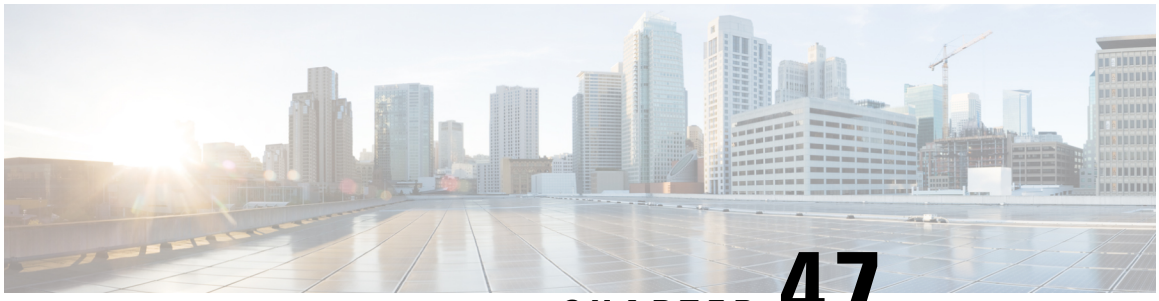
A GNMI Broker update is a software enhancement that:

- 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 47

Support for YANG

- Support for YANG Data Models (Call-home), on page 505
- Yang Data Model Support for Scada, on page 505
- Support for Model Driven support for GNMI Telemetry Dial-In, on page 505

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 17.1 release of IOS-XE on IR1101. The following references are available for earlier YANG models:

https://github.com/YangModels/yang/tree/master/vendor/cisco/xe/1651

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 Scada

Cisco IOS XE 17.1.1 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/1711

Support for Model Driven support for GNMI Telemetry Dial-In

Similar to YANG models, there is support on IOS-XE for open source models defined by Google and is referred as Google Network Management Interface (GNMI). Configurations of GNMI can be verified either with Secure or Insecure Mode.

- Secure Mode

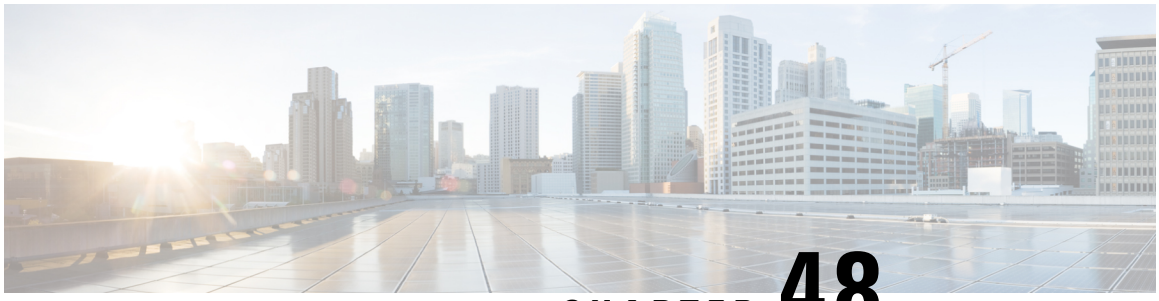
Secure Mode establishes secure connection using OpenSSL certificates between client and server. It sends GNMI telemetry updates using open source gnmi_cli tool.

- Insecure Mode

Insecure Mode sends GNMI telemetry updates between client and server using open source pygnmi tool.

For additional information about GNMI Telemetry see the following reference:

Programmability Configuration Guide, Cisco IOS XE



CHAPTER 48

Cellular Network Slicing

This document outlines the architectural framework and operational parameters required to deploy cellular network slicing effectively.

- Cellular network slicing, on page 507
- Restrictions for cellular network slicing, on page 508
- Configure cellular network slicing, on page 508
- Example for configuring cellular network slicing , on page 509
- Verify cellular network slicing configuration, on page 509

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 33: 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

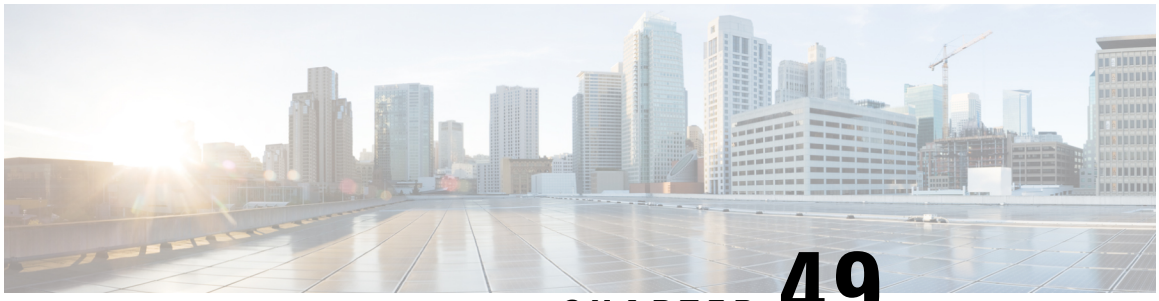
        Device(config-controller)# profile id 3 apn apn-ns authentication none pdn-type
ipv4 slice-type
        embb slice-differentiator
        <0-16777214>  Slice Differentiator
        Device(config-controller)#
        Device(config-controller)#$ pdn ipv6  slice-type miot slice-differentiator 6
        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 49

H-VPLS N-PE Redundancy for MPLS Access

This chapter contains the following sections:

- H-VPLS, on page 511

H-VPLS

The H-VPLS N-PE Redundancy for MPLS Access feature enables two network provider edge (N-PE) devices to provide failover services to a user provider edge (U-PE) device in a hierarchical virtual private LAN service (H-VPLS). Redundant N-PE devices increase stability and reliability by protecting against link and node failures. If a failure occurs in the network that disables one N-PE device from transmitting data, the other N-PE device takes over.

Table 34: Feature History Table

Feature name	Release information	Feature description
H-VPLS N-PE Redundancy for MPLS Access	Release 17.18.2	This feature enables a U-PE to connect to two N-PEs using active/backup pseudowires, delivering rapid switchover on link or node failure without service reconfiguration.

Prerequisites for H-VPLS

- Configure your H-VPLS network.
- Enable the MPLS Traffic Engineering-Fast Reroute feature in the MPLS core to improve convergence.
- Enable the L2VPN Pseudowire Redundancy feature on the U-PE devices for MPLS access.

Restrictions for H-VPLS

- You cannot use this feature with the VPLS Autodiscovery feature on pseudowires that connect to U-PE devices. When you create the virtual private LAN service (VPLS), manually create the virtual forwarding interface (VFI).

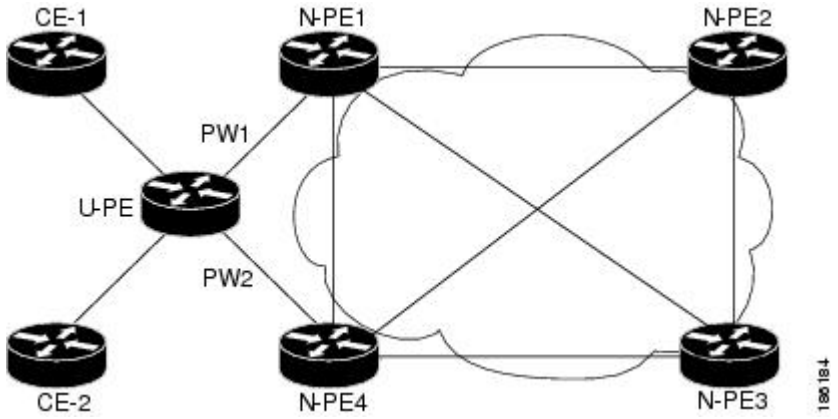
- You cannot configure more than one pseudowire to carry the bridge protocol data unit (BPDU) information between the network provider edge (N-PE) devices.
- You cannot configure a local loopback address as a neighbor when you configure the H-VPLS N-PE Redundancy feature on N-PE devices.
- Only two N-PE devices can be connected to each U-PE device.

H-VPLS N-PE redundancy with MPLS access based on pseudowire redundancy

For H-VPLS Redundancy with MPLS Access, the MPLS network uses redundant pseudowires. These pseudowires connect the MPLS network to N-PE devices in the VPLS core.

As shown in the figure, a primary pseudowire carries data between the U-PE and N-PE devices. If the U-PE path fails, a backup pseudowire and redundant N-PE device activate to continue data transport.

Figure 87: H-VPLS N-PE redundancy for MPLS access based on pseudowire Redundancy



Configuration of H-VPLS N-PE Redundancy for MPLS Access

Specify the devices in the layer 2 VPN VFI

Before you begin

Ensure that the device is in configuration mode.

Procedure

Step 1 Use the **l2vpn vfi context** *name* command to establish a L2VPN VFI between two or more separate networks, and enter L2VFI configuration mode.

Example:

```
Device(config)# l2vpn vfi context VPLS-10
```

Step 2 Use the **vpn id** *vpn id* command to set a VPN ID on the VPLS instance.

Example:

```
Device(config-vfi)# vpn id 10
```

- Use the same VPN ID for the PE devices that belong to the same VPN.
- Make sure the VPN ID is unique for each VPN in the service provider network. The range is from 1 to 4294967295.

Step 3 Use the **member** *ip address* **encapsulation mpls** command to specify the device that forms a point-to-point L2VPN VFI connection.

Example:

```
Device(config-vfi)# member 102.102.102.102 encapsulation mpls
```

- ip-address : IP address of the VFI neighbor (the N-PE device).
- encapsulation mpls: specifies MPLS as the data encapsulation method.

Step 4 Use the **exit** command to return to global configuration mode.

Example:

```
Device(config-vfi)# exit
```

Step 5 Use the **bridge domain** *bridge domain* command to configure components on a bridge domain, and enter bridge-domain configuration mode.

Example:

```
Device(config)# bridge domain 10
```

Step 6 Use the **member vfi** *vfi-name* command to configure the VFI member in the bridge-domain.

Example:

```
Device(config-bdomain)# member vfi VPLS-10
```

Step 7 Use the **member** *ip-address vc-id* **encapsulation mpls** command to specify the device that forms a point-to-point L2VPN VFI connection.

Example:

```
Device(config-vfi)# member 105.105.105.105 10  encapsulation mpls
```

- *ip-address*: IP address of the VFI neighbor (U-PE device).
- *vc-id*: Virtual circuit identifier.
- **encapsulation mpls**: Specifies MPLSas the data encapsulation method.

Specify the N-PE devices that form the layer 2 VPN cross connection with the U-PE

Before you begin

Ensure that the device is in configuration mode.

Procedure

Step 1 Use the **interface** *type number* command to specify the interface to configure, and enter interface configuration mode.

Example:

```
Device(config)# interface GigabitEthernet0/0/0
```

Step 2 Use the **service instance** *id* **ethernet** command to configure an ethernet service instance on the interface, and enter ethernet service configuration mode.

Example:

```
Device(config-if)# service instance 10 ethernet
```

Step 3 Use the **encapsulation untagged** command define the criteria to map tagged or untagged frame ingress on the interface to the appropriate service instance.

Example:

```
Device(config-if-srv)# encapsulation untagged
```

Step 4 Use the **exit** command to return to interface configuration mode.

Example:

```
Device(config-if-srv)# exit
```

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

Example:

```
Device(config-if)# exit
```

Step 6 Use the **l2vpn xconnect context** *context-name* to create a L2VPN cross connect context, and enter xconnect configuration mode.

Example:

```
Device(config)# l2vpn xconnect context XC-10
```

Step 7 Use the **member gigabitethernet** *interface-number* [**service-instance** *id*] command to specify devices that form a L2VPN cross connect.

Example:

```
Device(config-xconnect)# member GigabitEthernet0/0/0 service-instance 10
```

service-instance *id*: (Optional) Specifies the service instance identifier.

Step 8 Use the **member** *ip-address vc-id* **encapsulation mpls** [**group** *group-name* [**priority** *number*] command to specify devices that form a L2VPN cross connect.

Example:

```
Device(config-xconnect)# member 101.101.101.101 10 encapsulation mpls group pwred priority 9

Device(config-xconnect)# member 102.102.102.102 10 encapsulation mpls group pwred priority 10
```

- *ip-address*—IP address of the peer N-PE device.
- *vc-id*—Virtual circuit identifier.
- **encapsulation mpls**—Specifies Multiprotocol Label Switching (MPLS) as the data encapsulation method.
- **group** *group-name*—Specifies the cross-connect member redundancy group name.

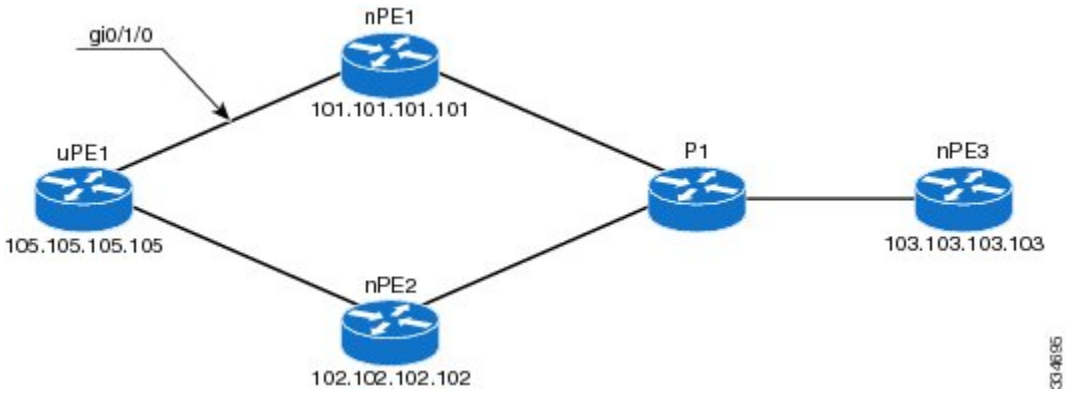
- **priority** *number*—Specifies the cross-connect member priority. The range is from 0 to 16. The highest priority is 0. Lowest priority is 16.

Configuration Example for H-VPLS N-PE Redundancy for MPLS Access

Example of configuring H-VPLS N-PE redundancy for MPLS access

The figure displays a configuration for H-VPLS N-PE redundancy with MPLS access. There is no option to configure multihoming on access VPLS, so the **xconnect** command with priority is used on uPE1.

Figure 88: H-VPLS N-PE Redundancy with MPLS Access Topology



nPE1 Configuration

```
l2vpn vfi context VPLS-10
 vpn id 10
 member 102.102.102.102 encapsulation mpls
 member 103.103.103.103 encapsulation mpls
!
bridge-domain 10
 member vfi VPLS-10
 member 105.105.105.105 10 encapsulation mpls
```

nPE2 Configuration

```
l2vpn vfi context VPLS-10
 vpn id 10
 member 101.101.101.101 encapsulation mpls
 member 103.103.103.103 encapsulation mpls
!
bridge-domain 10
 member vfi VPLS-10
 member 105.105.105.105 10 encapsulation mpls
```

nPE3 Configuration

```
l2vpn vfi context VPLS-10
 vpn id 10
 member 101.101.101.101 encapsulation mpls
 member 102.102.102.102 encapsulation mpls
```

```
!
bridge-domain 10
 member vfi VPLS-10
```

uPE1 Configuration

```
interface GigabitEthernet0/1/0
 service instance 10 ethernet
 encapsulation dot1q 10
!
l2vpn xconnect context XC-10
 member GigabitEthernet0/1/0 service-instance 10
 member 101.101.101.101 10 encapsulation mpls group pwred priority 9
 member 102.102.102.102 10 encapsulation mpls group pwred priority 10
```

Verification

Use the show commands to verify that H-VPLS N-PE redundancy for MPLS access is configured successfully.

Sample Output on uPE1

```
Device# show xconnect peer 101.101.101.101 vcid 10

Legend:    XC ST=Xconnect State  S1=Segment1 State  S2=Segment2 State
  UP=Up       DN=Down            AD=Admin Down      IA=Inactive
  SB=Standby  HS=Hot Standby     RV=Recovering      NH=No Hardware

XC ST  Segment 1                         S1 Segment 2                          S2
------+---------------------------------+--+---------------------------------+--
UP pri   ac Gi0/1/0:10(Eth VLAN)         UP mpls 101.101.101.101:10            UP

Device# show xconnect peer 102.102.102.102 vcid 10

Legend:    XC ST=Xconnect State  S1=Segment1 State  S2=Segment2 State
  UP=Up       DN=Down            AD=Admin Down      IA=Inactive
  SB=Standby  HS=Hot Standby     RV=Recovering      NH=No Hardware

XC ST  Segment 1                         S1 Segment 2                          S2
------+---------------------------------+--+---------------------------------+--
IA pri   ac Gi0/1/0:10(Eth VLAN)         UP mpls 102.102.102.102:10            SB
Device#
```

Pseudowire types for H-VPLS

This section describes how the IR1101 router determines Pseudowire (PW) types automatically based on the interface configuration and how to manually override these types to ensure compatibility with far-end devices.

Understanding Pseudowire types

The IR1101 router supports two primary Pseudowire types for H-VPLS:

- Type 4 (VLAN): Used for Ethernet VLAN interworking.
- Type 5 (Ethernet): Used for raw Ethernet interworking.

Automatic PW type selection

The router automatically selects the PW type based on the type of interface where the `xconnect` is configured.

Table 35: Automatic PW type selection

Interface Type	Configuration Method	Default PW Type
Layer 3 Interfaces (Onboard Gig, SP, SPMI, or 4S8I modules)	`xconnect` attached directly to the interface or sub-interface.	Type 5 (Ethernet)
Layer 2 Interfaces (Onboard FastEthernet (Fa) or 4A2T modules)	`xconnect` attached to a Switched Virtual Interface (SVI).	Type 4 (VLAN)

Limitations

These are the limitations for selecting Pseudowire types:

- Compatibility Check: Before configuring H-VPLS, verify the PW type used by the far-end device. If the IR1101 and the remote peer use mismatched types (one using Type 4 and the other Type 5), the connection will fail.
- SVI Limitation: Configuring an `xconnect` on an SVI automatically forces the router into Type 4 mode.
- FastEthernet Support: Onboard FastEthernet (Fa) interfaces are Layer 2 only and require an SVI for H-VPLS configuration.
- Cisco IOS XE 26.1.1 Cosmetic Issue: In Cisco IOS XE 26.1.1 release, the configuration parser may incorrectly allow you to apply an `xconnect` directly to a FastEthernet interface. This configuration is not supported and will not function. This issue is resolved in Cisco IOS XE 26.1.1 (or later releases).
- Always use an SVI for FastEthernet interfaces.

Overriding Pseudowire types

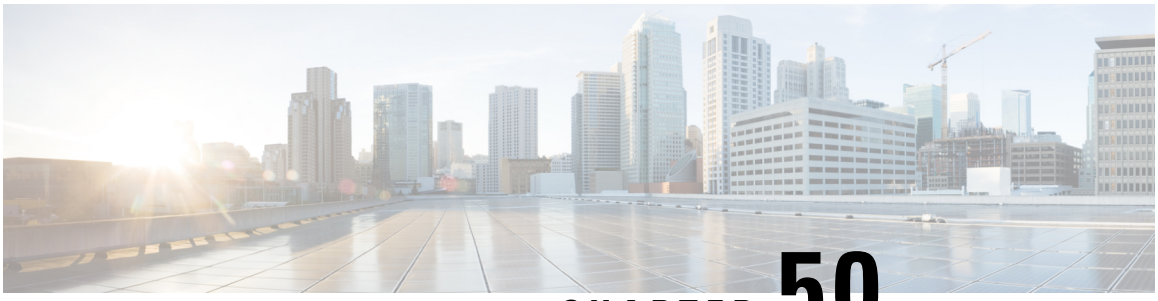
If the far-end device requires a specific PW type that does not match the IR1101 default, you can manually override the setting using an Layer 2 Virtual Private Network (L2VPN) cross-connect profile.

Here is an example of forcing Type 5 on an SVI:

```
Router(config-pw-class)# encapsulation mpls
Router(config-pw-class)# interworking ethernet
Router(config-pw-class)# exit
Router(config)# interface vlan 100
Router(config-if)# xconnect 10.1.1.1 123 pw-class PW_TYPE_5_PROFILE
Router(config-if)# end
```

To verify the current PW type use the **show l2vpn atom vc** [*vcid*/*interface-id*]**detail** command.

Note You can look for the local interface type or PW type fields in the output to confirm if it is Type 4 or Type 5.



CHAPTER 50

Support for 20,000 Routers

Release 26.1.x introduces updated computing resource recommendations for Cisco Catalyst SD-WAN Control components. The specifications support Industrial IoT deployments with up to 20,000 devices, which ensures stability, high availability, and optimal performance across the control plane.

These tables outline the instance type definitions, component resource recommendations, and testbed specifications.

Table 36: Instance type definitions

Instance type	Specifications (Approximation)			Qualified Instance Type	
vCPUs*	RAM*	Storage Size*	Azure	AWS	
Large	32	128 GB	5 TB	Not qualified	Not qualified

*vCPU, RAM, and Storage Size values are per Cisco Catalyst SD-WAN Manager.

Table 37: Instance types with number of devices, nodes and deployment models

Devices	Nodes and Deployment Models with Instance Type	Data Processing Factor	Number of days the data can be stored	Max Daily Processing Volume	Cisco Cloud	On-Prem (UCS)	Customer Cloud
20000	A six-node large Cisco Catalyst SD-WAN Manager cluster (with three nodes running ConfigDB), where all nodes serve as messaging server, statistics processor, and AppServer	100 GB per day	14	500 GB per day*	Not Qualified	Yes	Not Qualified

Table 38: Cisco Catalyst SD-WAN validator recommended computing resources

Devices	Number of Cisco SD-WAN Validators	vCPU	RAM	OS Volume	vNICs	Azure	AWS
20000	6	4	8 GB	20 GB	2 (one for the tunnel interface and one for management)	Not qualified	Not qualified

Table 39: Cisco Catalyst SD-WAN controller recommended computing resources

Devices	Number of Cisco SD-WAN Controllers	vCPU	RAM	OS Volume	vNICs	Azure	AWS
20000	16	8	16 GB	20 GB	2 (one for tunnel interface, one for management)	Not qualified	Not qualified

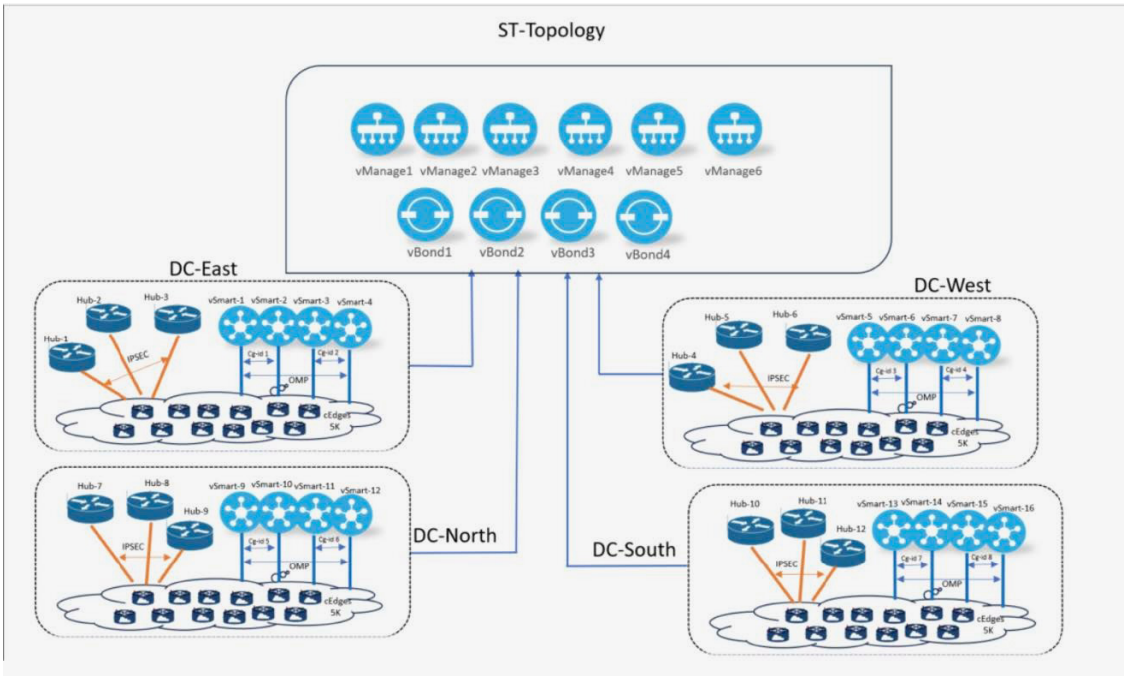
Table 40: UCS specifications

Hardware SKU	Specifications
UCSC-C240-M5SX	UCS C240 M5 24 SFF + 2 rear drives without CPU, memory cards, hard disk, PCIe, and PS.
UCS-MR-X16G1RT-H	16GB DDR4-2933-MHz RDIMM/1Rx4/1.2v
UCS-CPU-I6248R	Intel 6248R 3GHz/205W 24C/35.75MB DDR4 2933MHz Intel(R) Xeon(R) Gold 6330N CPU @ 2.20GHz
UCS-SD16T123X-EP	1.6TB 2.5in Enterprise Performance 12G SAS SSD (3X endurance)

Note

- Cisco Catalyst SD-WAN Controllers are supported on UCS platforms (fifth generation, sixth generation, and later) that meet or exceed the specified hardware, enabling similar scale numbers.
- Both AMD and Intel CPUs support the specified UCS specifications.

Figure 89: Cisco Catalyst SD-WAN test bed topology diagram for 20,000 devices



Considerations for topology testing

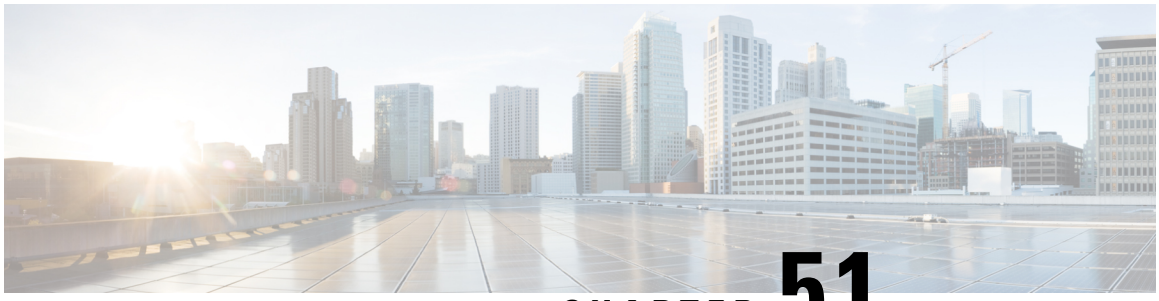
- The test bed features a vManage cluster with 6 nodes, 16 controllers, and 4 validators.
- There are 20,000 devices, divided evenly among the East, West, North, and South zones.
- Each zone uses c8kv devices as hubs.
- In each zone, 5,000 devices connect to these hubs, forming a Hub and Spoke topology with a single TLOC.

Key performance and scale metrics

1. vManage/Controllers/Validators stability, High Availability, and characterization.
2. UI characterization.
3. Bulk API with concurrent sessions (tested for various API calls).
4. DPI traffic for 20,000 cEdge devices with aggregated daily traffic up to 500 GB.
5. Configuration scale up to 6000 lines.
6. Policy scale up to 32,000 on 16 vSmarts.
7. OMP Routes up to 250,000.
8. BFD flap on 1000 devices and alarm updates on vManage.

Note

- Multi-tenant scale testing is not supported.
- Controller stability, High Availability, API response time, and UI performance are validated for up to 20,000 devices, consistent with enterprise-level testing.
- Points 4, 5, 6, and 7 (from the "Key performance and scale metrics" section) specifically address IIoT requirements.



CHAPTER 51

Segment Routing over IPv6

- Segment Routing over IPv6, on page 525
- Prerequisites, on page 526
- Configure SRv6, on page 526
- Configure SRv6 under IS-IS, on page 527
- Verify SRv6, on page 528

Segment Routing over IPv6

Segment Routing over IPv6 (SRv6) in Cisco IoT is a technology that enables source routing of IPv6 traffic by encoding a list of segments or instructions in the IPv6 packet header. This capability allows the source router to define the path that the packet must take through the network. Each segment is represented by an IPv6 address called a Segment Identifier (SID), and the packet carries a Segment Routing Header (SRH) that lists these segments.

Benefits of using SRv6:

- Operates over the IPv6 data plane and uses the SRH to encode an ordered list of segments that the packet must traverse.
- Enables network nodes to forward packets based on the active segment present in the destination address field and updates the destination address to the next segment as the packet progresses.
- Supports network programming by allowing each segment to execute at a node level.
- Simplifies network operations by removing the need for additional protocol layers like MPLS, which is beneficial for IoT environments requiring scalable and efficient routing.
- Helps enhance reliability and performance.
- Used in programmable, scalable, and automated networks, including IoT deployments, by providing end-to-end solutions.

Table 41: Feature History Table

Feature name	Release information	Feature description
Segment Routing over IPv6	Release 26.1.1a	This feature enables source routing of IPv6 traffic by encoding a list of segments or instructions in the IPv6 packet header. This allows the source router to define the path that the packet must take through the network.

Prerequisites

Cisco IOS XE supports micro-Segment Identifier (uSID) with 32-bit uSID block and 16-bit uSID IDs (3216). This format must be used for uSID locators in a SRv6 uSID domain.

Configure SRv6

Follow this procedure to configure SRv6.

Procedure

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

Example:

```
Router#configure terminal
```

Step 2 Use the **segment-routing srv6** command to enable SRv6.

Example:

```
Router(config)# segment-routing srv6
```

Step 3 Use the **locators** command to enable the locators for SRv6.

Example:

```
Router(config-srv6)# locators
```

Step 4 Use the **locator** [*locator-name*] command for assigning the locator for SRv6.

Example:

```
Router(config-srv6-locators)# locator locapoke3
```

Step 5 Use the **prefix** [*subnet-address*] command to specify IPv6 prefix for the locator.

Example:

```
Router(config-srv6-locs-locator)# prefix 2001:0:5::/48
```

Step 6 Use the **format** [*format-id*] command to specify the IS-IS level for SRv6 locator.

Example:

```
Router(config-srv6-locs-locator)# format usid-f3216
```

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

Example:

```
Router(config-srv6-locators)# end
```

What to do next

Verify SRv6 configuration.

Configure SRv6 under IS-IS

Follow this procedure to enable SRv6 under IS-IS IPv6 address family.

Intermediate System-to-Intermediate System (IS-IS) protocol supports segment routing with IPv6 data plane.

SRv6 as part of IS-IS:

- Interacts with SID Manager to learn local locator prefixes and announces the locator prefixes in the Interior Gateway Protocol (IGP) domain.
- Learns remote locator prefixes from other IS-IS neighbor routers and installs the learned remote locator IPv6 prefix in Routing Information Base (RIB).
- Allocates or learns prefix SID and adjacency SIDs, creates local SID entries, and advertises them in the IGP domain.

Procedure

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

Example:

```
Router#configure terminal
```

Step 2 Use the **router isis core** command to enable and configure the IS-IS routing protocol on the router.

Example:

```
Router(config)# router isis core
```

a) Use the **net** [*network-area-address-id*] command to define the area address of IS-IS.

Example:

```
Router(config-router)# net 49.0000.0000.0000.0005.00
```

b) Use the **metric-style wide** command to generate the new-style Type-Length-Value (TLV).

Example:

```
Router(config-router)# metric-style wide
```

c) Use the **lsp-mtu** [*bytes*] command to set the maximum size of link state packets.

Example:

```
Router(config-router)# lsp-mtu 1400
```

Step 3 Use the **address-family ipv6** command to enter the IPv6 unicast address family configuration mode.

Example:

```
Router(config-router)# address-family ipv6
```

Step 4 Use the **router-id Loopback0** command to enable loopback.

Example:

```
Router(config-router-af)# router-id Loopback0
```

Step 5 Use the **segment-routing srv6** command to enable SRv6.

Example:

```
Router(config-router-af)# segment-routing srv6
```

Step 6 Use the **locator** [*locator-name*] command for enabling the locator for SRv6.

Example:

```
Router(config-isis-af-srv6)# locator locapoke3
```

Step 7 Use the **level 1** command to specify the IS-IS level for SRv6 locator.

Example:

```
Router(config-isis-af-srv6-locator)# level 1
```

Step 8 Use the **end** command to save the configuration and exit the privileged EXEC mode.

Example:

```
Router(config-isis-srv6-locator)# end
```

Verify SRv6

Follow this procedure to verify SRv6 configuration.

Procedure

Step 1 Use the **show segment-routing srv6 locator** command to verify SRv6 configuration on the router.

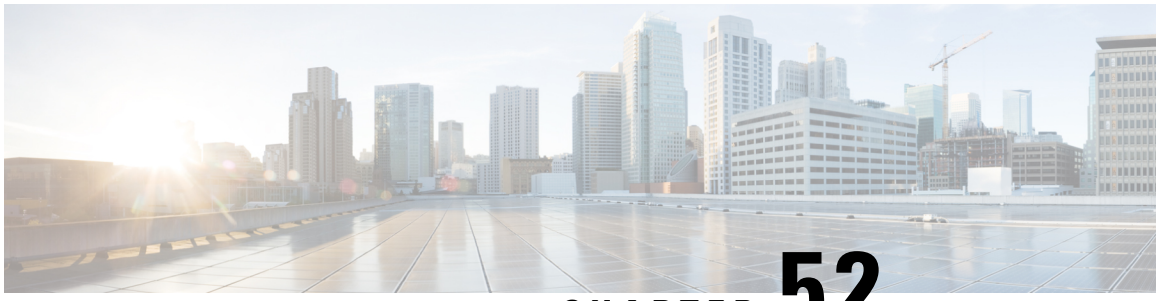
Example:

```
router# show segment-routing srv6 locator
Name            Algo     Prefix                    Format              Status
----            ----     ------                    ------              ------
locapoke3       0        2001:0:5::/48             usid-f3216          Up
```

Step 2 Use the **show segment-routing srv6 sid** command to display the SRv6 segment identifiers on the router.

Example:

```
router# show segment-routing srv6 sid
SID                                     Locator         Behavior             Context
                Owner
---                                     -------         --------             -------
                -----
2001:0:5::                              locapoke3       uN (PSP/USD)
                SID-MGR
2001:0:5:E001::                        locapoke3      uA (PSP/USD)         Tu2 FE80::6A79:9FF:FEA9:A800
             router isis core
```



CHAPTER 52

Quantum-Safe Encryption using Postquantum Preshared Keys

- Quantum-Safe Encryption with Postquantum Preshared Keys, on page 531
- Postquantum Preshared Key restrictions, on page 534
- Supported platforms, on page 534
- Configure manual Postquantum Preshared Keys, on page 534
- Configure dynamic Postquantum Preshared Keys, on page 535
- Configuration examples for Quantum-Safe Encryption Using Postquantum Preshared Keys, on page 536
- Postquantum Preshared Key configuration examples, on page 537
- Verify the Postquantum Preshared Key configuration, on page 539

Quantum-Safe Encryption with Postquantum Preshared Keys

A postquantum preshared key (PPK) is a preshared secret that IKEv2 mixes into session-key derivation to provide resistance to attacks by a future quantum computer.

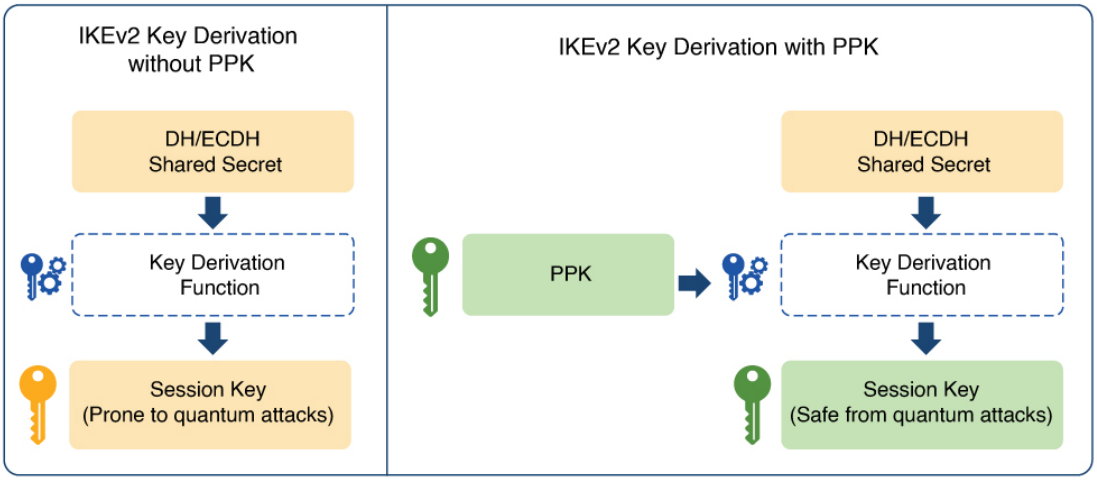
- The feature implements RFC 8784 extensions for IKEv2 and uses Cisco Secure Key Integration Protocol (SKIP) to obtain dynamic PPKs.
- Session keys based on PPKs are not vulnerable to quantum attacks if the PPKs have sufficient entropy and the pseudorandom function (PRF), encryption, and authentication transformations are quantum secure.

Postquantum Preshared Keys

Quantum computers pose a serious challenge to the cryptographic algorithms and protocols deployed widely today. A quantum computer can solve Diffie-Hellman (DH) and Elliptic Curve Diffie-Hellman (ECDH) problems in polynomial time, and this can compromise the security of existing IKEv2 systems. A man-in-the-middle storing the VPN communications today can decrypt them later when a quantum computer is available.

RFC 8784 defines PPK capability negotiation, communication of the PPK ID, use of the PPK as an additional input to session-key derivation, and optional fallback to a session that does not use a PPK.

Figure 90: IKEv2 Key Derivation - With and Without PPK



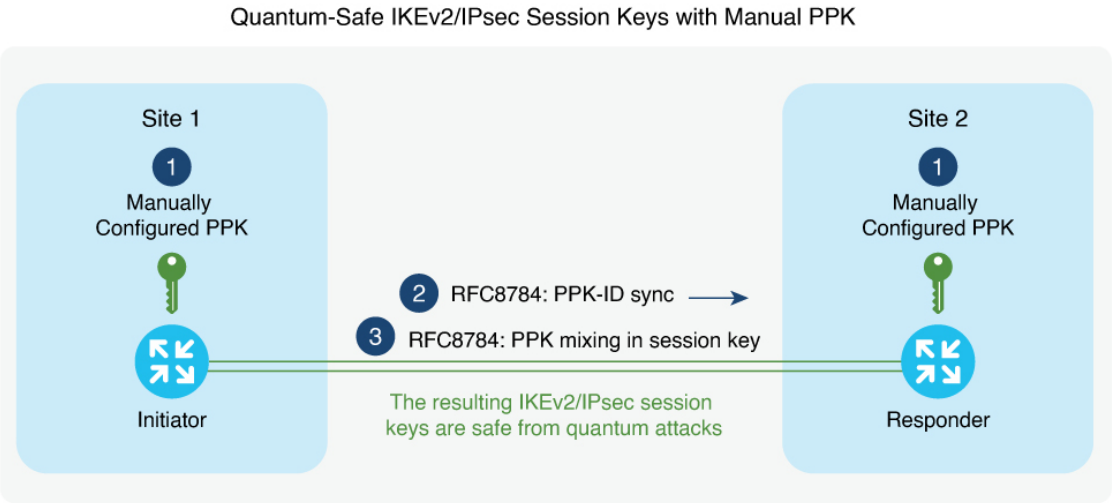
DH: Diffie-Hellman
ECDH: Elliptic-curve Diffie-Hellman
PPK: Postquantum Preshared Key

Manual Postquantum Preshared Keys

A manual PPK is configured on both the IKEv2 initiator and responder. The PPK ID and PPK must match on both peers.

The administrator must ensure that a manual PPK has sufficient size and entropy and is rotated frequently.

Figure 91: Quantum-Safe IKEv2 and IPsec Session Keys with Manual PPK



Dynamic Postquantum Preshared Keys and Cisco Secure Key Integration Protocol

A dynamic PPK is imported from an external key source through SKIP. Dynamic PPKs provide automated provisioning and refresh and can provide better entropy than manually provisioned keys.

Cisco SKIP is an HTTPS-based protocol that uses TLS1.2 with a PSK-DHE cipher suite. It allows encryption devices such as routers, to import PPKs from an external key source.

The IR1101 router must implement the SKIP client, and the external key source must implement the SKIP server.

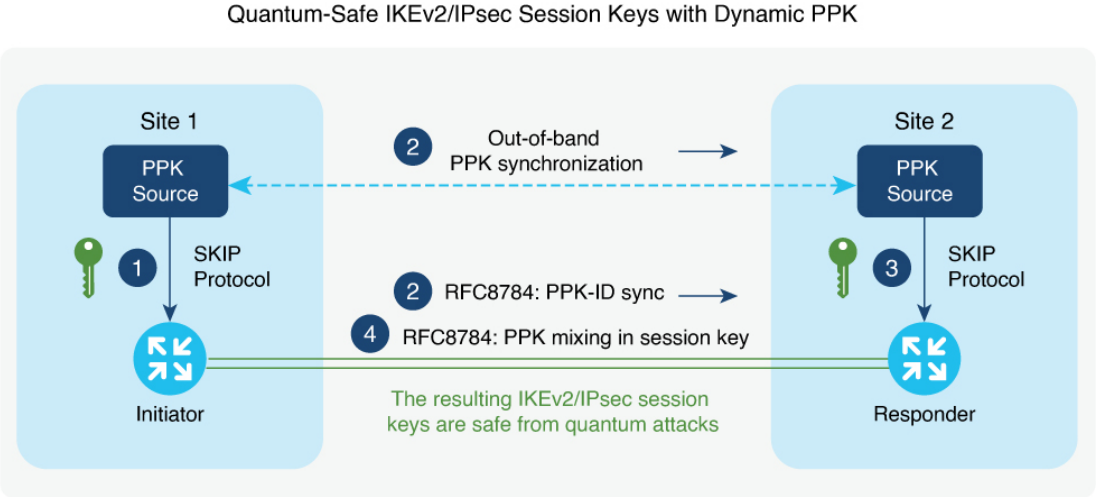
The external key source must meet the following expectations to be SKIP-compliant:

- Must implement SKIP protocol or API, as defined in the Cisco SKIP specification.
- Must provide the same PPK to the encryption device pair—initiator and responder—using an out-of-band synchronization mechanism.

Note

- Key source vendors, such as QKD vendors, should contact their Cisco representative to implement the Cisco SKIP protocol.

Figure 92: Quantum-Safe IKEv2 and IPsec Session Keys with Dynamic PPK



The IKEv2 initiator and responder are connected to their local key source and configured with the SKIP client that specifies the IP address and port of the key source and the preshared key for the TLS1.2 session. The PPK sources are configured with the SKIP parameters, including the local key source identity and the list of identities of the peer key sources.

Cisco SKIP protocol operates as given:

- The IKEv2 initiator places a request for a PPK from its key source. The key source replies with a PPK and the corresponding PPK ID.
- The initiator-side key source synchronizes the PPK to the responder-side key source using an out-of-band mechanism that is specific to the type of key source. The IKEv2 initiator communicates the PPK ID to the IKEv2 responder over IKEv2 using the RFC 8784 extensions.
- The IKEv2 responder requests from its key source, the PPK corresponding to the PPK ID received from the IKEv2 initiator. The key source replies with the PPK corresponding to the PPK ID.

- The IKEv2 initiator and responder mix the PPK in the key derivation, as specified in RFC 8784. The resulting IKEv2 and IPsec session keys are quantum-safe.

Postquantum Preshared Key restrictions

Consider these restrictions when you plan manual or dynamic PPK configuration.

Table 42: PPK restrictions

Condition	Restriction
VPN type	PPK-based quantum-safe encryption applies to IKEv2 and IPsec VPNs, including FlexVPN with SVTI or DVTI and DMVPN. It does not apply to GETVPN.
Dynamic PPK key format	When you configure a PPK through SKIP, the external key source must send the key in hexadecimal format.

Supported platforms

The Quantum-Safe Encryption Using Postquantum Preshared Keys feature is available on Cisco Catalyst IR1101 Rugged Series Router from Cisco IOS XE Release 17.18.1 onwards.

Configure manual Postquantum Preshared Keys

Before you begin

Note The PPK ID and PPK must match on the initiator and responder.

Configure a manual PPK for one peer or a group of peers in an IKEv2 keyring. Use these steps to configure manual postquantum preshared keys:

Procedure

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

Example:

```
Router# configure terminal
```

The router enters global configuration mode.

Step 2 Use the **crypto ikev2 keyring** *keyring-name* command to create an IKEv2 keyring and enter IKEv2 keyring configuration mode.

Example:

```
Router(config)# crypto ikev2 keyring keyring1
```

Step 3 Use the **peer** *name* command to define a peer or peer group and enter IKEv2 keyring peer configuration mode.

Example:

```
Router(config-ikev2-keyring)# peer peer1
```

Note
The WAN IP address is the IKE endpoint address and is independent of the identity address. The **identity** command is available for key lookup only on the IKEv2 responder.

Step 4 Use either the **address** {*ipv4-address mask* | *ipv6-address prefix*} command or the **identity** {**address** {*ipv4-address* | *ipv6-address*} | **fqdn domain** *domain-name* | **email domain** *domain-name* | **key-id** *key-id*} command to identify the remote IKEv2 peer.

The **address** command identifies an IPv4 or IPv6 address or range. The **identity** command identifies a peer by e-mail address, fully qualified domain name, IPv4 or IPv6 address, or key ID.

```
Router(config-ikev2-keyring-peer)# address 10.0.0.1 255.0.0.0
Router(config-ikev2-keyring-peer)# identity address 10.0.0.1
```

Step 5 Use the **ppk manual id** *ppk-id* **key** [**0** | **6** | **hex**] *password* [**required**] command to configure the PPK ID and PPK.

The *ppk-id* identifies the PPK, and *password* specifies the PPK. Include **required** when PPK-based quantum-safe encryption is mandatory and the peers must not fall back to a normal IKEv2 or IPsec session.

```
Router(config-ikev2-keyring-peer)# ppk manual id ppk_id key cisco123
```

Step 6 Use the **crypto ikev2 profile** *profile-name* command to create an IKEv2 profile and enter IKEv2 profile configuration mode.

Example:

```
Router(config)# crypto ikev2 profile profile1
```

Step 7 Use the **keyring ppk** *keyring-name* command to associate the PPK keyring with the IKEv2 profile.

Example:

```
Router(config-ikev2-profile)# keyring ppk keyring1
```

Note
To remove the keyring from the IKEv2 profile, use the **no keyring** {**aaa** | **local** | **ppk**} *keyring-name* command.

Step 8 Use the **end** command to return to privileged EXEC mode

Example:

```
Router(config-ikev2-profile)# end
```

Configure dynamic Postquantum Preshared Keys

Before you begin

Note On the responder, when multiple SKIP client instances are configured, only the first SKIP client entry in the configuration list is used. Map the **skip-client** configuration under the keyring to the first SKIP client block.

Configure the parameters that the IR1101 router uses to request dynamic PPKs from an external SKIP-compliant key source. A SKIP client configuration identifies the external key source and the preshared key used for the TLS 1.2 session. Use these steps to configure dynamic postquantum preshared keys:

Procedure

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

Example:

```
Router# configure terminal
```

The router enters global configuration mode.

Step 2 Use the **crypto skip-client** *skip-client-name* command to create a SKIP client configuration block and enter SKIP client configuration mode.

Example:

```
Router(config)# crypto skip-client skip-client-cfg
```

Step 3 Use the **server** {**ipv4** *ipv4-address* | **ipv6** *ipv6-address* | **fqdn** *domain-name*} **port** *port-number* command to specify the external key source.

The command identifies the IPv4 address, IPv6 address, or FQDN and port used to connect to the external key source.

```
Router(config-crypto-skip-client)# server ipv4 10.10.0.3 port 9993
```

Step 4 Use the **psk id** *id-name* **key** [**0** | **6** | **hex**] *password* command to configure the preshared key identity and key for the SKIP TLS session.

Example:

```
Router(config-crypto-skip-client)# psk id psk-id key 0 cisco123
```

Step 5 Use the **end** command to return to privileged EXEC mode.

Example:

```
Router(config-crypto-skip-client)# end
```

Configuration examples for Quantum-Safe Encryption Using Postquantum Preshared Keys

The following sections provide detailed configuration examples relating to the configuration of quantum-safe encryption using PPKs..

Postquantum Preshared Key configuration examples

The following sections provide initiator and responder configuration examples for manual and dynamic PPKs.

The manual PPK ID and PPK must match on both peers. Replace `<ppk-value>` with the same PPK on the initiator and responder.

Manual PPK initiator configuration

This example configures a manual PPK on the initiator.

```
conf t
hostname Router1
!
crypto ikev2 keyring ppk-keyring
 peer 1
  address 10.10.0.1 255.255.255.0
  ppk manual id ppk_id key <ppk-value>
!
crypto ikev2 profile prof
 match identity remote address 10.10.0.1
 authentication local pre-share key cisco
 authentication remote pre-share key cisco
 keyring ppk ppk-keyring
!
crypto ipsec profile prof
 set ikev2-profile prof
!
interface Tunnel0
 ip address 10.10.0.1 255.255.255.0
 tunnel source GigabitEthernet1
 tunnel destination 10.10.10.1
 tunnel protection ipsec profile prof
!
interface GigabitEthernet1
 ip address 10.10.10.2 255.255.255.0
 no shut
!
```

Manual PPK responder configuration

This example configures the matching manual PPK on the responder.

```
conf t
hostname Router2
!
crypto ikev2 keyring ppk-keyring
 peer 1
  address 10.10.0.1 255.255.255.0
  ppk manual id ppk_id key <ppk-value>
!
crypto ikev2 profile prof
 match identity remote address 10.10.0.1
 authentication local pre-share key cisco
 authentication remote pre-share key cisco
 keyring ppk ppk-keyring
!
crypto ipsec profile prof
 set ikev2-profile prof
!
interface Tunnel0
```

```
 ip address 10.10.0.2 255.255.255.0
 tunnel source GigabitEthernet1
 tunnel destination 10.10.10.2
 tunnel protection ipsec profile prof
!
interface GigabitEthernet1
 ip address 10.10.0.1 255.255.255.0
 no shut
!
```

Dynamic PPK initiator configuration

This example configures a SKIP client and dynamic PPK on the initiator.

```
conf t
hostname Router1
!
crypto skip-client skip-client-cfg
 server ipv4 10.10.0.4 port 9991
 psk id psk-id1 key 0 cisco123
!
crypto ikev2 keyring ppk-keyring
 peer 1
  address 10.10.0.1 255.255.255.0
  ppk dynamic skip-client-cfg
!
crypto ikev2 profile prof
 match identity remote address 10.10.0.1
 authentication local pre-share key cisco
 authentication remote pre-share key cisco
 keyring ppk ppk-keyring
!
crypto ipsec profile prof
 set ikev2-profile prof
!
interface Tunnel0
 ip address 10.10.0.2 255.255.255.0
 tunnel source GigabitEthernet1
 tunnel destination 10.10.10.1
 tunnel protection ipsec profile prof
!
interface GigabitEthernet1
 ip address 10.10.10.2 255.255.255.0
 no shut
!
interface GigabitEthernet1
 ip address 10.10.10.3 255.255.255.0
 no shut
!
```

Dynamic PPK responder configuration

This example configures a SKIP client and dynamic PPK on the responder.

```
conf t
hostname Router2
!
crypto skip-client skip-client-cfg
 server ipv4 10.10.0.4 port 9992
 psk id vedge-sim-1 key 0 cisco123
!
crypto ikev2 keyring ppk-keyring
 peer 1
  address 10.10.0.1 255.255.255.0
```

```
  ppk dynamic skip-client-cfg
!
crypto ikev2 profile prof
 match identity remote address 10.10.0.1
 authentication local pre-share key cisco
 authentication remote pre-share key cisco
 keyring ppk ppk-keyring
!
crypto ipsec profile prof
 set ikev2-profile prof
!
interface Tunnel0
 ip address 10.10.0.2 255.255.255.0
 tunnel source GigabitEthernet1
 tunnel destination 10.10.10.2
 tunnel protection ipsec profile prof
!
interface GigabitEthernet1
 ip address 10.10.10.1 255.255.255.0
 no shut
!
interface GigabitEthernet1
 ip address 10.10.10.4 255.255.255.0
!
```

Verify the Postquantum Preshared Key configuration

Use these steps to verify the postquantum preshared key configuration:

Procedure

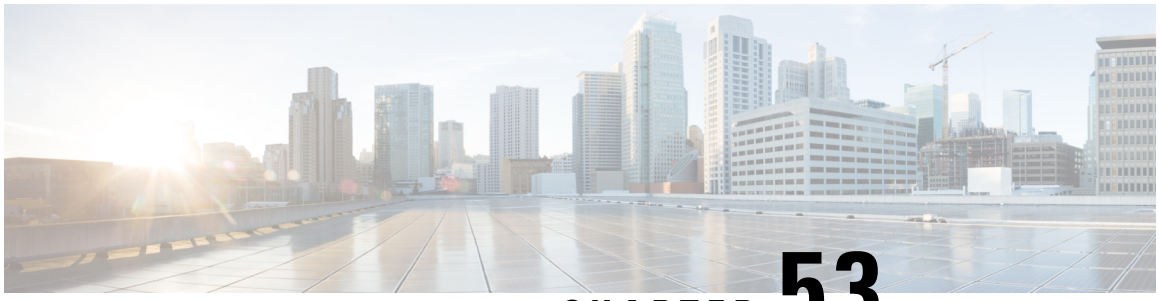
Step 1 Use the **show crypto ikev2 sa detailed** command to display detailed information about the current IKEv2 security associations.

The command displays details for the current IKEv2 security associations.

```
Device# show crypto ikev2 sa detailed
IPv4 Crypto IKEv2 SA
Tunnel-id  Local             Remote            fvrf/ivrf  Status
3          <src IP>/SrcPort  <Dst IP>/DstPort  none/none  READY
Encr: AES-CBC, keysize: 256, PRF: SHA512, Hash: SHA512, DH Grp:19, Auth sign: . . .
Initiator of SA : No
Quantum Resistance Enabled
```

Step 2 Confirm that the output includes **Quantum Resistance Enabled**.

This message indicates that PPK-based quantum-safe encryption is enabled.



CHAPTER 53

DHCP Option 82

Table 43: 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 541
- Prerequisites, on page 542
- Key considerations, on page 543
- Limitations, on page 543
- DHCP Option 82 packet flow, on page 543
- Hexadecimal pattern and suboption decoding, on page 544
- Configure DHCP Option 82 on server, on page 545
- Configure DHCP address pool, on page 547
- Configure DHCP relay agent, on page 548
- Configure relay chaining on relay agents, on page 549
- Verify DHCP Option 82, on page 550
- Debug DHCP Option 82 logs, on page 551

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.

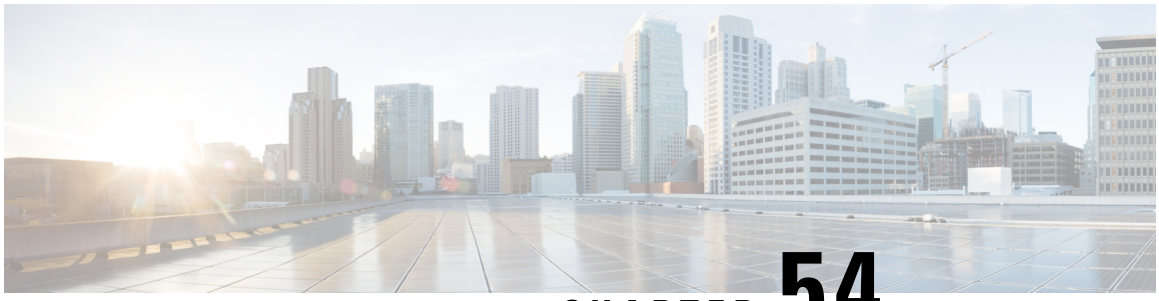
Example:

```
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
```



CHAPTER 54

IKEv2 and IPsec PFS Post Quantum Cryptography

Table 44: 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 553
- Benefits of PQC, on page 554
- Prerequisites, on page 554
- Limitations, on page 554
- ML-KEM algorithm and parameter sets, on page 555
- PFS negotiation behavior, on page 556
- Configure an IKEv2 proposal for PQC, on page 556
- Configure Perfect Forward Secrecy with PQC, on page 558
- Verify the PQC hybrid key exchange, on page 559
- Troubleshooting PQC hybrid key exchange, on page 559

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 45: 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 46: 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.

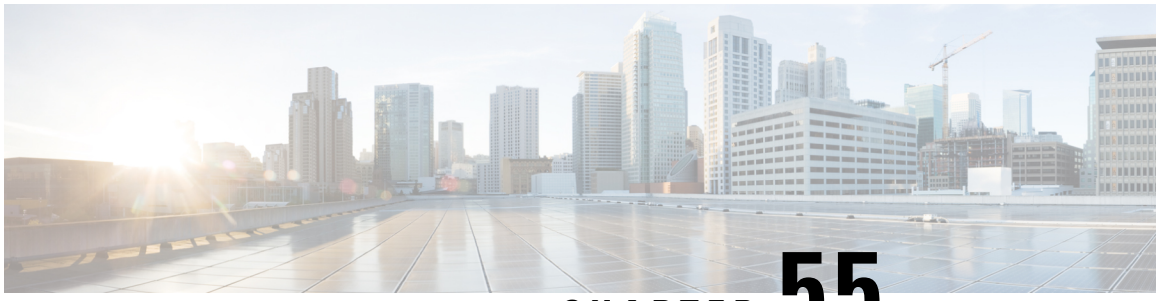
Example:

```
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 55

Post Quantum Cryptography for SSHv2 sessions

Table 47: 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 561
- Overview, on page 562
- Factors to consider for deploying PQC, on page 562
- Protocol requirements, on page 562
- Restrictions for PQC, on page 562
- PQC negotiation scenarios , on page 563
- Configure PQC for SSHv2 sessions, on page 563
- Verify PQC for SSHv2 sessions, on page 564
- Key elements of PQC, on page 564
- Supported platforms, on page 566

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 48: 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:

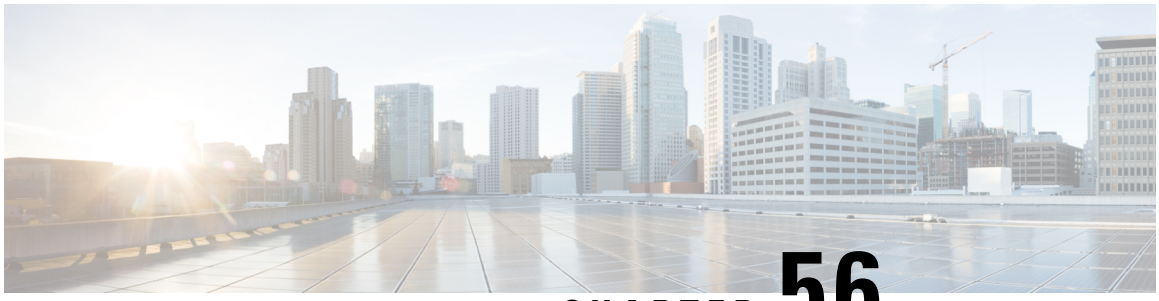
ML-KEM algorithm	What this algorithm means	Security Level
mlkem768x25519-sha256	ML-KEM 768 is a PQC algorithm based on module-lattice cryptography, providing resistance against quantum attacks. X25519 is a widely used classical elliptic curve Diffie-Hellman key exchange algorithm. SHA-256 is the hash 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 56

Segment Routing over MPLS

Table 49: Feature History Table

Feature name	Release information	Feature description
Segment Routing over MPLS	Release 26.2.1	Starting from Cisco IOS XE Release 26.2.1, you can configure Segment Routing over Multiprotocol Label Switching (SR-MPLS) on supported WAN ports. SR-MPLS encodes an ordered list of segments in the packet and integrates with MPLS services such as Layer 3 VPN and Ethernet VPN.

- Segment Routing over MPLS, on page 567
- Prerequisites, on page 568
- Configure Segment Routing over MPLS, on page 568
- Configure Segment Routing MPLS on OSPF network, on page 569
- Configure Prefix-SID for OSPF, on page 570
- Verify SR-MPLS configuration, on page 570

Segment Routing over MPLS

Segment Routing over Multi-protocol Label Switching (SR-MPLS) on WAN ports is a routing architecture that provides a flexible and scalable approach to source routing. It allows the source to define a path and encode it in the packet header as an ordered list of segments. Each segment is identified by a Segment ID (SID). When operating using the MPLS dataplane, segment routing integrates with the multi-service capabilities of MPLS, including Layer 3 VPN (L3VPN) and Ethernet VPN (EVPN).

Note SR-MPLS is applied specifically to the WAN ports of the supported routers. Ensure the WAN interfaces are active and configured with the appropriate routing protocols (OSPF or IS-IS) before proceeding.

Benefits of using SR-MPLS

SR-MPLS helps in:

- Scalability: The network does not need to maintain per-application or per-flow states.
- Traffic Engineering: Offers lower latency and more effective bandwidth utilization than traditional MPLS networks.
- Automatic Protection: Provides automatic traffic protection against link and node failures without requiring additional signaling.

Prerequisites

Before configuring SR-MPLS on your router, you must ensure that:

- Cisco Catalyst IR1101 or IR8340 routers with active WAN ports are used.
- The software image supports SR-MPLS configuration for these routers.
- Intermediate System to Intermediate System (IS-IS) and Open Shortest Path First (OSPF) routing protocols are enabled and configured for SR-MPLS operation.

Configure Segment Routing over MPLS

Use this task to enable SR-MPLS mode.

Procedure

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

Example:

```
Router# configure terminal
```

Step 2 Use the **segment-routing mpls** command to enable segment routing MPLS.

Example:

```
Router(config)# segment-routing mpls
```

Step 3 Use the **global-block** *start-range end-range* command to enable Segment Routing Global Block (SRGB) to define the label range for Prefix-SIDs.

Example:

```
Router(config)# global-block 16000 23999
```

For more information, see Segment Routing Global Block

Step 4 Use the **connected-prefix-sid-map** command to enter the sub-mode for configuring address-family specific mappings.

Example:

```
Router(config-srmpls)# connected-prefix-sid-map
```

Step 5 Use the **address-family** *ipv4* command to specify the IPv4 address family.

Example:

```
Router(config-srmpls-conn)# address-family ipv4
```

Step 6 Use the *ip-address/mask* **index** *sid-index* **range** *range-value* command to associate a Segment Identifier (SID) with the IP address.

Example:

```
Router(config-srmpls-conn-af)# 4.4.4.4/32 index 4 range 1
```

Step 7 Use the **exit-address-family** command to exit the address family configuration mode and return to the **connected-prefix-sid-map** sub-mode.

Example:

```
Router(config-srmpls-conn-af)# exit-address-family
```

Configure Segment Routing MPLS on OSPF network

Use this task to define the network topology, select OSPF as the control protocol, and configure the necessary SID mappings.

Procedure

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

Example:

```
Router# configure terminal
```

Step 2 Use the **router ospf** *process-id* command to enable the OSPF routing process.

Example:

```
Router(config)# router ospf 10
```

Step 3 Use the **router-id** *id* command to manually configure the OSPF router ID.

Example:

```
Router(config-router)# router-id 10.0.0.1
```

Step 4 Use the **segment-routing mpls** command to enable segment routing under the OSPF process.

Example:

```
Router(config-router)# segment-routing mpls
```

Step 5 Use the **segment-routing area** *area-id* **mpls** command to configure segment routing MPLS mode for a specific OSPF area.

Example:

```
Router(config-router)# segment-routing area 0 mpls
```

Configure Prefix-SID for OSPF

This task allows you to map a specific Prefix-SID to a loopback or interface address. Mapping the Prefix-SID to a loopback address creates a Node-SID, which serves as the node's unique identity. This enables other routers to forward traffic to this specific prefix using the associated SID label.

Before you begin

Ensure that segment routing is enabled on the corresponding address family.

Procedure

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

Example:

```
Router# configure terminal
```

Step 2 Use the **segment-routing mpls** command to enable segment routing MPLS mode.

Example:

```
Router(config)# segment-routing mpls
```

Step 3 Use the **connected-prefix-sid-map** command to enter the sub-mode for configuring address-family specific mappings.

Example:

```
Router(config-srmpls)# connected-prefix-sid-map
```

Step 4 Use the **address-family** *ipv4* command to specify the IPv4 address family.

Example:

```
Router(config-srmpls-conn)# address-family ipv4
```

Step 5 Use the *ip-address/mask* **index** *sid-index* **range** *range-value* command to associate a SID with the specified IPv4 prefix.

Example:

```
Router(config-srmpls-conn-af)# 10.1.1.1/32 index 100 range 1
```

Step 6 Use the **exit** command to exit to global configuration mode.

Example:

```
Router(config-srmpls-conn-af)# exit
```

Verify SR-MPLS configuration

Use this task to validate connectivity, path establishment, and resilience of your SR-MPLS deployment.

Before you begin

Verify the node SID reachability to ensure each node in the SR-MPLS domain can reach its designated node SID.

Procedure

Step 1 Use the **show segment-routing mpls connected-prefix-sid-map local ipv4** command to verify the SR-MPLS configuration on the router.

Example:

```
Router# show segment-routing mpls connected-prefix-sid-map local ipv4
```

Step 2 Use the **show mpls forwarding-table** command to check the MPLS label table.

Example:

```
Router# show mpls forwarding-table
```

Step 3 Use the **ping mpls** command to validate the end-to-end connectivity between critical devices.

Note

- You can simulate link or node failures to ensure traffic is automatically rerouted along available alternate paths.
- You can also test adjacency SID reachability between adjacent nodes.