Cisco Catalyst IR1800 Rugged Series Router Software Configuration Guide

First Published: 2020-08-13

Last Modified: 2022-12-13

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

© 2021–2022 Cisco Systems, Inc. All rights reserved.

CONTENTS

Full Cisco Trademarks with Software License ?

CHAPTER 1

Overview 1

Introduction 1

Accessing the CLI Using a Router Console 2

Using the Console Interface 5

Initial Bootup Security 5

Enforce Changing Default Password 5

Telnet and HTTP 7

Accessing the CLI from a Remote Console 7

Preparing to Connect to the Router Console 7

Setting Up the IR1800 to Run SSH 8

Using Telnet to Access a Console Interface 9

CLI Session Management 10

Information About CLI Session Management 10

Changing the CLI Session Timeout 11

Locking a CLI Session 11

CHAPTER 2

Using Cisco IOS XE Software 13

Understanding Command Modes 13

Keyboard Shortcuts 15

Using the no and default Forms of Commands 15

Using the History Buffer to Recall Commands 16

Managing Configuration Files 16

Saving Configuration Changes 16

Filtering Output from the show and more Commands 17

Finding Support Information for Platforms and Cisco Software Images 18
Using Cisco Feature Navigator 18
Getting Help 18
Finding Command Options: Example 19
Using Software Advisor 22
Using Software Release Notes 22

CHAPTER 3 **Basic Router CLI Configuration** 23
IR1800 Interface Naming 23
Basic Configuration 24
Configuring Global Parameters 28
Configuring the Gigabit Ethernet Interface 29
Support for sub-interface on GigabitEthernet0/0/0 30
Configuring a Loopback Interface 30
Enabling Cisco Discovery Protocol 31
Configuring Command-Line Access 32
Configuring Static Routes 33
Configuring Dynamic Routes 35
Configuring Routing Information Protocol 35
Configuring Enhanced Interior Gateway Routing Protocol 36
Modular QoS (MQC) 37
Configuring the Serial Interface 37
Specifying an Asynchronous Serial Interface 37
Specifying Asynchronous Serial Encapsulation 38
Configuring the Serial Port 38

CHAPTER 4 **Web User Interface (WebUI)** 41
Day 0 Web User Interface 41
Day 0 Cellular Mode 41
Configuration Notes 42
Configuring Your Computer to Connect to the Router 42
Configuring the WebUI through the Browser 43

CHAPTER 5 **Configuring Secure Shell** 49

Information About Secure Shell 49
Prerequisites for Configuring Secure Shell 49
Restrictions for Configuring Secure Shell 49
SSH And Router Access 50
SSH Servers, Integrated Clients, and Supported Versions 50
SSH Configuration Guidelines 51
How to Configure Secure Shell 51
Setting Up the IR1800 to Run SSH 51
Configuring the SSH Server 52
Monitoring the SSH Configuration and Status 54
Configuring the Router for Local Authentication and Authorization 54
Information about Secure Copy 56
Prerequisites for Secure Copy 56
Restrictions for Configuring Secure Copy 57
Configuring Secure Copy 57
Additional References 58

CHAPTER 6 **NTP Timing Based on GPS Clock** 59
Configuring NTP using GPS Time 59

CHAPTER 7 **New Features in Cisco IOS XE 17.7.1** 61
Support 1G SFPs 61

CHAPTER 8 **New Features for Cisco IOS XE 17.8.1** 63
Cellular Serviceability Enhancements 63
GNMI Broker (GNMIB) Update 64
gRPC Network Operations Interface Update 64
Raw Socket Feature Enhancement 64
SCADA Enhancement for TNB 64

CHAPTER 9 **New Features For Cisco IOX XE 17.9.1** 67
Install Mode Support 67
Cellular Boot Time Improvements 68
IOS XE Downgrade Warning 68

SNMP Polling of Temperature OID 69
GPS Mode Enabled By Default 70
Packet Capture Support for CANBUS 70
GPS and Dead Reckoning Support for the J1939 Connector 71

CHAPTER 10 **New Features for Cisco IOS XE 17.10.1a** 73
Digital Subscriber Line (DSL) SFP Support on the IR1800 73
vManage Support for the WP-WIFI6-x Module 73
Enable Secure Data Wipe Capabilities 73
Rawsocket Keepalive Configuration CLI 74

CHAPTER 11 **Managing Configuration Files** 77
Understanding Configuration Files 77
Finding the Software Version 78
Managing and Configuring a Consolidated Package Using copy and boot Commands 78
Upgrading the Router Image through the WebUI 80

CHAPTER 12 **Cisco IOS XE Installation Methods** 83
Bundle Mode versus Install Mode 83
Installing the Software using install Commands 83
Restrictions for Installing the Software Using install Commands 84
Information About Installing the Software Using install Commands 84
Install Mode Process Flow 85
Booting the Platform in Install Mode 89
One-Step Installation OR Converting from Bundle Mode to Install Mode 90
Three-Step Installation 91
Upgrading in Install Mode 93
Downgrading in Install Mode 93
Terminating a Software Installation 93
Configuration Examples 93
One Step Installation 94
Three Step Installation 95
Showing the Installed Packages 97
Showing Committed and Uncommitted Packages 98

Removing Inactive Packages 99
Troubleshooting Software Installation Using install Commands 100

CHAPTER 13 **Installing the Software** 101
Installing the Software 101
Licensing 101
Cisco Software Licensing 101
Consolidated Packages 102
Network-Essentials 102
Network-Advantage 102
Related Documentation 102
How to Install the Software for Cisco IOS XE 102
Installing the Cisco IOS XE Release 103
ROMMON Images 104
File Systems 104
Option to Enable or Disable USB Access 105
Autogenerated File Directories and Files 106
Flash Storage 107
LED Indicators 107
Related Documentation 108

CHAPTER 14 **Software Maintenance Upgrade (SMU)** 109
Software Maintenance Upgrade (SMU) 109
SMU Work-flow and Basic Requirements 110
SMU Example 110
Installing a Patch Image 110
Uninstalling the Patch Image 113
Uninstalling the Patch Image Using Rollback 113
Uninstalling the Patch Image Using Deactivate, Commit, and Remove 115

CHAPTER 15 **Smart Licensing Using Policy (SLP)** 119
SLP Overview 119
License Enforcement Types 119
SLP Architecture 120

Product Instance 120
Cisco Smart Software Manager (CSSM) 120
Cisco Smart Licensing Utility (CSLU) 121
Customer Topologies 121
License Installation Procedure - Full Offline Access Topology 122
Procedure to Register Product Instance in CSSM 123
Importing the ACK file from CSSM to your Device 125
Removing the Device from CSSM 126
License Installation Procedure - CSLU has No Access to CSSM 127
Procedure when devices are connected to the CSLU 128
Exporting the AuthRequest File to CSSM 131
Uploading the Authorization Request Code file into CSLU 137
License Installation Process in the Router 138

CHAPTER 16 **Configuring Ethernet Switch Ports** 141
Configuring VLANs 141
VLAN Trunking Protocol (VTP) 142
Configuring 802.1x Authentication 142
Configuring Spanning Tree Protocol 143
Configuring MAC Address Table Manipulation 145
Configuring Switch Port Analyzer 146
Configuring IGMP Snooping 147

CHAPTER 17 **Power Over Ethernet (PoE)** 149
Power over Ethernet Overview 149
Device Detection and Power Allocation 149
Command Line Interface 149

CHAPTER 18 **Cellular Pluggable Interface Module Configuration Guide** 153

CHAPTER 19 **Cisco Wi-Fi Interface Module (WIM)** 155
Cisco Wi-Fi Interface Module (WIM) Overview 155
Hardware Overview 156
Software Overview 157

Management Commands 158
WIM Configuration Overview 158
WIM Baseline Configuration 159
Configuration Steps for the WIM 160
Normal Bootup 162
Manual Reload 162
Bootloader Mode 163
Router Commands 163
WIM Power Down 163
Connect to the Access Point through the Router Console 164
Typical Deployment Scenarios 165
Embedded Wireless Controller 165
EWC Baseline Configuration 166
EWC mode – WebUI Management 166
Control And Provisioning of Wireless Access Points (CAPWAP) 167
CAPWAP Baseline Configuration 168
The CAPWAP Joining Process 169
CAPWAP Management from Wireless LAN Controller 170
Workgroup Bridge (WGB) 170
WGB/uWGB Baseline Configuration 171
Wi-Fi Mode Conversion 173
Converting From CAPWAP to EWC Mode 174
Converting From CAPWAP to WGB Mode 175
Converting From EWC to CAPWAP Mode 175
Upgrading the Firmware on the Cisco Wi-Fi Interface Module 175
Connection to the Ethernet Switch 177
Host Router Configuration 177
Access Mode 178
Configuration Example 179
Trunk Mode 179
Configuration Example 180
Wi-Fi Traffic Behind a NAT 180
Configuration Example 181
Factory Reset 182

Factory Reset Example 182
Flexible Antenna Port 183
Switching Bands 183
Verify the Band Mode 184
Configuration 184
Verify the Configuration 185
Related Documentation 185

CHAPTER 20 **Digital I/O, Ignition, and CAN Bus Connectivity** 187
Overview 187
Digital IO 187
Controller Area Network (CAN) Bus 188
IOx CAN Bus Support 189
Important Notes 189
Configuring Digital IO 189
Ignition Power Management Overview 190
Features of Ignition Power Management 190
Ignition Sense Overview 191
IR1835 Ignition Switch 192
IR1800 Ignition and Battery Voltage 194
Command Line Interface (CLI) 194
Default Values 196
Ignition Power Management Yang Model 196
Support SNMP MIB for Ignition Power Management 197

CHAPTER 21 **Configuring GPS** 199
GPS Overview 199
Cellular Modem-Based GPS 201
GPS/Dead Reckoning module (IRM-GNSS-ADR) 201
GPS Dead Reckoning 201
Dead Reckoning Overview 201
Command Line Interface 202
Feature Limitations 206
IR1800 GPS DR Module Calibration 206

Dead Reckoning for GPS NMEA data streaming 209
National Marine Electronics Association (NMEA) IOx Support 210
NMEA UDP Socket Support 210
NMEA UDP Configuration with Yang 214
Yang Data Model Support 216
Example: Connecting to a Server Hosting a GPS Application 217

CHAPTER 22 **Information About SCADA** 219
Supervisory Control And Data Acquisition (SCADA) Overview 219
Role of the IR1800 219
Key Terms 220
Protocol Translation Application 220
Prerequisites 221
Guidelines and Limitations 222
Default Settings 222
Configuring Protocol Translation 222
Enabling the IR1800 Serial Port and SCADA Encapsulation 222
EXAMPLE 223
Configuring T101 and T104 Protocol Stacks 223
Prerequisites 223
Configuring the T101 Protocol Stack 224
EXAMPLE 225
Configuring the T104 Protocol Stack 226
EXAMPLE 228
Configuration Example 229
Configuring the DNP3 Protocol Stacks 231
Configuring DNP3 Serial 231
EXAMPLE 232
Configuring DNP3 IP 232
EXAMPLE 234
Starting and Stopping the Protocol Translation Engine 235
EXAMPLE 235
Verifying Configuration 235
Debug Commands 236

CHAPTER 23 **Raw Socket Transport** 237

Overview 237

Information About Raw Socket Transport 237

TCP Transport 238

UDP Transport 238

Serial Data Processing 239

VRF-Aware Raw Socket 239

Prerequisites 239

Guidelines and Limitations 240

Default Settings 240

Configuring Raw Socket Transport 240

Enabling Raw Socket Transport on the Serial Interface 240

EXAMPLE 241

Configuring Common Raw Socket Line Options 241

EXAMPLE 242

Configuring Raw Socket TCP 242

Configuring the Raw Socket TCP Server 242

EXAMPLE 243

Configuring the Raw Socket TCP Client 243

EXAMPLE 244

Configuring a Raw Socket UDP Peer-to-Peer Connection 245

EXAMPLE 245

Verifying Configuration 246

Configuration Example 246

Raw Socket TCP 246

Raw Socket UDP 247

Raw Socket VRF 248

CHAPTER 24 **System Messages** 251

Information About Process Management 251

How to Find Error Message Details 251

CHAPTER 25 **Environmental Monitoring** 257

Environmental Monitoring and Reporting Functions 257
Environmental Monitoring Functions 257
Environmental Reporting Functions 258
Additional References 265
Technical Assistance 266

CHAPTER 26 **IOx Application Hosting** 267
Application Hosting 267
Information About Application Hosting 267
Need for Application Hosting 267
IOx Overview 268
Cisco Application Hosting Overview 268
IOXMAN 268
Application Hosting on the IR1800 Industrial Integrated Services Router 269
VirtualPortGroup 269
vNIC 270
How to Configure Application Hosting 271
Enabling IOx 271
Configuring a VirtualPortGroup to a Layer 3 Data Port 272
Installing and Uninstalling Apps 274
Overriding the App Resource Configuration 274
Verifying the Application Hosting Configuration 276
Configuration Examples for Application Hosting 277
Example: Enabling IOx 277
Example: Configuring a VirtualPortGroup to a Layer 3 Data Port 277
Example: Installing and Uninstalling Apps 277
Example: Overriding the App Resource Configuration 278
Native docker support 278
Digital IO for IOx container applications 279
Signed Application Support 280

CHAPTER 27 **Serial Relay Service** 281
IOx Serial Relay Service 281
Data Paths 281

Configuration Commands 283

CHAPTER 28 **Cisco SD-WAN Support** 285

Cisco SD-WAN Overview 285

Related Documentation 286

CHAPTER 29 **ROM Monitor Overview** 287

ROM Monitor Overview 287

Access ROM Monitor Mode 288

Checking the Current ROMMON Version 288

Commonly Used ROM Monitor Commands 289

Examples 289

Changing the ROM Monitor Prompt 290

Displaying the Configuration Register Setting 290

Environment Variable Settings 290

Frequently Used Environmental Variables 291

Displaying Environment Variable Settings 291

Entering Environment Variable Settings 291

Saving Environment Variable Settings 291

Exiting ROM Monitor Mode 292

Configuration Example 292

Upgrading the ROMmon for a Router 292

CHAPTER 30 **WAN Monitoring** 295

Information About WANMon 295

Built-in Recovery Actions 295

Prerequisites 296

Guidelines and Limitations 296

Configuring WANMon 297

Verifying WANMon Configuration 298

Configuration Examples 299

WANMon Cellular Interface Configuration Example 299

Multiple WAN Link Monitoring Example 299

CHAPTER 31 **Yang Data Models** 301

Support for YANG Data Models (Call-home) 301

Yang Data Model Support for Raw Socket Transport 301

Yang Data Model Support for Scada 302

Scada Yang Models 302

Cisco-IOS-XE-scada-gw 303

Cisco-IOS-XE-scada-gw-oper 304

CHAPTER 32 **Connected Grid NMS Agent (CGNA) Support** 305

Connected Grid NMS Agent (CGNA) Support 305

Overview 305

WebSocket Support 306

CHAPTER 33 **Process Health Monitoring** 307

Monitoring Control Plane Resources 307

Avoiding Problems Through Regular Monitoring 307

Cisco IOS Process Resources 307

Overall Control Plane Resources 314

Monitoring Hardware Using Alarms 316

Router Design and Monitoring Hardware 316

BootFlash Disk Monitoring 317

Approaches for Monitoring Hardware Alarms 317

Viewing the Console or Syslog for Alarm Messages 317

Enabling the logging alarm Command 317

Network Management System Alerts a Network Administrator when an Alarm is Reported Through SNMP 317

CHAPTER 34 **Troubleshooting** 319

Troubleshooting 319

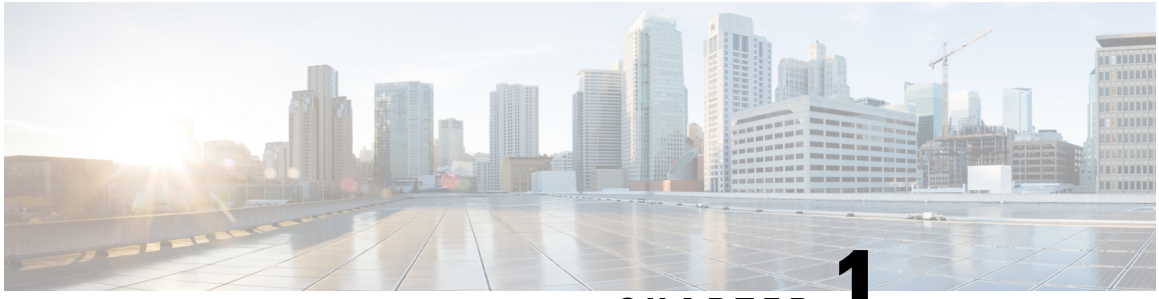
Understanding Diagnostic Mode 319

Before Contacting Cisco or Your Reseller 320

show interfaces Troubleshooting Command 320

Software Upgrade Methods 321

Change the Configuration Register 321
Configuring the Configuration Register for Autoboot 323
Reset the Router 323
Recovering a Lost Password 324
Reset the Password and Save Your Changes 325
Reset the Configuration Register Value 326
Configuring a Console Port Transport Map 326
Viewing Console Port, SSH, and Telnet Handling Configurations 328
Using the factory reset Commands 330



CHAPTER 1

Overview

This section contains the following:

- Introduction, on page 1
- Accessing the CLI Using a Router Console, on page 2
- Initial Bootup Security, on page 5
- Accessing the CLI from a Remote Console , on page 7
- CLI Session Management, on page 10

Introduction

The Cisco Catalyst IR1800 Rugged Series Router is a modular industrial router. The IR1800 series has four Base platforms with additional Pluggable Modules that can be added. The Pluggable Modules provides the flexibility of adding different interfaces to the base platform.

The IR1800 ISR series features a Base Platform with modularity that includes:

- Pluggable Interface Module (PIM)
- mSATA Module (SSDM)
- GPS Module (GNSS)
- Wi-Fi Interface Module (WIM)

The IR1800 series consists of four base platforms. They are:

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

The following table shows details of the differences:

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

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

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.

Accessing the CLI Using a Router Console

Cisco IR1800 routers have console port with only USB support.

The console port is a micro USB Type C 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

http://www.ftdichip.com/Support/Documents/InstallGuides/Mac_OS_X_Installation_Guide.pdf

Note The latest VCP Drivers do not work with MAC OS 10.14.x and beyond. If you require OS X 10.4 support, please install version 3.1 of the VCP driver.

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
GigabitEthernet0/1/0   unassigned      YES unset  down                  down
GigabitEthernet0/1/1   unassigned      YES unset  down                  down
GigabitEthernet0/1/2   unassigned      YES unset  down                  down
GigabitEthernet0/1/3   unassigned      YES unset  up                    up
Async0/2/0             unassigned      YES unset  up                    down
Vlan1                  unassigned      YES unset  up                    up
```

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 GigabitEthernet0/1/0
!
interface GigabitEthernet0/1/1
!
interface GigabitEthernet0/1/2
!
interface GigabitEthernet0/1/3
!
interface Vlan1
no shutdown
ip address 192.168.1.1 255.255.255.0
no mop enabled
ip dhcp pool wDHCPool
network 192.168.1.0 255.255.255.0
default-router 192.168.1.1
!
end


[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
```

Initial Bootup Security

This section contains the following:

Enforce Changing Default Password

When the device is first booted after factory reset or fresh from the factory, the following prompt is received on the console:

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

In previous documentation, Cisco recommended using the **enable secret** command instead of the **enable password** command because this offers an improved encryption algorithm.

The initial dialog forces setting a new enable password, and also using the **enable secret** command instead. The following is an example:

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

  Enter host name [Router]: router-1

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

  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: **********

The virtual terminal password is used to protect
  access to the router over a network interface.
  Enter virtual terminal password: *********
  Configure SNMP Network Management? [yes]: no

Enter interface name used to connect to the
management network from the above interface summary: Ethernet0/0

Configuring interface Ethernet0/0:
  Configure IP on this interface? [yes]: no

The following configuration command script was created:
hostname router-1
enable secret 9 $9$emUzIshVXwlUaE$nTzhgi9STdZKzQc4VJ0kEaCqafjUNdCD7ZUf37SY9qg
enable password password-1
.
.
[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
.
.
router-1>en
Password:
router-1#sh run | sec enable
enable secret 9 $9$emUzIshVXwlUaE$nTzhgi9STdZKzQc4VJ0kEaCqafjUNdCD7ZUf37SY9qg
enable password password-1
```

The following is an example of what happens if you answer **no** to the initial configuration dialog:

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

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

After the enable secret is prompted during the first login, and the admin enters a password, the admin entered password will be always masked. If the admin enters a weak password, they will be prompted again to enter strong password (i.e. the standard mix of upper/lower case characters, special characters, numbers etc.). The prompting will continue until the admin enters a strong password. The admin will be prompted to enter the strong secret password twice for confirming that admin is sure that it is the secret that they want to configure.

Telnet and HTTP

There has been a change in the telnet and http boot configuration as of release 17.3.1. When the device is first booted after factory reset or fresh from the factory, the following takes place:

- Disable telnet
- Disable http server. HTTP client works.
- Enable SSH
- Enable https server

Accessing the CLI from a Remote Console

The remote console of the IR1800 can be accessed through Telnet or SSH. Telnet is disabled by default, and the more secure SSH should be used. For details on SSH access see the SSH chapter.

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

Preparing to Connect to the Router Console

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.

Setting Up the IR1800 to Run SSH

Follow the procedure given below to set up your device to run SSH:

Before you begin

Configure user authentication for local or remote access. This step is required. For more information, see Related Topics below.

Procedure

	Command or Action	Purpose
Step 1	**enable** **Example:** `IR1800> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `IR1800# configure terminal`	Enters global configuration mode.
Step 3	**hostname** *hostname* **Example:** `IR1800(config)# hostname your_hostname`	Configures a hostname and IP domain name for your device. **Note** Follow this procedure only if you are configuring the device as an SSH server.
Step 4	**ip domain-name** *domain_name* **Example:** `IR1800(config)# ip domain-name your_domain_name`	Configures a host domain for your device.

	Command or Action	Purpose
Step 5	**crypto key generate rsa** **Example:** `IR1800(config)# crypto key generate rsa`	Enables the SSH server for local and remote authentication on the device and generates an RSA key pair. Generating an RSA key pair for the device automatically enables SSH. We recommend that a minimum modulus size of 1024 bits. When you generate RSA keys, you are prompted to enter a modulus length. A longer modulus length might be more secure, but it takes longer to generate and to use. **Note** Follow this procedure only if you are configuring the device as an SSH server.
Step 6	**end** **Example:** `IR1800(config)# end`	Returns to privileged EXEC mode.
Step 7	**show running-config** **Example:** `IR1800# show running-config`	Verifies your entries.
Step 8	**copy running-config startup-config** **Example:** `IR1800# copy running-config startup-config`	(Optional) Saves your entries in the configuration file.

Using Telnet to Access a Console Interface

Before you begin

Telnet is considered a security risk, and is disabled by defualt. If you need to enable it, see Configuring Telnet

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.

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`

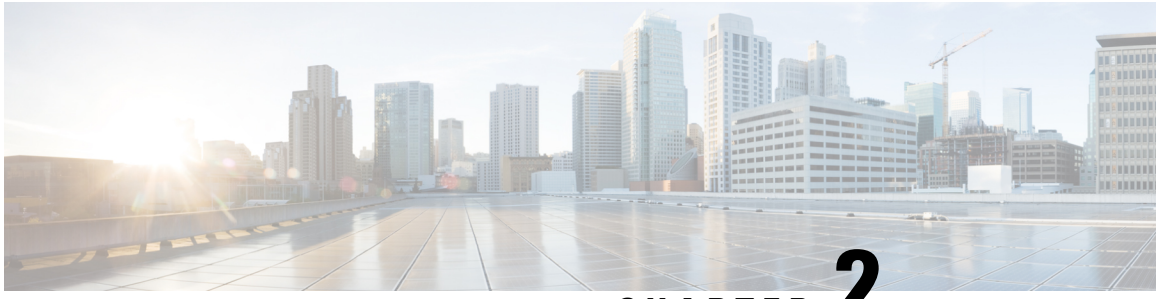
Enables the line to be locked.

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

Using Cisco IOS XE Software

This chapter contains the following sections:

- Understanding Command Modes, on page 13
- Keyboard Shortcuts, on page 15
- Using the no and default Forms of Commands, on page 15
- Using the History Buffer to Recall Commands, on page 16
- Managing Configuration Files, on page 16
- Saving Configuration Changes, on page 16
- Filtering Output from the show and more Commands, on page 17
- Finding Support Information for Platforms and Cisco Software Images, on page 18

Understanding 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.

The following table describes how to access and exit various common command modes of the Cisco IOS XE software. It also shows examples of the prompts displayed for each mode.

Table 1: 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 the following 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.

Command Mode	Access Method	Prompt	Exit Method
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.

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.

The following table lists the keyboard shortcuts for entering and editing commands.

Table 2: 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.

Using the no and default Forms of Commands

Almost every configuration command has a **no** form. In general, use the **no** form 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.

Many CLI commands also have a **default** form. 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.

Using 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.

The following table lists the history substitution commands.

Table 3: 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.

Managing Configuration Files

The startup configuration file is stored in the nvram: file system and the running configuration files are stored in the system: file system. 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.

Saving Configuration Changes

Use the **copy running-config startup-config** command to save your configuration changes to the startup configuration so that the changes will not be lost if the software reloads or a power outage occurs. For example:

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

```
IR1101#
*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.

This task saves the configuration to the NVRAM.

Filtering Output from the show and more Commands

You can search and filter the output of **show** and **more** commands. This functionality is useful if you need to sort through large amounts of output or if you want to exclude output that you need not see.

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.

Example

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#
```

Finding 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.

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

The group of feature sets that are available for a specific platform depends on which Cisco software images are included in a release. 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.

Using Cisco Feature Navigator

Use Cisco Feature Navigator to find information about platform support and software image support. 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. To use the navigator tool, an account on Cisco.com is not required.

Getting 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 the following 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.

Finding Command Options: Example

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.

The following table shows examples of using the question mark (**?**) to assist you in entering commands.

Table 4: 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/?`
`<0-6> Port Adapter number`
 	
`Router (config)# interface GigabitEthernet 0/0/?`
`<0-143> GigabitEthernet interface number`
 	
`Router (config)# interface GigabitEthernet 0/0/0?`
`. <0-147>`
 	
`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.

Using Software Advisor

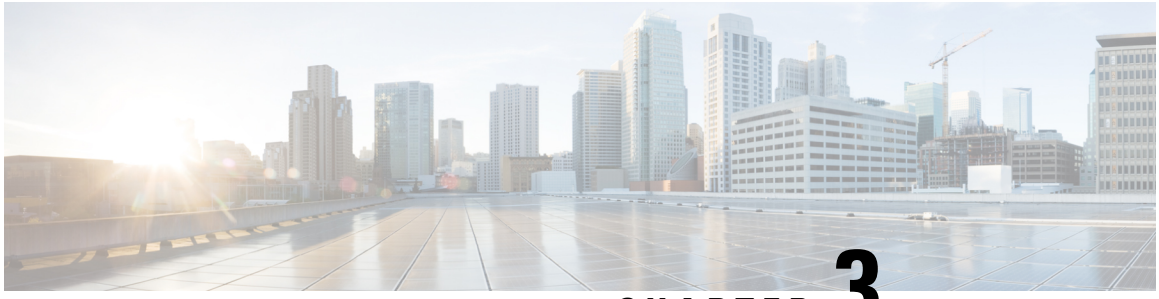
Cisco maintains the Software Advisor tool. See Tools and Resources. Use the Software Advisor tool to see if a feature is supported in a Cisco IOS XE release, to locate the software document for that feature, or to check the minimum software requirements of Cisco IOS XE software with the hardware installed on your router. You must be a registered user on Cisco.com to access this tool.

Using Software Release Notes

See the release notes for information about the following:

- Memory recommendations
- Open and resolved severity 1 and 2 caveats

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 3

Basic Router CLI Configuration

This chapter contains the following sections:

- IR1800 Interface Naming, on page 23
- Basic Configuration, on page 24
- Configuring Global Parameters, on page 28
- Configuring the Gigabit Ethernet Interface, on page 29
- Support for sub-interface on GigabitEthernet0/0/0, on page 30
- Configuring a Loopback Interface, on page 30
- Enabling Cisco Discovery Protocol, on page 31
- Configuring Command-Line Access, on page 32
- Configuring Static Routes, on page 33
- Configuring Dynamic Routes, on page 35
- Modular QoS (MQC), on page 37
- Configuring the Serial Interface, on page 37

IR1800 Interface Naming

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

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

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

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

Basic Configuration

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

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

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

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

```
4AA4E80D DB6FD1C9 60B1FD18 FFC69C96 6FA68957 A2617DE7 104FDC5F EA2956AC
7390A3EB 2B5436AD C847A2C5 DAB553EB 69A9A535 58E9F3E3 C0BD23CF 58BD7188
68E69491 20F320E7 948E71D7 AE3BCC84 F10684C7 4BC8E00F 539BA42B 42C68BB7
C7479096 B4CB2D62 EA2F505D C7B062A4 6811D95B E8250FC4 5D5D5FB8 8F27D191
C55F0D76 61F9A4CD 3D992327 A8BB03BD 4E6D7069 7CBADF8B DF5F4368 95135E44
DFC7C6CF 04DD7FD1 02030100 01A34230 40300E06 03551D0F 0101FF04 04030201
06300F06 03551D13 0101FF04 05300301 01FF301D 0603551D 0E041604 1449DC85
4B3D31E5 1B3E6A17 606AF333 3D3B4C73 E8300D06 092A8648 86F70D01 010B0500
03820101 00507F24 D3932A66 86025D9F E838AE5C 6D4DF6B0 49631C78 240DA905
604EDCDE FF4FED2B 77FC460E CD636FDB DD44681E 3A5673AB 9093D3B1 6C9E3D8B
D98987BF E40CBD9E 1AECA0C2 2189BB5C 8FA85686 CD98B646 5575B146 8DFC66A8
467A3DF4 4D565700 6ADF0F0D CF835015 3C04FF7C 21E878AC 11BA9CD2 55A9232C
7CA7B7E6 C1AF74F6 152E99B7 B1FCF9BB E973DE7F 5BDDEB86 C71E3B49 1765308B
5FB0DA06 B92AFE7F 494E8A9E 07B85737 F3A58BE1 1A48A229 C37C1E69 39F08678
80DDCD16 D6BACECA EEBC7CF9 8428787B 35202CDC 60E4616A B623CDBD 230E3AFB
418616A9 4093E049 4D10AB75 27E86F73 932E35B5 8862FDAE 0275156F 719BB2F0
D697DF7F 28
quit
crypto pki certificate chain TP-self-signed-2276770909
certificate self-signed 01
30820330 30820218 A0030201 02020101 300D0609 2A864886 F70D0101 05050030
31312F30 2D060355 04031326 494F532D 53656C66 2D536967 6E65642D 43657274
69666963 6174652D 32323736 37373039 3039301E 170D3230 30393230 31343036
30365A17 0D333030 39323031 34303630 365A3031 312F302D 06035504 03132649
4F532D53 656C662D 5369676E 65642D43 65727469 66696361 74652D32 32373637
37303930 39308201 22300D06 092A8648 86F70D01 01010500 0382010F 00308201
0A028201 0100BAC3 88D3A9B7 259E58A4 0FCF6DB2 2794CC97 CF8DC253 D1CFB83B
ACFA305A 28BA6174 2452EE0B C45E92EA BBA30235 C142D2D3 DA04C6FD A916507C
BAFE6806 BBAB6B02 86B5AC61 05FB5A67 C5449A92 EFAA9519 9A2A084E 94A29BF5
B78604F2 76927505 371AD917 67D8EACF CEBBA6A1 278F5647 DDDBE8AF 8E451772
4709D928 04039C51 C2FA72E2 0C03C426 BB844F76 0BE65C37 60DFDA8E 38EBAFD8
9B3908BF 9B5A50B2 37539BF4 9D3256D9 B118DDF4 BC912AA1 B1E9DFF0 34729AE9
4B594142 B46D7C93 13FF997B 2FECC956 2362A8CC A0CD51EF 5691A2C3 9EB200FE
F4D341AE F35D3C06 8BCC1ACF 42E983FF F8C0B5A5 70906FCD 07F854D3 41CE9402
0572AE66 EF050203 010001A3 53305130 0F060355 1D130101 FF040530 030101FF
301F0603 551D2304 18301680 145C7DAD E37AB191 53C24775 8FC918B5 8059336C
12301D06 03551D0E 04160414 5C7DADE3 7AB19153 C247758F C918B580 59336C12
300D0609 2A864886 F70D0101 05050003 82010100 673243D9 3BBC0321 1FAC5459
926E99BF 60E55344 123B8A22 359B5DA8 E98E0A4F 5FDD49FC 5AF99F8B 87F30704
E74BEC68 DF4D2116 9DBD58D0 F4ABEE17 D9155CAE DBB7E94E 7A058507 CFA8DFB2
90E44C50 F95AD87F 934F904D 8C07CE47 5AEBBB7A EBA3E0C9 6CBA7B34 CC4642B6
DE641222 E045CEF4 27625FD2 FE51853C 574CCCA8 F036874B 93C97278 3D3776F1
E6419A07 46065203 FB81BFFD 1B2D5270 84FA9BAE CC06EE2A DF667257 DA97D96D
3E226378 28CE8460 2570D7D3 4D78C9E2 66FBA5B1 9A6E46AD E466D67F 425BFC40
FA717361 CBAA9AA0 7DB343F9 563B675B F1B6D193 12162EAA 6389A57C CF65AA08
53B07581 87A0C15A D5B6900B E3F98713 F3918F89
quit
!
!
no license feature hseck9
license udi pid IR1835-K9 sn FHH2416P00V
memory free low-watermark processor 47775
!
diagnostic bootup level minimal
!
spanning-tree extend system-id
!
!
redundancy
mode none
!
!
controller Gps-Dr
!
```

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

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

end
```

Configuring Global Parameters

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

Procedure

	Command or Action	Purpose

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

Step 2	**hostname** *name* **Example:** `Router(config)# hostname Router`	Specifies the name for the router.
Step 3	**enable password** *password* or **enable secret password** *password* **Example:**	Specifies a password to prevent unauthorized access to the router.

	Command or Action	Purpose
	`Router(config)# enable password cr1ny5ho`	**Note** In this form of the command, password is not encrypted. To encrypt the password use enable secret password as noted in the previously mentioned Device Hardening Guide.

Configuring the Gigabit Ethernet Interface

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

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

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

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

Procedure

	Command or Action	Purpose
Step 1	**configure terminal**	
Step 2	**ipv6 unicast-routing** **Example:** `Router# ipv6 unicast-routing`	Enables forwarding of IPv6 unicast data packets.
Step 3	**interface GigabitEthernet** *slot/bay/port* **Example:** `Router(config)# interface GigabitEthernet 0/0/0`	Enters the configuration mode for an interface on the router.
Step 4	**ip address** *ip-address mask* **Example:** `Router(config-if)# ip address 192.168.12.2 255.255.255.0`	Sets the IP address and subnet mask for the specified interface. Use this Step if you are configuring an IPv4 address.
Step 5	**ipv6 address** *ipv6-address/prefix* **Example:** `Router(config-if)# ipv6 address 2001.db8::ffff:1/128`	Sets the IPv6 address and prefix for the specified interface. Use this step instead of Step 2, if you are configuring an IPv6 address. IPv6 unicast-routing needs to be set-up as well, see further information in the **IPv6 Addressing and Basic Connectivity Configuration Guide** located here: https://www.cisco.com/c/en/us/td/

	Command or Action	Purpose
		docs/ios-xml/ios/ipv6_basic/configuration/xe-16-10/ip6b-xe-16-10-book/read-me-first.html
Step 6	**no shutdown** **Example:** `Router(config-if)# no shutdown`	Enables the interface and changes its state from administratively down to administratively up.
Step 7	**exit** **Example:** `Router(config-if)# exit`	Exits the configuration mode of interface and returns to the global configuration mode.

Support for sub-interface on GigabitEthernet0/0/0

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

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

Configuring a Loopback Interface

Before you begin

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

To configure a loopback interface, follow these steps.

Procedure

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

	Command or Action	Purpose
Step 4	(Option 2) **ipv6 address** *ipv6-address/prefix* **Example:** `Router(config-if)# ipv6 address 2001:db8::ffff:1/128`	Sets the IPv6 address and prefix on the loopback interface.
Step 5	**exit** **Example:** `Router(config-if)# exit`	Exits configuration mode for the loopback interface and returns to global configuration mode.

Example

Verifying Loopback Interface Configuration

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

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

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

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

Enabling Cisco Discovery Protocol

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

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

Configuring Command-Line Access

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

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

Procedure

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

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

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

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

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

Example

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

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

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

Configuring Static Routes

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

To configure static routes, follow these steps.

Procedure

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

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

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

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

Verifying Configuration

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

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

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

Gateway of last resort is not set

     10.0.0.0/24 is subnetted, 1 subnets
```

```
C       10.108.1.0 is directly connected, Loopback0
S*   0.0.0.0/0 is directly connected, GigabitEthernet0
```

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

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

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

Configuring Dynamic Routes

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

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

Configuring Routing Information Protocol

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

Procedure

	Command or Action	Purpose
Step 1	**router rip** **Example:** `Router(config)# router rip`	Enters router configuration mode, and enables RIP on the router.
Step 2	**version** {**1** | **2**} **Example:** `Router(config-router)# version 2`	Specifies use of RIP version 1 or 2.
Step 3	**network** *ip-address* **Example:** `Router(config-router)# network`	Specifies a list of networks on which RIP is to be applied, using the address of the network of each directly connected network.

	Command or Action	Purpose
	192.168.1.1 Router(config-router)# **network 10.10.7.1**	
Step 4	**no auto-summary** **Example:** Router(config-router)# **no auto-summary**	Disables automatic summarization of subnet routes into network-level routes. This allows subprefix routing information to pass across classful network boundaries.
Step 5	**end** **Example:** Router(config-router)# **end**	Exits router configuration mode, and enters privileged EXEC mode.

Example

Verifying Configuration

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

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

Gateway of last resort is not set

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

Configuring Enhanced Interior Gateway Routing Protocol

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

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

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

Modular QoS (MQC)

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

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

Configuring the Serial Interface

This section describes configuring serial interface management.

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

Table 5: Naming Conventions

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

IR1821

The IR1821 has only a single Async serial port.

IR1831

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

IR1833

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

IR1835

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

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

Specifying an Asynchronous Serial Interface

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

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

Specifying Asynchronous Serial Encapsulation

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

The asynchronous serial interfaces support the following serial encapsulation methods:

- Raw Socket
- Line Relay
- SCADA protocol translation

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

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

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

Configuring the Serial Port

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

Table 6: Configuration Examples

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

`#sh run int Async 0/2/1`
`Building configuration...`
`Current configuration : 64 bytes`
`!`
`interface Async0/2/1`
` no ip address`
` encapsulation raw-tcp`
`end`
	DCE Port with media-type RS232 [Default Configuration]

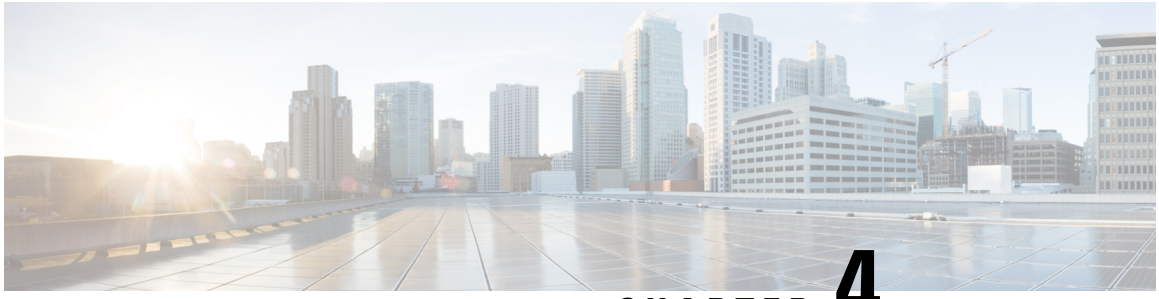
sh run int Async 0/2/0 Building configuration... Current configuration : 64 bytes ! interface Async0/2/0 no ip address encapsulation relay-line end	DTE Port with default media-type RS232. (RS485 not supported).

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

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

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

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

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



CHAPTER 4

Web User Interface (WebUI)

This section contains the following topics:

- Day 0 Web User Interface, on page 41
- Day 0 Cellular Mode, on page 41
- Configuration Notes, on page 42
- Configuring Your Computer to Connect to the Router, on page 42
- Configuring the WebUI through the Browser, on page 43

Day 0 Web User Interface

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

The Day 0 Web User Interface (WebUI) will be 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 either a Windows, Linux or Mac PC/Laptop to one of the LAN ports of the IR1800 and boot the router on Day 0. The PC/Laptop should be configured to obtain an IP address through DHCP.

Once the router boots up in Day 0, the PC/Laptop will acquire IP address in 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".

Note Issue a **write memory** command once the configuration is applied through the WebUI.

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.

Configuration Notes

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 IR1800.

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.

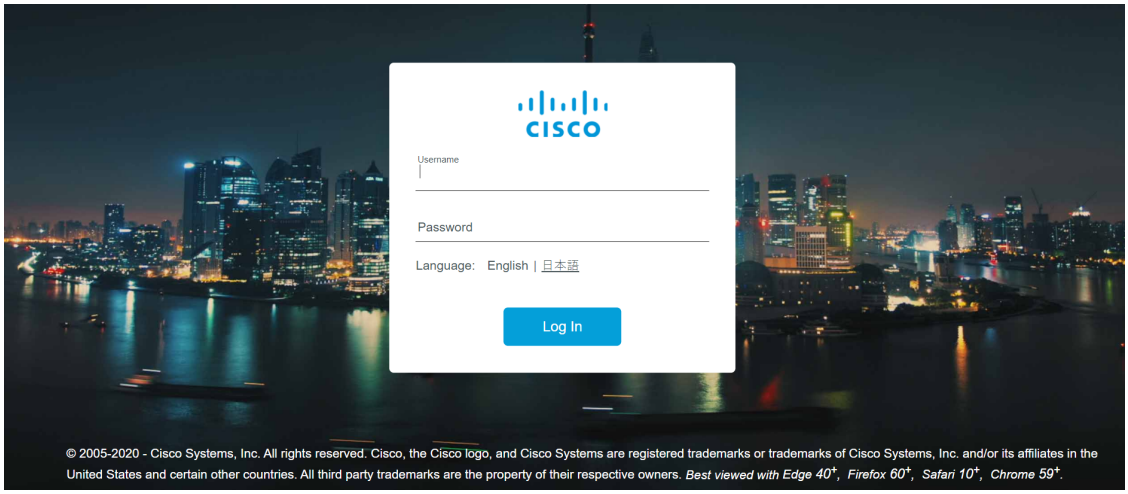
Configuring the WebUI through the Browser

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

Procedure

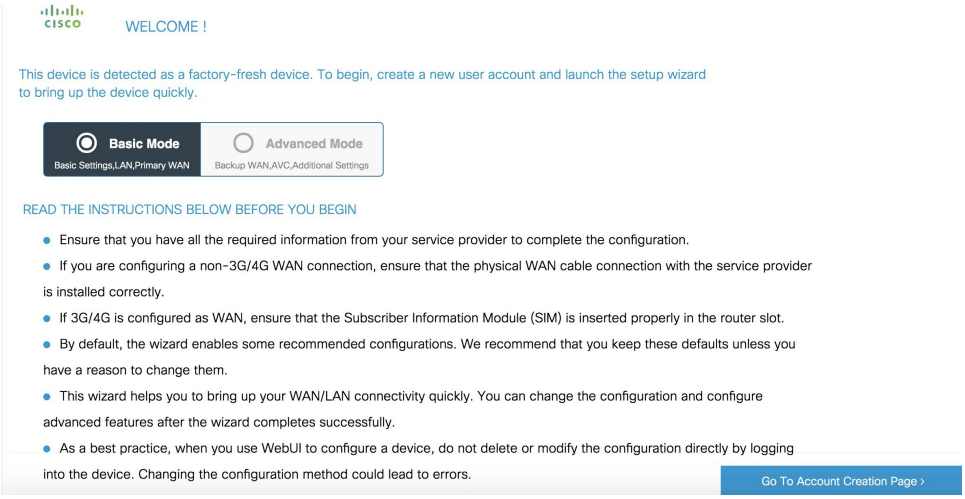
Step 1 Open your browser and enter 192.168.1.1in the address bar. The Login Screen appears. Enter the Username **webui** and the Password **cisco**. Then click **Log In**.

Figure 1: 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 guide, Basic Mode is used. Select **Basic Mode**, then click **Go To Account Creation Page**.

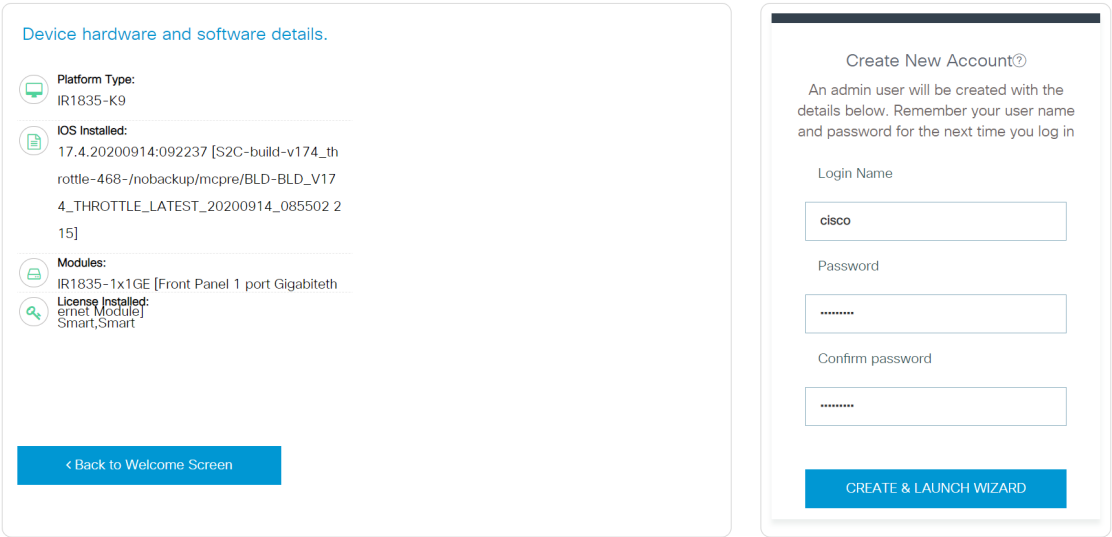
Figure 2: Welcome Screen



Example:

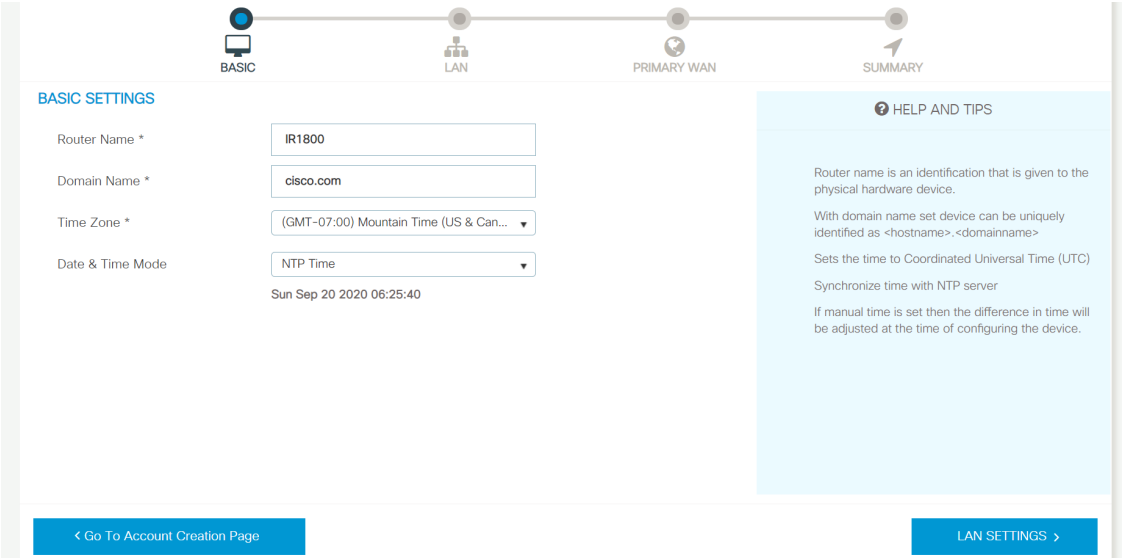
Step 3 The Create New Account Screen appears. Create a new Login Name and Password to access the WebUI. Click **Create and Launch Wizard**.

Figure 3: Create New Account Screen



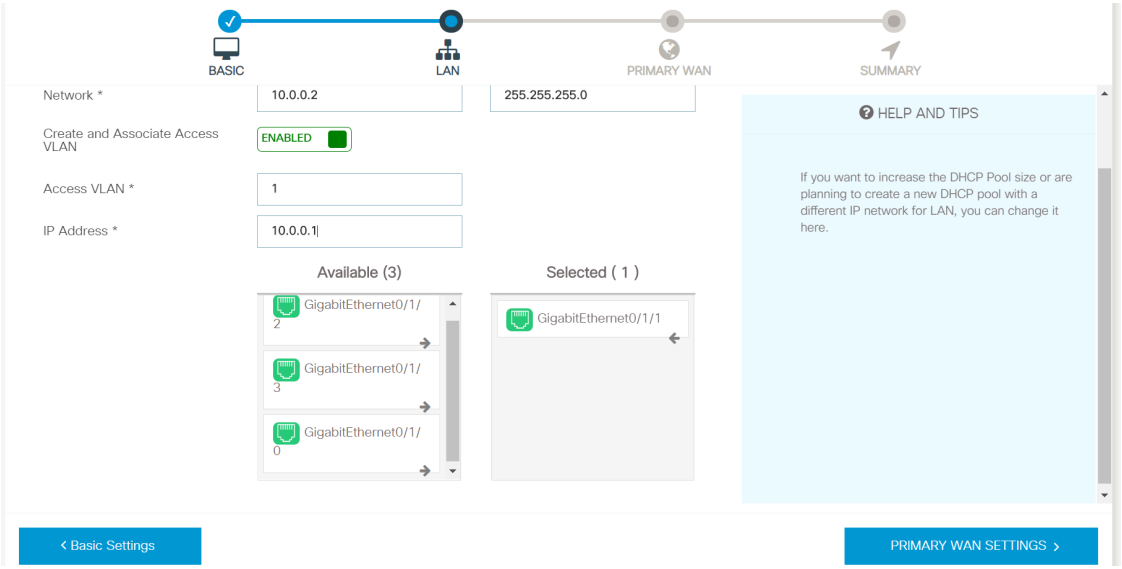
Step 4 The Basic Settings Screen appears. Provide a Router Name (hostname), Domain Name, Time Zone and Date & Time Mode. Click **LAN SETTINGS**.

Figure 4: Basic Settings Screen



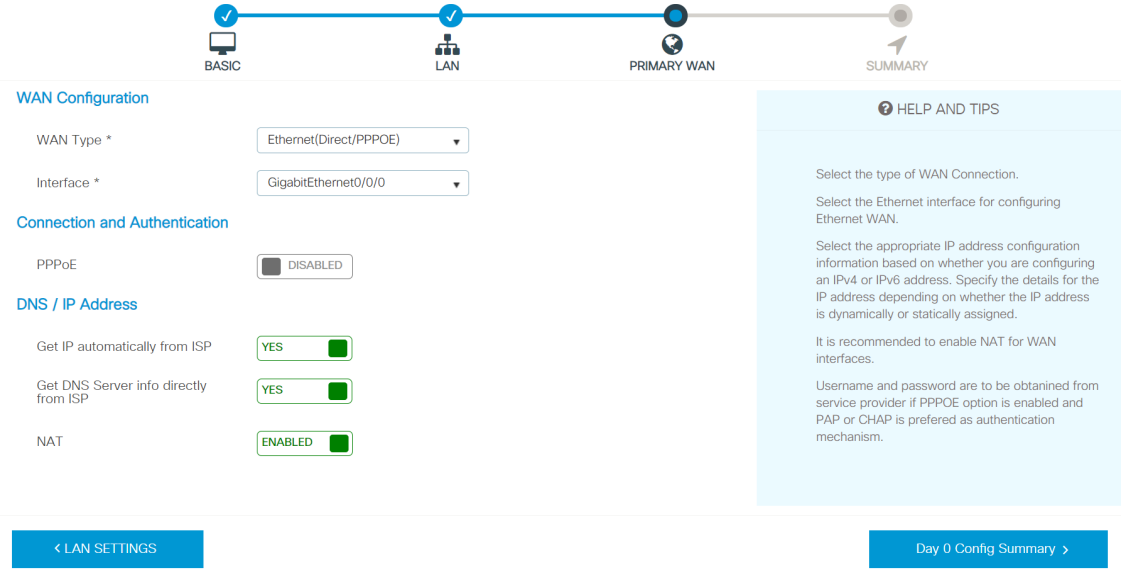
Step 5 The LAN Configuration Screen appears. Enter the webui_dhcp Pool Name, VLAN interface IP address, and select the interface that is connected to your laptop from the list of available interfaces. Click **PRIMARY WAN SETTINGS**.

Figure 5: LAN Configuration Screen



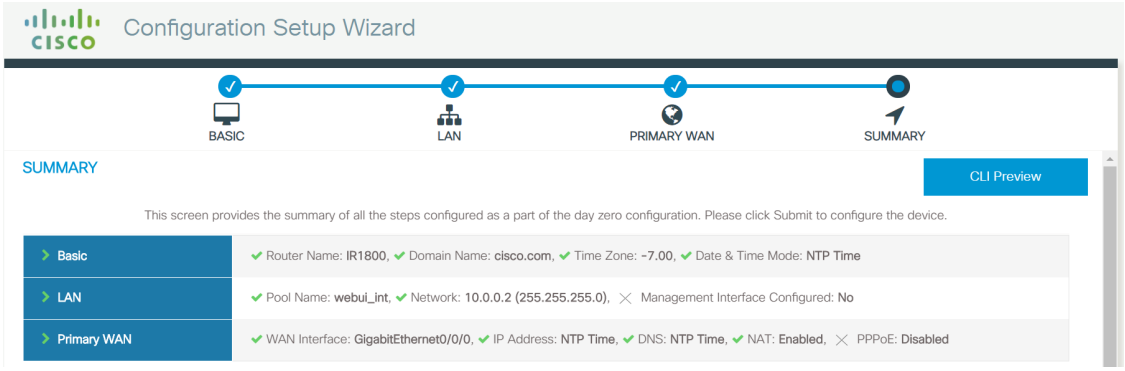
Step 6 The PRIMARY WAN SETTINGS Screen appears. Configure the WAN interface by selecting the WAN Type and Interface from the available options. Next enter your DNS IP address information and select Enable/Disable NAT. Click **Day 0 Config Summary**.

Figure 6: Primary WAN Interface Screen



Step 7 The Review Summary Screen appears. Verify your entries before applying the configuration.

Figure 7: Summary Screen

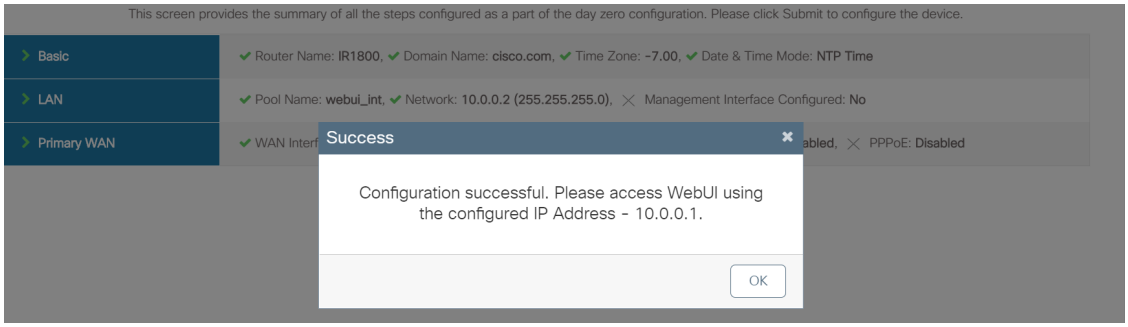


Step 8 (Optional) You can click on **CLI Preview** to see the Configuration that is being applied to the router. Close the CLI Preview and if you are ready, Click **Submit**.

Figure 8: CLI Preview Screen

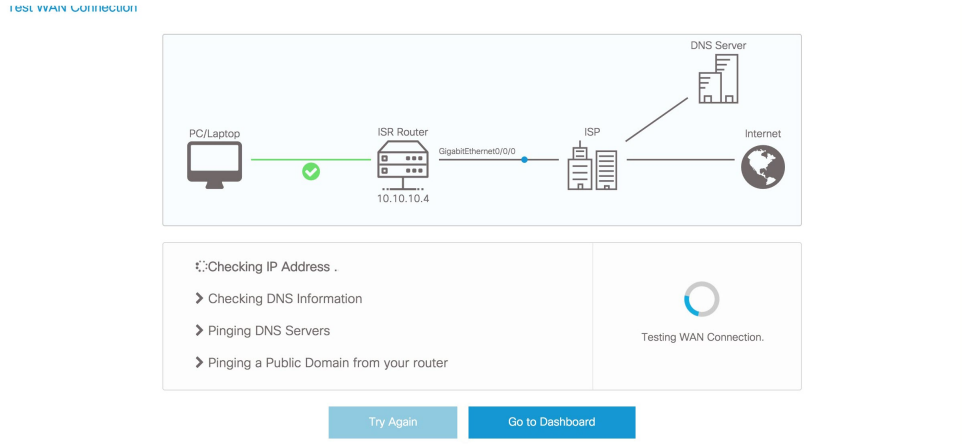
Step 9 After clicking on Submit, a dialog box will appear which informs you that the configuration has been applied successfully. The new WebUI ip address is also presented.

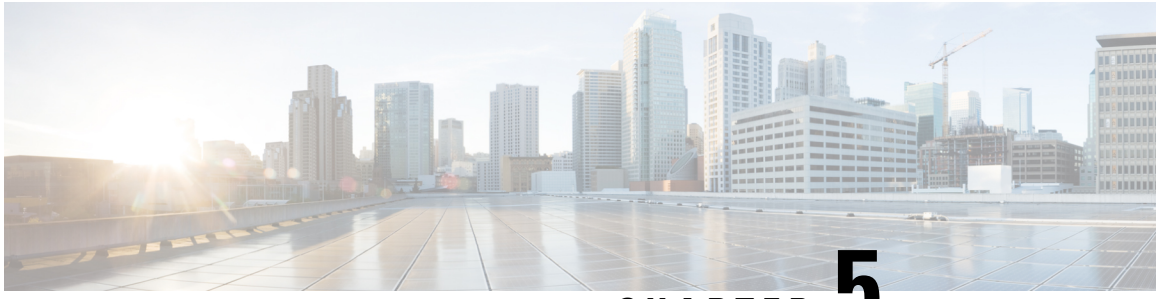
Figure 9: Submit Dialog Box



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 10: Test VLAN Connection Screen





CHAPTER 5

Configuring Secure Shell

This section contains the following topics:

- Information About Secure Shell , on page 49
- How to Configure Secure Shell, on page 51
- Information about Secure Copy, on page 56
- Additional References, on page 58

Information About Secure Shell

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

Prerequisites for Configuring 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

The following are restrictions for configuring the IR1800 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 IR1800 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. SSH provides more security for remote connections than Telnet does by providing strong encryption when a device is authenticated. This software release supports SSH Version 1 (SSHv1) and SSH Version 2 (SSHv2). 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. The SSH client enables a Cisco device to make a secure, encrypted connection to another Cisco device or to any other device running the SSH server. This connection provides functionality similar to that of an outbound Telnet connection except that the connection is encrypted. With authentication and encryption, the SSH client allows for secure communication over an unsecured network.

The SSH server and SSH integrated client are applications that run on the switch. The SSH server works with the SSH client supported in this release and with non-Cisco SSH clients. The SSH client works with publicly and commercially available SSH servers. The 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 the following user authentication methods:

- TACACS+
- RADIUS
- Local authentication and authorization

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

Setting Up the IR1800 to Run SSH

Follow the procedure given below to set up your device to run SSH:

Before you begin

Configure user authentication for local or remote access. This step is required. For more information, see Related Topics below.

Procedure

	Command or Action	Purpose
Step 1	**enable** **Example:** `IR1800> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `IR1800# configure terminal`	Enters global configuration mode.
Step 3	**hostname** *hostname* **Example:**	Configures a hostname and IP domain name for your device.

	Command or Action	Purpose
	`IR1800(config)# hostname your_hostname`	**Note** Follow this procedure only if you are configuring the device as an SSH server.
Step 4	**ip domain-name** *domain_name* **Example:** `IR1800(config)# ip domain-name your_domain_name`	Configures a host domain for your device.
Step 5	**crypto key generate rsa** **Example:** `IR1800(config)# crypto key generate rsa`	Enables the SSH server for local and remote authentication on the device and generates an RSA key pair. Generating an RSA key pair for the device automatically enables SSH. We recommend that a minimum modulus size of 1024 bits. When you generate RSA keys, you are prompted to enter a modulus length. A longer modulus length might be more secure, but it takes longer to generate and to use. **Note** Follow this procedure only if you are configuring the device as an SSH server.
Step 6	**end** **Example:** `IR1800(config)# end`	Returns to privileged EXEC mode.
Step 7	**show running-config** **Example:** `IR1800# show running-config`	Verifies your entries.
Step 8	**copy running-config startup-config** **Example:** `IR1800# copy running-config startup-config`	(Optional) Saves your entries in the configuration file.

Configuring the SSH Server

Follow these steps to configure the SSH server:

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

Procedure

	Command or Action	Purpose
Step 1	**configure terminal** **Example:** `IR1800# configure terminal`	Enters global configuration mode.
Step 2	**ip ssh version** [2] **Example:** `IR1800(config)# ip ssh version 2`	(Optional) Configures the device to run SSH Version 2. 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 3	**ip ssh** {**timeout** *seconds* | **authentication-retries** *number*}
	Example:
	`IR1800(config)# ip ssh timeout 90`
	`ip ssh authentication-retries 2`
		Configures the SSH control parameters:
		• Specify the time-out value in seconds; the default is 120 seconds. 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 time-out 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 time-out 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; the range is 0 to 5.
		Repeat this step when configuring both parameters.

Step 4	Use one or both of the following: • **line vty** *line_number* [*ending line number*] • **transport input ssh** **Example:**	(Optional) Configures the virtual terminal line settings. • Enters line configuration mode to configure the virtual terminal line settings. For the *line_number* and

	Command or Action	Purpose

	`IR1800(config)# line vty 1 10`
	or
	`IR1800(config-line)# transport input ssh`
		ending_line_number arguments, the range is from 0 to 15.
		• Specifies that the device prevents non-SSH Telnet connections, limiting the device to only SSH connections.

Step 5	**end** **Example:** `IR1800(config-line)# end`	Exits line configuration mode and returns to privileged EXEC mode.

Monitoring the SSH Configuration and Status

Table 7: 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.

Configuring the Router for Local Authentication and Authorization

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

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

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 does not secure the router for HTTP access by using AAA methods.

Procedure

	Command or Action	Purpose
Step 1	**enable** **Example:** `IR1800> enable`	Enables privileged EXEC mode. • Enter your password if prompted.

	Command or Action	Purpose
Step 2	**configure terminal** **Example:** `IR1800# configure terminal`	Enters global configuration mode.
Step 3	**aaa new-model** **Example:** `IR1800(config)# aaa new-model`	Enables AAA
Step 4	**aaa authentication login default local** **Example:** `IR1800(config)# aaa authentication login default local`	Sets the login authentication to use the local username database. The default keyword applies the local user database authentication to all ports.
Step 5	**aaa authorization exec local** **Example:** `IR1800(config-line)# aaa authorization exec local`	Configures user AAA authorization, check the local database, and allow the user to run an EXEC shell.
Step 6	**aaa authorization network local** **Example:** `IR1800(config-line)# aaa authorization network local`	Configures user AAA authorization for all network-related service requests.
Step 7	**username** *name* **privilege** *level* **password** *encryption-type password* **Example:** `IR1800(config-line)# username your_user_name privilege 1 password 7 secret567`	Enters the local database, and establishes a username-based authentication system. Repeat this command for each user. **a.** For *name*, specify the user ID as one word. Spaces and quotation marks are not allowed. **b.** (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. **c.** For encryption-type, enter 0 to specify that an unencrypted password follows. Enter 7 to specify that a hidden password follows.

	Command or Action	Purpose
		d. For password, specify the password the user must enter to gain access to the switch. 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 8	**end** **Example:** `IR1800(config-line)# end`	Exits line configuration mode and returns to privileged EXEC mode.
Step 9	**show running-config** **Example:** `IR1800# show running-config`	Verifies your entries.
Step 10	**copy running-config startup-config** **Example:** `IR1800# copy running-config startup-config`	(Optional) Saves your entries in the configuration file.

Information about Secure Copy

The Secure Copy Protocol (SCP) feature provides a secure and authenticated method for copying router configuration or router image files. SCP relies on Secure Shell (SSH), an application and a protocol that provide a secure replacement for the Berkeley r-tools.

Prerequisites for Secure Copy

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

- 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.

Configuring Secure Copy

To configure the Cisco IR1800 for Secure Copy (SCP) server-side functionality, 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	**aaa new-model** **Example:** `Device(config)# aaa new-model`	Sets AAA authentication at login.
Step 4	**aaa authentication login** {**default** | *list-name}* *method1 [method2...*] **Example:** `Device(config)# aaa authentication login default group tacacs+`	Enables the AAA access control system.
Step 5	**username** *name* [**privilege** *level*] **password** *encryption-type encrypted-password* **Example:** `Device(config)# username superuser privilege 2 password 0 superpassword`	Establishes a username-based authentication system. **Note** You may omit this step if a network-based authentication mechanism, such as TACACS+ or RADIUS, has been configured.

	Command or Action	Purpose
Step 6	**ip scp server enable** **Example:** `Device(config)# ip scp server enable`	Enables SCP server-side functionality.
Step 7	**exit** **Example:** `Device(config)# exit`	Exits global configuration mode and returns to privileged EXEC mode.
Step 8	**show running-config** **Example:** `Device# show running-config`	(Optional) Displays the SCP server-side functionality.
Step 9	**debug ip scp** **Example:** `Device# debug ip scp`	(Optional) Troubleshoots SCP authentication problems.

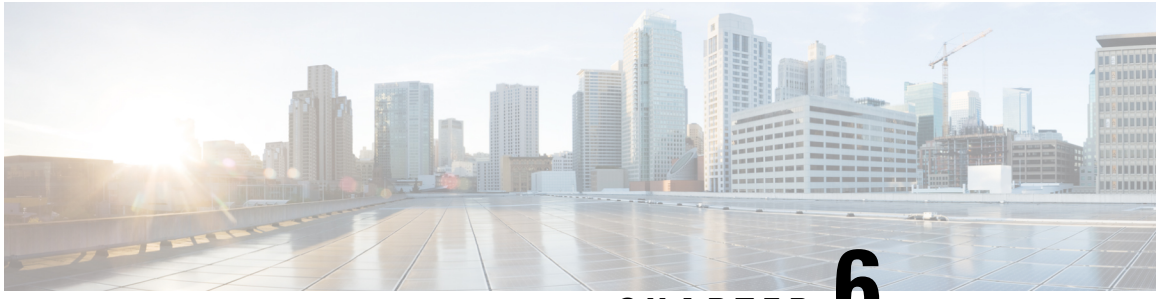
Example

IR1800# **copy scp** <***somefile***> **your_username@remotehost:/**<***some/remote/directory***>

Additional References

The following 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

NTP Timing Based on GPS Clock

This chapter contains the following sections:

- Configuring NTP using GPS Time, on page 59

Configuring NTP using GPS Time

You can configure the GPS time as the reference clock for NTP using the command **ntp refclock gps**.

Note This feature is available with IOS XE release 17.6.1. Further information can be found in NTP Clock Sync with GPS in the Cellular Pluggable Interface Module Configuration Guide.

The GPS time acts as a stratum 0 source, and the Cisco IOS NTP server acts as a stratum 1 device, which in turn provides clock information to its NTP clients (stratum 2 and 3).

Procedure

Step 1 Enter global configuration mode:

Example:

```
Router# configure terminal
```

Step 2 Configure the NTP reference clock as GPS:

Example:

```
Router(config)#ntp refclock gps
```

Step 3 To verify the configuration, use the **show** commands in the following example:

Example:

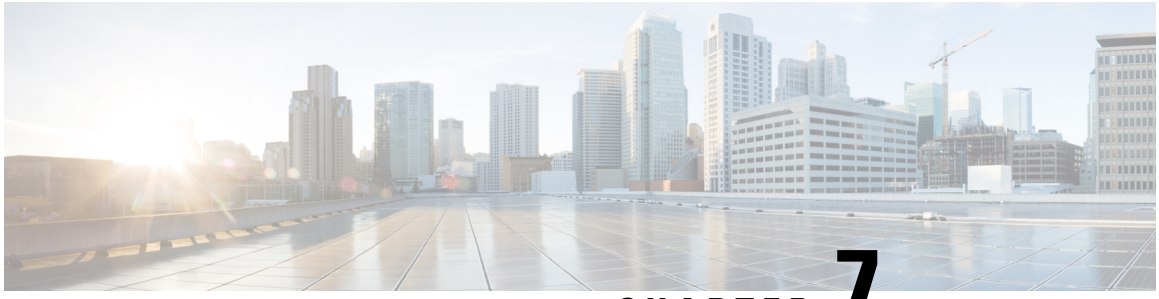
```
Router#
Sep 24 19:58:43.046 GMT: %PKI-6-AUTHORITATIVE_CLOCK: The system clock has been set.
Router#show ntp status
Clock is synchronized, stratum 1, reference is .GPS.
nominal freq is 250.0000 Hz, actual freq is 249.9970 Hz, precision is 2**10
ntp uptime is 94000 (1/100 of seconds), resolution is 4016
```

```
reference time is E31778F3.0B851ED8 (19:58:43.045 GMT Thu Sep 24 2020)
clock offset is 11.0000 msec, root delay is 0.00 msec
root dispersion is 3950.55 msec, peer dispersion is 3938.47 msec
loopfilter state is 'CTRL' (Normal Controlled Loop), drift is 0.000011995 s/s
system poll interval is 64, last update was 7 sec ago.
Router#
Router#
Router#show ntp associations

address ref clock st when poll reach delay offset disp
*~127.127.5.1 .GPS. 0 38 64 7 0.000 11.000 1938.8
* sys.peer, # selected, + candidate, - outlyer, x falseticker, ~ configured
Router#
Router#show clock
20:00:43.660 GMT Thu Sep 24 2020
Router#
```

Step 4 Use the **debug ntp refclock** command to troubleshoot the configuration:

Example:

```
Router#debug ntp ?
adjust NTP clock adjustments
all NTP all debugging on
core NTP core messages
events NTP events
packet NTP packet debugging
refclock NTP refclock messages

Router#debug ntp re
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 7

New Features in Cisco IOS XE 17.7.1

This chapter contains the following sections:

- Support 1G SFPs, on page 61

Support 1G SFPs

Release 17.7.1 will add support for the following SFPs:

GLC-T-RGD

CWDM-SFP-1470=

CWDM-SFP-1610=

CWDM-SFP-1530=

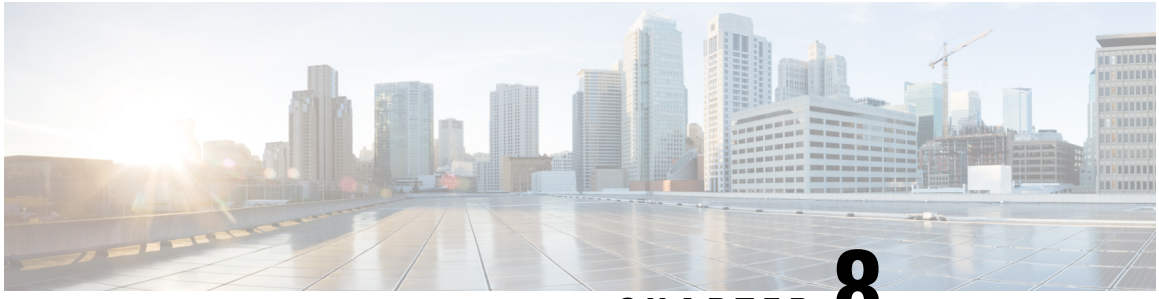
DWDM-SFP-3033=

DWDM-SFP-3112=

GLC-BX-D-I=

GLC-BX-U-I=

GLC-TE



CHAPTER 8

New Features for Cisco IOS XE 17.8.1

This chapter contains the following sections:

- Cellular Serviceability Enhancements, on page 63
- GNMI Broker (GNMIB) Update, on page 64
- gRPC Network Operations Interface Update, on page 64
- Raw Socket Feature Enhancement, on page 64
- SCADA Enhancement for TNB, on page 64

Cellular Serviceability Enhancements

Enhancements have been made for cellular and GPS features as follows:

Trigger points and debug code can be enabled via controller cellular CLIs for generating and trap the debug data automatically without manual intervention. The following CLI options are available:

```
(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 the following:

- 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.

GNMI Broker (GNMIB) Update

The GNMI Broker (GNMIB) has been extended to support the gRPC Network Operations Interface (gNOI) reset.proto service. This service provides functionality for restoring the device to its factory defaults via gRPC.

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. The additional steps below 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.

gRPC Network Operations Interface Update

gNOI is the gRPC Network Operations Interface. gNOI defines a set of gRPC-based microservices for executing operational commands and procedure on network devices, such as OS Install, Activate, and Verification.

Through gNOI os.proto will be possible to perform operating system related tasks such as OS activation, install, detailed overview, internal OS commands, and finally to output a summary of OS operations.

Furthermore, gNOI os.proto can also be used to display the gnmib detailed state, check the gnmib operational statistics, and also to output modifiers.

Raw Socket Feature Enhancement

This enhancement allows the user to input the maximum number of retries available to the write socket. The range of the number of retries goes from 1 to 1000. The default number of retries is 10. To accommodate this feature, a new CLI has been created, **raw-socket tcp max-retries <1-1000>**. <1-1000> is the maximum number of retries.

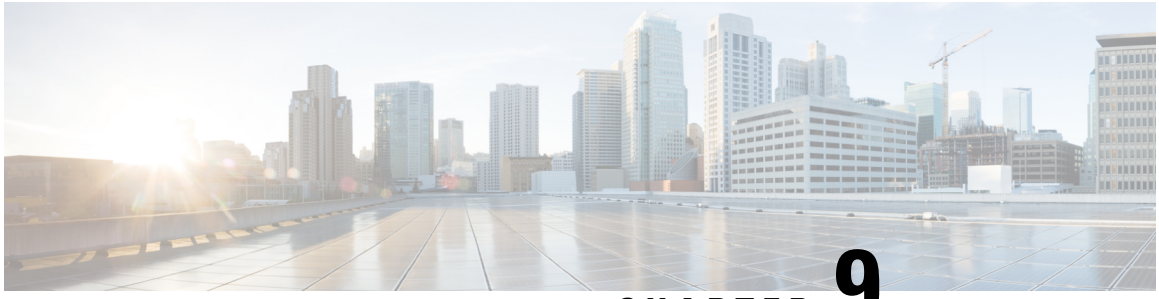
SCADA Enhancement for TNB

This enhancement provides compatibility with TNB's WG RTUs, including the following:

- 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.



CHAPTER 9

New Features For Cisco IOX XE 17.9.1

This chapter contains the following sections:

- Install Mode Support, on page 67
- Cellular Boot Time Improvements, on page 68
- IOS XE Downgrade Warning, on page 68
- SNMP Polling of Temperature OID, on page 69
- GPS Mode Enabled By Default, on page 70
- Packet Capture Support for CANBUS, on page 70
- GPS and Dead Reckoning Support for the J1939 Connector, on page 71

Install Mode Support

The following 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.

Cellular Boot Time Improvements

Numerous improvements have been made in the Cellular link up-time with IOS-XE release 17.9.1. In previous releases, the cellular interface was taking approximately 2:30 minutes to come up and pass traffic after the router booted up. The Cellular link up-time has been improved by approximately 20% in this release.

IOS XE Downgrade Warning

This feature will present 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. The downgrade operation will still be possible by ignoring the warning message presented to the user. Booting an image with the same or higher version of the running image is allowed without warning. The feature is only intended for images already loaded on the bootflash of the router, this means only for the **boot system flash** <*file_name*> CLI (excluding other sources/devices like ftp, mop, rpc, tftp, rom).

The following are examples of how the system compares versions:

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.

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** on an IR1101:

```
IR1101#show env

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)

GPS Mode Enabled By Default

In IOS XE versions prior to 17.9.1, GPS was enabled by defaut, however, GPS Mode was disabled by default. This required that the user perform an additional modem power-cycle after the router came up in order to use GPS.

Starting with IOS XE 17.9.1, GPS Mode will be enabled by default, and will be set to standalone mode. This will help reduce the 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:IR1101-4001#sh cellular 0/3/0 gps

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

Packet Capture Support for CANBUS

When enabled, this feature will capture packets sent and received on the IR1800 series CANBUS. Once captured, the data will be exported as a packet capture (PCAP) file to allow for further examination. The feature is configured in exec mode and is only temporary, meaning it is not permanent across a reboot/reload.

A file name is required for the capture. The default location for the capture file is at bootflash:/canbus_dumplogs. If the capture is started without specifying the file initially, or after the router is reloaded, you will get the following message when you check the status:

```
canbus packetdump file pcapfile path bootflash:/canbus_dumplogs/pcapfile didn't start
```

After stopping the capture, if you want to start the capture again without specifying the file name, the old specified name will be overwritten.

Use the following command to specify the name of the capture file:

```
Router#monitor canbus packetdump file <filename>
```

Note You do not need to specify the path, the only supported path is the default path **bootflash:/canbus_dumplogs**

Use the following command to start the capture using the specified <filename> from the command above:

```
Router#monitor canbus packetdump start
```

Use the following command to stop the capture:

```
Router#monitor canbus packetdump stop
```

Use the following command to check the status of the monitoring:

```
Router#show canbus packetdump
```

Command Examples

```
Router#monitor canbus packetdump ?
file CAN Bus interface packet capture destination file
start CAN Bus interface packet capture start
stop CAN Bus interface packet capture stop

Router#monitor canbus packetdump file canbusfile
Router#show canbus packetdump
canbus packetdump file canbusfile path bootflash:/canbus_dumplogs/canbusfile didn't start

Router#monitor canbus packetdump start
Router#show canbus packetdump
canbus packetdump file canbusfile path bootflash:/canbus_dumplogs/canbusfile started

Router#monitor canbus packetdump stop
Router#show canbus packetdump
canbus packetdump file canbusfile path bootflash:/canbus_dumplogs/canbusfile didn't start
```

GPS and Dead Reckoning Support for the J1939 Connector

Automotive Dead-Reckoning (DR) refers to the capability of a GNSS receiver to continue to navigate on an automotive platform when there are an insufficient number of GNSS satellite signals available. To do this, the receiver uses information provided by external sensors concerning the state of the vehicle in order to propagate the navigation solution.

Automotive DR requires information regarding the change in directional heading of the vehicle, which is provided by a three-axis digital gyroscope. Automotive DR also requires information about speed and direction of the vehicle. Speed is provided by an odometer (wheel tick) count, which is input into the IRM-GNSS-ADR pluggable module.

The automotive DR feature also accepts data from a three-axis digital accelerometer, which provides information that can be used to determine the orientation of the gyro when it is installed at a tilt angle. This information is also used to estimate elevation. The accelerometer is integrated within the sensor included inside the pluggable module.

Prior to the 17.9.1 release, only mode obdii was available. In 17.9.1, mode j1939 is added with the existing default mode obdii.

The J1939 connector is supported on heavy duty trucks, which provides speed and reverse status data to be fed into the GPS/DR module using J1939 protocol. It is configured through the command line interface under the controller.

Configuration

The following CLIs are available.

To show what is available for dead reckoning:

```
Router(config-controller)#dead-reckoning ?
enable enable GPS feature
mode DR mode configuration
nmea NMEA Configuration
```

To configure mode j1939:

```
Router(config)#controller Gps-Dr
Router(config-controller)#dead-reckoning mode j1939
```

To view the status:

```
Router#show platform hardware gps dead-reckoning
============================
DR Vehicle interface mode: J1939
GPS/DR Vendor Info: TELIT
GPS/DR module FW Version: V33-1.0.5-CLDR-4.7.10-N115R115-003291-3
DR Calibration Status:
DR is not calibrated
Odometer is not calibrated
Gain is not calibrated
Offset is not calibrated

CAN Bus Status:
CAN Bus Tx Count: 1874
CAN Bus Tx error Count: 0

CAN Bus Rx Count: 571
CAN NULL packet Bus RX Count: 0
CAN Bus Rx unsupported packet Count: 448

CAN Bus TX to DR Count: 123
CAN Bus TX to DR error Count: 0

DR data:
DR Sample TimeStamp in usec: 0
DR odometer count received from module: 0
DR odometer count sent to module: 1353
DR odometer is not valid from module
DR odometer delta count from module: 0
DR reverse status: 0

DR in use for location fix: No
time duration for loss of line of sight:
travel distance for loss of line of sight:
travel heading error at exit:
travel yaw error at exit:
travel gyro gain error at exit:
position error at exit:
position error ratio at exit:
position noise error at exit:
Raw Accel Data in X: 0
Raw Accel Data in Y: 0
Raw Accel Data in Z: 0
Raw Gyro Data in X: 0
Raw Gyro Data in Y: 0
Raw Gyro Data in Z: 0
```

CHAPTER 10

New Features for Cisco IOS XE 17.10.1a

This chapter contains the following sections:

- Digital Subscriber Line (DSL) SFP Support on the IR1800, on page 73
- vManage Support for the WP-WIFI6-x Module, on page 73
- Enable Secure Data Wipe Capabilities , on page 73
- Rawsocket Keepalive Configuration CLI, on page 74

Digital Subscriber Line (DSL) SFP Support on the IR1800

The IR1800 now supports the DSL SFP in the same manner as the IR1101. For complete details, see the Configuring Digital Subscriber Line (DSL) chapter in the IR1101 Configuration Guide.

vManage Support for the WP-WIFI6-x Module

This release will enable configuration and monitoring of the WP-WIFI6-x module through vManage from SDWAN. This applies only when the module is running in EWC mode.

For further information about vManage, see the product landing page here: https://www.cisco.com/site/us/en/products/networking/wan/vmanage/index.html

Enable Secure Data Wipe Capabilities

Secure data wipe is a Cisco wide initiative to ensure storage devices on all the IOS XE based platforms to be properly purged using NIST SP 800-88r1 compliant secure erase commands. Whenever possible, IoT platforms will leverage the corresponding ENG design and implementation available so far on their platforms.

This feature is supported on the following IoT platforms:

- IR1101
- IR1800
- IR8140
- ESR6300

When the enable secure data wipe is executed, the following will get wiped out:

- IR1101, IR1800, IR8140: NVRAM, rommon variables, bootflash, and msata
- ESR6300: NVARM, rommon variables, bootflash

The router will be in rommon prompt with default factory settings (baud rate 9600) after the command is executed. The bootflash will not get formatted until booting with IOS image thru usbflash or tftp download if the platform is supported.

Performing a Secure Data Wipe

To enable the feature, perform the following:

```
Router#factory-reset all secure
The factory reset operation is irreversible for securely reset all. Are you sure? [confirm]Y
```

Important This operation may take hours. Please do not power cycle.

To check the log after the command is executed, and booting up IOS XE, perform the following:

```
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.
```

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.

CHAPTER 11

Managing Configuration Files

This chapter contains the following sections:

- Understanding Configuration Files, on page 77
- Finding the Software Version, on page 78
- Managing and Configuring a Consolidated Package Using copy and boot Commands, on page 78
- Upgrading the Router Image through the WebUI, on page 80

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
```

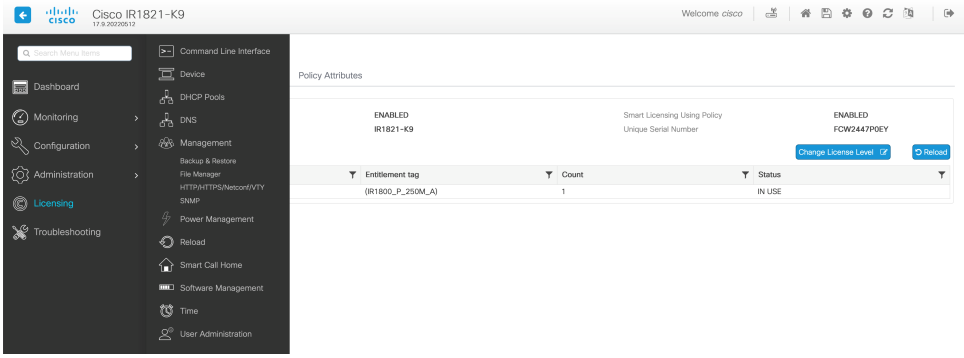
Copy the running configuration and save it. Then when reloading the router, it restarts with the saved configuration.

```
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 ...
```

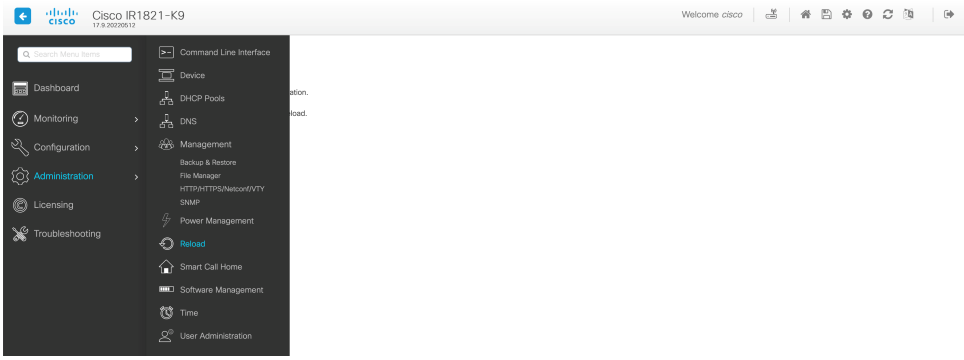
Upgrading the Router Image through the WebUI

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.

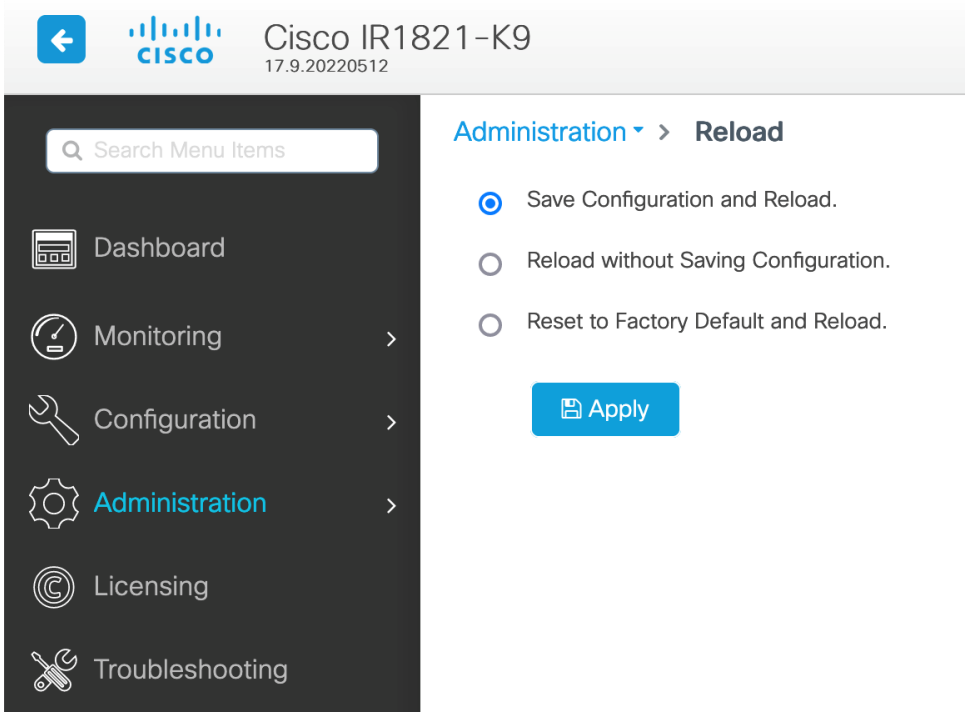
After you launch the WebUI, go to the **Administration** tab.



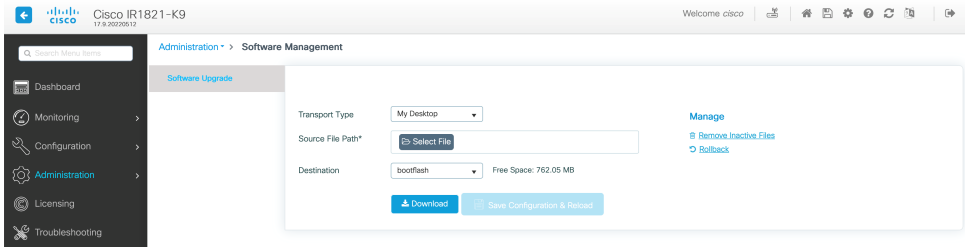
Reload the router by selecting **Administration** > **Reload**.



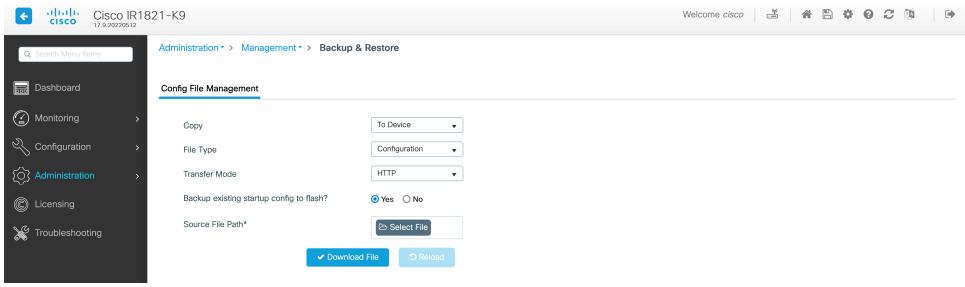
Select your option from the choices, then click **Apply**.



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 12

Cisco IOS XE Installation Methods

This chapter contains the following sections:

- Bundle Mode versus Install Mode, on page 83
- Installing the Software using install Commands, on page 83
- Restrictions for Installing the Software Using install Commands, on page 84
- Information About Installing the Software Using install Commands, on page 84
- Configuration Examples, on page 93
- Troubleshooting Software Installation Using install Commands, on page 100

Bundle Mode versus 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.

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.

Installing the Software using install Commands

From Cisco IOS XE 17.9.1, Cisco IoT routers are shipped in install mode by default. Users can boot the platform, and upgrade or downgrade to Cisco IOS XE software versions using a set of **install** commands.

Restrictions for Installing the 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. Other management tools such as IOT OD and FND will be available in a future release.

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

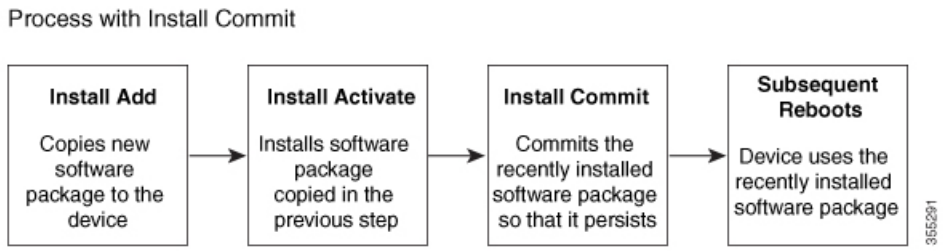
Table 9: Bundle Mode vs Install Mode

Bundle Mode	Install Mode
This mode provides a consolidated boot process, using local (hard disk, flash) or remote (TFTP) .bin image.	This mode uses the local (bootflash) packages.conf file for the boot process.
This mode uses a single .bin file.	.bin file is replaced with expanded .pkg files in this mode.
CLI: `Router(config)#boot system bootflash:<filename>`	CLI: `#install add file bootflash: [activate commit]`
To upgrade in this mode, point the boot system to the new image.	To upgrade in this mode, use the **install** commands.
Image Auto-Upgrade: When a new Field-Replaceable Unit (FRU) is inserted in a modular chassis, manual intervention is required to get the new FRU running with the same version as the active FRUs.	Image Auto-Upgrade: When a new FRU is inserted in a modular chassis, the joining FRU is auto-upgraded to the image version in sync with the active FRUs.
Rollback: Rollback to the previous image with multiple Software Maintenance Updates (SMUs) may require multiple reloads.	Rollback: Enables rollback to an earlier version of Cisco IOS XE software, including multiple patches in single reload.

Install Mode Process Flow

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

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



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 the following command:

```
IR1831#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
```

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

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

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

The following set of install commands is available:

Table 10: List of install Commands

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

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

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

Command	Syntax	Purpose
install deactivate	**install deactivate file** <*filename*>	Removes a package from the platform repository. This command is supported only for SMUs. • Use this command with the **prompt-level none** keyword to automatically ignore any confirmation prompts.

The following show commands are also available:

Table 11: List of show Commands

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

Booting the Platform in Install Mode

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

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

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:IR1101;
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#
```

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

Note

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

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

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

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

Procedure

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

Three-Step Installation

Note

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

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

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

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	**install add file location:** *filename* **Example:** `Device#install add file bootflash:<router_image>.SSA.bin`	Copies the software install package from a remote location (through FTP, HTTP, HTTPs, or TFTP) to the platform, and extracts the individual components of the .package file into subpackages and packages.conf files.
Step 3	**show install summary** **Example:** `Device#show install summary`	(Optional) Provides an overview of the image versions and their corresponding install state.
Step 4	**install activate auto-abort-timer** <*time*> **Example:** `Device# install activate auto-abort-timer 120`	Activates the previously added package and reloads the platform. • When doing a full software install, do not provide a package filename. • In the three-step variant, **auto-abort-timer** starts automatically with the **install activate** command; the default for the timer is 120 minutes. If the **install commit** command is not run before the timer expires, the install process is automatically terminated. The platform reloads and boots up with the last committed version.
Step 5	**install abort** **Example:** `Device#install abort`	(Optional) Terminates the software install activation and returns the platform to the last committed version. • Use this command only when the image is in activated state, and not when the image is in committed state.
Step 6	**install commit** **Example:** `Device#install commit`	Commits the new package installation and makes the changes persistent over reloads.
Step 7	**install rollback to committed** **Example:** `Device#install rollback to committed`	(Optional) Rolls back the platform to the last committed state.
Step 8	**install remove** {**file** *filesystem: filename* | **inactive**} **Example:** `Device#install remove inactive`	(Optional) Deletes software installation files. • **file**: Deletes a specific file • **inactive**: Deletes all the unused and inactive installation files.

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

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

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

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

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

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

Terminating a Software Installation

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

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

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

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

Configuration Examples

This section shows examples of using install commands.

One Step Installation

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

```
Router# install add file
flash:ir1101-universalk9.BLD_POLARIS_DEV_LATEST_20220421_143208.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.BLD_POLARIS_DEV_LATEST_20220421_143208.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_iot.BLD_POLARIS_DEV_LATEST_20220421_143208.SSA.pkg
/flash/ir1101-rpboot.BLD_POLARIS_DEV_LATEST_20220421_143208.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
--------------------------------------------------------------------------------
```

Three Step Installation

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

Install Add

```
Router# install add file flash:ir1101-universalk9.17.09.01prd1.SPA.bin
install_add: START Tue May 31 01:35:40 UTC 2022
install_add: Adding IMG
--- Starting initial file syncing ---
Copying flash:ir1101-universalk9.17.09.01prd1.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/ir1101-universalk9.17.09.01prd1.SPA.bin Tue May 31 01:37:10
UTC 2022
Router#

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

Install Activate

```
Router#install activate
install_activate: START Tue May 31 01:37:14 UTC 2022
install_activate: Activating IMG
Following packages shall be activated:
/flash/ir1101-mono-universalk9_iot.17.09.01prd1.SPA.pkg
/flash/ir1101-rpboot.17.09.01prd1.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.

IR1101-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!

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

Install Commit

```
Router#install commit
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

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

Showing the Installed Packages

```
Router# show install package flash:ir1101-universalk9.17.09.01prd1.SPA.bin
  Package: ir1101-universalk9.17.09.01prd1.SPA.bin
    Size: 674114352
    Timestamp:
  Canonical path: /flash1/ir1101-universalk9.17.09.01prd1.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: 17.09.01prd1
    CardTypes:

  Package is bootable from media and tftp.
  Package contents:

  Package: ir1101-mono-universalk9_iot.17.09.01prd1.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: 17.09.01prd1
      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
--------------------------------------------------------------------------------
```

Showing 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
```

Removing 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.17.09.01prd1.SPA.pkg File is in use, will not
 delete.
    [R0]: /flash/ir1101-universalk9.17.09.01prd1.SPA.conf File is in use, will not delete.
    [R0]: /flash/ir1101-rpboot.17.09.01prd1.SPA.pkg File is in use, will not delete.

The following files will be deleted:
    [R0]: /flash/ir1101-universalk9.17.09.01prd1.SPA.bin
   [R0]: /flash/ir1101-mono-universalk9_iot.BLD_POLARIS_DEV_LATEST_20220421_143208.SSA.pkg

    [R0]: /flash/ir1101-universalk9.BLD_POLARIS_DEV_LATEST_20220421_143208.SSA.conf
    [R0]: /flash/ir1101-rpboot.BLD_POLARIS_DEV_LATEST_20220421_143208.SSA.pkg

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


Deleting file /flash/ir1101-universalk9.17.09.01prd1.SPA.bin ... done.
Deleting file
/flash/ir1101-mono-universalk9_iot.BLD_POLARIS_DEV_LATEST_20220421_143208.SSA.pkg ... done.
Deleting file /flash/ir1101-universalk9.BLD_POLARIS_DEV_LATEST_20220421_143208.SSA.conf ...
 done.
Deleting file /flash/ir1101-rpboot.BLD_POLARIS_DEV_LATEST_20220421_143208.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
```

Troubleshooting Software Installation Using install Commands

Problem Troubleshooting the software installation

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

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

Problem Other installation issues

Solution Use the following commands to resolve installation issue:

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

CHAPTER 13

Installing the Software

This chapter contains the following sections:

- Installing the Software, on page 101
- How to Install the Software for Cisco IOS XE , on page 102
- ROMMON Images, on page 104
- File Systems, on page 104
- Option to Enable or Disable USB Access, on page 105
- Autogenerated File Directories and Files, on page 106
- Flash Storage, on page 107
- LED Indicators, on page 107
- Related Documentation, on page 108

Installing the Software

Installing software on the router involves installing a consolidated package (bootable image). This consists of a bundle of subpackages (modular software units), with each subpackage controlling a different set of functions.

It is better to upgrade software in a planned period of maintenance when an interruption in service is acceptable. The router needs to be rebooted for a software upgrade to take effect.

Licensing

This section contains the following:

Cisco Software Licensing

Cisco software licensing consists of processes and components to activate Cisco IOS software feature sets by obtaining and validating Cisco software licenses.

You can enable licensed features and store license files in the bootflash of your router. Licenses pertain to consolidated packages, technology packages, or individual features.

The IR1800 uses Enhanced Smart Licensing, which is discussed in detail in the next chapter.

Consolidated Packages

To obtain software images for the router, go to: https://software.cisco.com/download/home/286200112

Note All of the IOS-XE feature set may not apply to the IR1800. Some features may not have been implemented yet, or are not appropriate for this platform.

An image-based license is used to help bring up all the subsystems that correspond to a license. This license is enforced only at boot time.

One of the following image-based licenses can be pre-installed on the IR1800 router:

- Network-Essentials
- Network-Advantage

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

Network-Essentials

The **Network-Essentials** technology package includes the baseline features. It also supports security features.

The **Network-Essentials_npe** technology package (npe = No Payload Encryption) includes all the features in the Network-Essentials technology package without the payload encryption functionality. This is to fulfill export restriction requirements. The Network-Essentials_npe is available only in the Network-Essentials_npe image. The difference in features between the Network-Essentials package and the Network-Essentials_npe package is therefore the set of payload encryption features such as IPsec and Secure VPN.

Network-Advantage

The **Network-Advantage** technology package includes all crypto features.

The **Network-Advantage_npe** package (npe = No Payload Encryption) includes all the features in the **Network-Advantage** technology package without the payload-encryption functionality. This is to fulfill export restriction requirements. The **Network-Advantage_npe** package is available only in the **Network-Advantage_npe** image. The difference in features between the **Network-Advantage** package and the **Network-Advantage_npe** package is therefore the set of payload-encryption-enabling features such as IPsec and Secure VPN.

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 described in this section to use the software from a consolidated package or an individual package.

Installing the Cisco IOS XE Release

When the device boots up with Cisco IOS XE image for the first time, the device checks the installed version of the ROMMON, and upgrades if the system is running an older version. During the upgrade, do not power cycle the device. The system automatically power cycles the device after the new ROMMON is installed. After the installation, the system will boot up with the Cisco IOS XE image as normal.

Note When the device boots up for first time and if the device requires an upgrade, the entire boot process may take several minutes. This process will be longer than a normal boot due to the ROMMON upgrade.

The following example illustrates the boot process of a consolidated package:

```
Router# configure terminal
 Router(config)#boot system bootflash:/ir1800-universalk9.17.06.01prd18.SPA.bin
Router(config)#config-register 0x2102
Router(config)#exit
*Nov  7 00:07:06.784: %SYS-5-CONFIG_I: Configured from console by console

Router#
Router#show run | inc license
license udi pid IR1800-K9 sn FCW2150TH0F
license boot level network-advantage
Router#

Router#reload ?
  /noverify  Don't verify file signature before reload.
  /verify    Verify file signature before reload.
  at         Reload at a specific time/date
  cancel     Cancel pending reload
  in         Reload after a time interval
  pause      Pause during reload
  reason     Reload reason
  <cr>       <cr>

Router#reload /verify

System configuration has been modified. Save? [yes/no]: yes
Building configuration...

[OK]
*Nov  7 00:08:48.101: %SYS-2-PRIVCFG_ENCRYPT: Successfully encrypted private config file
Verifying file integrity of bootflash:/ir1800-universalk9.17.06.01prd18.SPA.bin...........
.....................................

Embedded Hash   SHA1 : B0315BDC4F545D624BB128CE0FFAA468E6EF7587
Computed Hash   SHA1 : B0315BDC4F545D624BB128CE0FFAA468E6EF7587
Starting image verification
Hash Computation:    100%Done!
Computed Hash   SHA2: 03febcc07fbeadeed664f2f5ef87f6c3
                      5b343e6f7aecdd70e50e5203909aec8f
                      3d276529d2a6af6859d4c77237f812d5
                      0da93678edc942c8874edca2d5224101

Embedded Hash   SHA2: 03febcc07fbeadeed664f2f5ef87f6c3
                      5b343e6f7aecdd70e50e5203909aec8f
                      3d276529d2a6af6859d4c77237f812d5
                      0da93678edc942c8874edca2d5224101

Digital signature successfully verified in file bootflash:/ir1800-universalk9.16.10.01.SPA.bin
```

```
Signature Verified

Proceed with reload? [confirm]<Enter>

*Jul  9 06:43:37.910: %SYS-5-RELOAD: Reload requested by console. Reload Reason: Reload
Command. Jul  9 14:43:59.134: %PMAN-5-EXITACTION: R0/0: pvp: Process manager is exiting:
process exit with reload chassis code

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



Press RETURN to get started!
```

ROMMON Images

A ROMMON image is a software package used by ROM Monitor (ROMMON) software on a router. The software package is separate from the consolidated package normally used to boot the router.

An independent ROMMON image (software package) may occasionally be released and the router can be upgraded with the new ROMMON software. For detailed instructions, see the documentation that accompanies the ROMMON image.

Note A new version of the ROMMON image is not necessarily released at the same time as a consolidated package for a router.

File Systems

The following table provides a list of file systems that can be seen on the Cisco IR1800 router.

Table 12: 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.

File System	Description
usbflash0:	The Universal Serial Bus (USB) flash drive file systems. **Note** The USB flash drive file system is visible only if a USB drive is installed in the usb port. **Note** Only Cisco supported USB flash drives may be used. A list of supported devices are found in the Hardware Installation Guide

Use the **?** help option if you find a file system that is not listed in the table above.

Option to Enable or Disable USB Access

USB flash drives offer inexpensive and easy storage space for the routers to store the images, configuration files and other files.

Note The IR1800 supports ext2 and vfat file systems for USB flash drives. Only Cisco approved USB Flash drives may be used.

The IR1800 supports hot plug/unplug of USB flash drives. To access the USB flash drive, insert the device into Router's USB interface. Once the USB is recognized, an alert message is seen on the console:

```
Aug  1 11:08:53.198 PDT: %IOSD_INFRA-6-IFS_DEVICE_OIR: Device usbflash0 added
```

After this message is seen, the USB flash drive is accessible. Users can access the USB contents using the **dir usbflash0:** command:

```
Device#dir usbflash0:
Directory of usbflash0:/
    5  drwx              512  Aug 23 2019 10:42:18 -07:00  System Volume Information
    6  -rwx               35  Aug 27 2019 17:40:38 -07:00  test.txt
206472192 bytes total (206470144 bytes free)
Device#
```

Contents can be copied to and from the USB flash drive using the copy command. Once the copy is complete, a log message showing number of bytes copied is displayed.

```
Device#copy flash:test.txt usbflash0:
Destination filename [test.txt]? <Enter>
Copy in progress...C
35 bytes copied in 0.020 secs (1750 bytes/sec)
Device#
```

While hot plug/unplug of a USB flash drive is supported, the functionality comes with security vulnerabilities. To prevent users from copying sensitive information to the USB flash drive, USB enable/disable functionality has been added.

By default, the USB flash drive is enabled. If a user wishes to disable USB, they can do so using the disable command:

```
Device# config terminal
```

```
Device(config)#platform usb disable

Device(config)#end
```

Once the USB flash drive has been disabled, the file system is not shown on the Device and syslog messages will not be displayed when the USB is inserted. Users will not be able to access the contents of the USB.

For example:

```
Device#dir usbflash0:
dir usbflash0:
    ^
% Invalid input detected at '^' marker.
Device#
```

The USB is enabled by issuing a '**no**' with the disable command:

```
Device#config terminal

Device(config)#no platform usb disable
Device(config)#end
```

The USB status can be displayed using the following command:

```
Device#show platform usb status
USB enabled
Device#
```

The USB port could be considered a potential security risk. If you wish to disable the USB port, use these steps:

```
Configure terminal
platform usb disable
exit

show platform usb
```

Autogenerated File Directories and Files

This section discusses the autogenerated files and directories that can be created, and how the files in these directories can be managed.

Table 13: 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.

File or Directory	Description
managed directory	This directory is created on bootup if a system check is performed. Its appearance is completely normal and does not indicate any issues with the router.
tracelogs directory	The storage area for trace files. Trace files are useful for troubleshooting. If the Cisco IOS process fails, for instance, users or troubleshooting personnel can access trace files using diagnostic mode to gather information related to the Cisco IOS failure. Trace files, however, are not a part of router operations, and can be erased without impacting the router's performance.

Important Notes About Autogenerated Directories

Important information about autogenerated directories include:

- Autogenerated files on the bootflash: directory should not be deleted, renamed, moved, or altered in any way unless directed by Cisco customer support.

Note Altering autogenerating files on the bootflash: may have unpredictable consequences for system performance.

- Crashinfo files and files in the core and tracelogs directory can be deleted.

Flash Storage

Subpackages are installed to local media storage, such as flash. For flash storage, use the **dir bootflash:** command to list the file names.

Note Flash storage is required for successful operation of a router.

LED Indicators

For information on LEDs on the router, see "LED Indicators" in the "Product Overview" section of the Hardware Installation Guide.

To monitor the LED status of the system, the alarm and interface ports, the show LED command line is supported in IOS mode.

```
Router# show LED
SYSTEM LED : Green

GigabitEthernet0/0/0 LED : On
GigabitEthernet0/1/0 LED : Off
```

```
GigabitEthernet0/1/1 LED : Off
GigabitEthernet0/1/2 LED : Off
GigabitEthernet0/1/3 LED : Off

*Cellular 0/4*
LTE module Enable LED : Green
LTE module SIM 0 LED : Green
LTE module SIM 1 LED : Yellow
LTE module GPS LED : Off
LTE module RSSI 0 LED : On
LTE module RSSI 1 LED : On
LTE module RSSI 2 LED : On
LTE module RSSI 3 LED : On

*Cellular 0/5*
LTE module Enable LED : Green
LTE module SIM 0 LED : Green
LTE module SIM 1 LED : Off
LTE module GPS LED : Off
LTE module RSSI 0 LED : On
LTE module RSSI 1 LED : On
LTE module RSSI 2 LED : On
LTE module RSSI 3 LED : Off
Router#
```

Related Documentation

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

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

CHAPTER 14

Software Maintenance Upgrade (SMU)

This section contains the following:

- Software Maintenance Upgrade (SMU), on page 109
- SMU Work-flow and Basic Requirements, on page 110
- SMU Example, on page 110
- Installing a Patch Image, on page 110
- Uninstalling the Patch Image, on page 113

Software Maintenance Upgrade (SMU)

The Software Maintenance Upgrade (SMU) is a package that can be installed on a system to provide a patch fix or security resolution to a released image for a specific defect in order to respond to immediate issues. It does not contain new features.

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

Some of the caveats of the SMU are:

- Provided on a per release, per component basis and is specific to the platform. SMU versions are synchronized to the package major, minor, and maintenance versions they upgrade.
- SMUs are not an alternative to maintenance releases. All defects fixed by SMUs are then automatically integrated into the upcoming maintenance releases.
- The Cisco IOS XE platform internally validates the SMU compatibility and does not allow you to install non-compatible SMUs. This is based on rules/limitations for a SMU change-set.
- An SMU provides a significant benefit over classic IOS software as it allows you to address the network issue quickly while reducing the time and scope of the testing required.
- SMU is a method to fix bugs in an existing release, and allows the application of a PSIRT fix in an existing release.
- SMU is NOT an upgrade path from release X to maintenance release X.1

- SMU is NOT an upgrade path from release X to release Y

The device only supports “Hot Patching”. This means:

- The running image is modified in-place or in-service
- This avoids downtime and interruption of service
- The updated code to fix the defect is written in a different location, and where the patch redirects the program run

SMU Work-flow and Basic Requirements

The work-flow for the patch requires that you complete the following sequence of operation in exec mode:

1. Addition of the SMU to the file system
2. Activation of the SMU onto the system
3. Committing the SMU change
4. Removal and Uninstallation of the SMU

The basic requirements for SMU are:

- The image where the defect was discovered
- The patch file that contains the fix for the defect must be formatted as ir1800-image_name.release_version.CSCxxyyyyy.SPA.smu.bin

SMU Example

This section shows an example of a patch created as a test. Your patch will have a name associated with a CDET to be installed as a fix.

Installing a Patch Image

Perform the following steps to install the patch image:

Procedure

Step 1 Show a standard command.

```
Router#show power
Main PSU :
    Total Power Consumed: 11.37 Watts
    Configured Mode : N/A
    Current runtime state same : N/A
    PowerSupplySource : External PS
POE Module :
```

```
    Configured Mode : N/A
    Current runtime state same : N/A
    Total power available : 30 Watts
Router#
```

Step 2 Add the image.

```
Router# install add file
bootflash:ir1800-universalk9.2020-08-06_10.38.0.CSCxx12345.SSA.smu.bin
install_add: START Thu Aug  6 11:52:52 PDT 2020
cat: /tmp/patch/patch.sta: No such file or directory
install_add: Adding SMU
install_add: Checking whether new add is allowed ....

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

SUCCESS: install_add  Thu Aug  6 11:53:31 PDT 2020

Router#
```

Step 3 Activate the patch image.

```
Router# install activate file
bootflash:ir1800-universalk9.2020-08-06_10.38.0.CSCxx12345.SSA.smu.bin
install_activate: START Thu Aug  6 11:53:59 PDT 2020

System configuration has been modified.
Press Yes(y) to save the configuration and proceed.
Press No(n) for proceeding without saving the configuration.
Press Quit(q) to exit, you may save configuration and re-enter the command. [y/n/q] y
Building configuration...
[OK]Modified configuration has been saved
install_activate: Activating SMU
Executing pre scripts....
Executing pre sripts done.

--- Starting SMU Activate operation ---
Performing SMU_ACTIVATE on Active/Standby
/usr/sbin/kgv_update: kgv_update [
/flash1/ir1800-universalk9.2020-08-06_10.38.0.CSCxx12345.SSA.smu.bin, NOT slot local is ics
 ] continuing ....
/usr/sbin/kgv_update: Signature validated for
/flash1/ir1800-universalk9.2020-08-06_10.38.0.CSCxx12345.SSA.smu.bin
/usr/sbin/kgv_update: TAM hash len:32
val:4407CBB447F0EEE3B12120D902F48FBA1C0D4900EED1FB614441198BE2302934
/usr/sbin/kgv_update: PCR8 before extend ctr:2
0817449B454BF036AF9D593D726D94D8942C50A9FFE93278FDA78EA62F2989F2
/usr/sbin/kgv_update: PCR8 after extend ctr:3
EF5F579FCDFA989D044296F0584B99F719F2B6215895524B5E8AD55DF5671560
/usr/sbin/kgv_update: PCR extend successful
/usr/sbin/kgv_update: Chasfs updated for
name:ir1800-universalk9.2020-08-06_10.38.0.CSCxx12345.SSA.smu.bin
hash:975352C1562A492D582D09D5BB91230863F6CC18E6F9C0FB512AF27CC0C77E2C05F29596AD34AD7808C9B39EC23D4412F0D3AFA707BC906FE03D554A845F42D4
/usr/sbin/kgv_update: Update successful for
ir1800-universalk9.2020-08-06_10.38.0.CSCxx12345.SSA.smu.bin
  [1] SMU_ACTIVATE package(s) on R0
```

```
  [1] Finished SMU_ACTIVATE on R0
Checking status of SMU_ACTIVATE on [R0]
SMU_ACTIVATE: Passed on [R0]
Finished SMU Activate operation
SUCCESS: install_activate /flash1/ir1800-universalk9.2020-08-06_10.38.0.CSCxx12345.SSA.smu.bin
 Thu Aug  6 11:55:14 PDT 2020
Router#
```

Step 4 Commit the installation.

```
Router# install commit
install_commit: START Thu Aug  6 11:55:29 PDT 2020
install_commit: Committing SMU
Executing pre scripts....
Executing pre sripts done.
--- Starting SMU Commit operation ---
Performing SMU_COMMIT on Active/Standby
  [1] SMU_COMMIT package(s) on R0
  [1] Finished SMU_COMMIT on R0
Checking status of SMU_COMMIT on [R0]
SMU_COMMIT: Passed on [R0]
Finished SMU Commit operation

SUCCESS: install_commit /flash1/ir1800-universalk9.2020-08-06_10.38.0.CSCxx12345.SSA.smu.bin
 Thu Aug  6 11:56:08 PDT 2020
Router#
```

Step 5 Show the status summary of the installation procedure.

```
Router# show install summary
[ R0 ] Installed Package(s) Information:
State (St): I - Inactive, U - Activated & Uncommitted,
            C - Activated & Committed, D - Deactivated & Uncommitted
--------------------------------------------------------------------------------
Type  St   Filename/Version
--------------------------------------------------------------------------------
IMG   C    17.04.01.0.118999
SMU   C    flash:ir1800-universalk9.2020-08-06_10.38.0.CSCxx12345.SSA.smu.bin

--------------------------------------------------------------------------------
Auto abort timer: inactive
--------------------------------------------------------------------------------
Router#
```

Step 6 Verify the result of the patch by showing the same command.

```
Router#show power
Main PSU :
    Total Power Consumed: 11.04 Watts
Device HOT SMU works!

    Configured Mode : N/A
    Current runtime state same : N/A
    PowerSupplySource : External PS
POE Module :
    Configured Mode : N/A
    Current runtime state same : N/A
    Total power available : 0 Watts
Router#
```

Uninstalling the Patch Image

There are two methods to remove or uninstall the patch image.

- Restoring the image to its original version by using the following command:
  - **install rollback to base**
- Specific removal of a patch by using the following commands in sequence:
  - **install deactivate file flash:**<*file*>
  - **install commit**
  - **install remove file flash:**<*file*>

Uninstalling the Patch Image Using Rollback

This section shows an example of using the rollback method.

Show what patches are installed:

```
Router# show install summary
[ R0 ] Installed Package(s) Information:
State (St): I - Inactive, U - Activated & Uncommitted,
            C - Activated & Committed, D - Deactivated & Uncommitted
--------------------------------------------------------------------------------
Type  St   Filename/Version
--------------------------------------------------------------------------------
IMG   C    17.04.01.0.118999
SMU   C    flash:ir1800-universalk9.2020-08-06_10.38.0.CSCxx12345.SSA.smu.bin

--------------------------------------------------------------------------------
Auto abort timer: inactive
--------------------------------------------------------------------------------
Router#
```

The following commands are available:

```
Router# install ?
  abort             Abort the current install operation
  activate          Activate an installed package
  add               Install a package file to the system
  auto-abort-timer  Install auto-abort-timer
  commit            Commit the changes to the loadpath
  deactivate        Deactivate an install package
  label             Add a label name to any installation point
  prepare           Prepare package for operation
  remove            Remove installed packages
  rollback          Rollback to a previous installation point
Router# install rollback to ?
  base       Rollback to the base image
  committed  Rollback to the last committed installation point
  id         Rollback to a specific install point id
  label      Rollback to a specific install point label
```

The **install rollback to base** command removes the entire patch and returns to the base image version with the found defect.

```
Router# install rollback to base
install_rollback: START Thu Aug  6 12:04:04 PDT 2020
install_rollback: Rolling back SMU
Executing pre scripts....
Executing pre sripts done.

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

CSCxx12345:SUCCESS
SUCCESS: install_rollback
/flash1/ir1800-universalk9.2020-08-06_10.38_shchang2.0.CSCxx12345.SSA.smu.bin Thu Aug  6
12:04:57 PDT 2020
Router#
```

Show what patches are installed:

```
Router# show install summary
[ R0 ] Installed Package(s) Information:
State (St): I - Inactive, U - Activated & Uncommitted,
            C - Activated & Committed, D - Deactivated & Uncommitted
--------------------------------------------------------------------------------
Type  St   Filename/Version
--------------------------------------------------------------------------------
IMG   C    17.04.01.0.118999

--------------------------------------------------------------------------------
Auto abort timer: inactive
--------------------------------------------------------------------------------
Router#
```

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

Verify the result of the patch by showing the same command.

```
Router#show power
Main PSU :
    Total Power Consumed: 11.98 Watts
    Configured Mode : N/A
    Current runtime state same : N/A
    PowerSupplySource : External PS
POE Module :
    Configured Mode : N/A
    Current runtime state same : N/A
    Total power available : 30 Watts
Router#
```

Uninstalling the Patch Image Using Deactivate, Commit, and Remove

In the following sequence, there are two patches installed on the device. CSCvq11111 and CSCvt22222 Only CSCvt22222 will be removed.

Show what patches are installed.

```
Router# show install summary
[ R0 ] Installed Package(s) Information:
State (St): I - Inactive, U - Activated & Uncommitted,
            C - Activated & Committed, D - Deactivated & Uncommitted
--------------------------------------------------------------------------------
Type  St   Filename/Version
--------------------------------------------------------------------------------
SMU   C    /flash1/ir1800-universalk9.<release>.CSCvq11111.SPA.smu.bin
SMU   C    /flash1/ir1800-universalk9.<release>.CSCvt22222.SPA.smu.bin
IMG   C    17.04.1
```

Procedure

Step 1 Deactivate the patch.

```
Router# install deactivate file flash:ir1800-universalk9.<release>.CSCvt22222.SPA.smu.bin
install_deactivate: START Fri Apr 24 22:54:10 UTC 2020
install_deactivate: Deactivating SMU
Executing pre scripts....
Executing pre sripts done.

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

SUCCESS: install_deactivate /flash1/ir1800-universalk9.<release>.CSCvt22222.SPA.smu.bin Fri
 Apr 24 22:54:49 UTC 2020
```

Show what patches are installed:

```
Router# show install summary
[ R0 ] Installed Package(s) Information:
State (St): I - Inactive, U - Activated & Uncommitted,
            C - Activated & Committed, D - Deactivated & Uncommitted
--------------------------------------------------------------------------------
Type  St   Filename/Version
--------------------------------------------------------------------------------
SMU   C    /flash1/ir1800-universalk9.<release>.CSCvt11111.SPA.smu.bin
SMU   D    /flash1/ir1800-universalk9.<release>.CSCvt22222.SPA.smu.bin
IMG   C    17.01.1
```

Step 2 Commit the action.

```
Router# install commit
install_commit: START Fri Apr 24 22:56:11 UTC 2020
install_commit: Committing SMU
```

```
*Apr 24 22:56:15.169: %INSTALL-5-INSTALL_START_INFO: R0/0: install_engine: Started install
 commitExecuting pre scripts....
Executing pre sripts done.
--- Starting SMU Commit operation ---
Performing SMU_COMMIT on Active/Standby
  [R0] SMU_COMMIT package(s) on R0
  [R0] Finished SMU_COMMIT on R0
Checking status of SMU_COMMIT on [R0]
SMU_COMMIT: Passed on [R0]
Finished SMU Commit operation

SUCCESS: install_commit /flash1/ir1800-universalk9.<release>.CSCvt22222.SPA.smu.bin Fri Apr
 24 22:56:32 UTC 2020

*Apr 24 22:56:33.342: %INSTALL-5-INSTALL_COMPLETED_INFO: R0/0: install_engine: Completed
install commit SMU
```

Show what patches are installed:

```
Router# show install summary
[ R0 ] Installed Package(s) Information:
State (St): I - Inactive, U - Activated & Uncommitted,
            C - Activated & Committed, D - Deactivated & Uncommitted
--------------------------------------------------------------------------------
Type  St   Filename/Version
--------------------------------------------------------------------------------
SMU   C    /flash1/ir1800-universalk9.<release>.CSCvt11111.SPA.smu.bin
SMU   I    /flash1/ir1800-universalk9.<release>.CSCvt22222.SPA.smu.bin
IMG   C    <release>
```

Step 3 Remove the patch.

```
Router# install remove file flash:ir1800-universalk9.<release>.CSCvt22222.SPA.smu.bin
install_remove: START Fri Apr 24 22:57:17 UTC 2020

*Apr 24 22:57:20.775: %INSTALL-5-INSTALL_START_INFO: R0/0: install_engine: Started install
 remove flash:ir1800-universalk9.<release>.CSCvt22222.SPA.smu.bininstall_remove: Removing
SMU
Executing pre scripts....
Executing pre scripts done.

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

SUCCESS: install_remove /flash1/ir1800-universalk9.<release>.CSCvt22222.SPA.smu.bin Fri Apr
 24 22:57:34 UTC 2020

*Apr 24 22:57:34.902: %INSTALL-5-INSTALL_COMPLETED_INFO: R0/0: install_engine: Completed
install remove flash:ir1800-universalk9.<release>.CSCvt22222.SPA.smu.bin
```

Show what patches are installed:

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

```
Type  St   Filename/Version
-------------------------------------------------------------------------------
SMU   C    /flash1/ir1800-universalk9.<release>.CSCvt11111.SPA.smu.bin
IMG   C    <release>
```

CHAPTER 15

Smart Licensing Using Policy (SLP)

This section contains the following:

- SLP Overview, on page 119
- Customer Topologies, on page 121
- License Installation Procedure - Full Offline Access Topology, on page 122
- License Installation Procedure - CSLU has No Access to CSSM, on page 127

SLP Overview

Smart Licensing Using Policy (SLP), previously known as Smart Licensing Enhanced (SLE), is the default mode starting with IOS-XE release 17.3.2. SLE replaces Smart Software Licensing. The IR1800 series only supports SLP. Some of the feature differences are:

- An Authorization Code is required only for export control requirement
- Throughput greater than 250MB requires an HSEC license
- No more EVAL licenses. Authorized status has changed to In Use or Not In Use with an Enforcement Type class.
- Cisco Smart Licensing Utility (CSLU) is a new tool interfacing between the devices and Cisco Smart Software Manager (CSSM) in specific customer topologies.

License Enforcement Types

A given license belongs to one of three enforcement types. The enforcement type indicates if the license requires authorization before use, or not.

- Unenforced or Not Enforced

The vast majority of licenses belong to this enforcement type. Unenforced licenses do not require authorization before use in air-gapped networks, or registration, in connected networks. The terms of use for such licenses are as per the end user license agreement (EULA).

- Enforced

Licenses that belong to this enforcement type require authorization before use. The required authorization is in the form of an authorization code, which must be installed in the corresponding product instance.

An example of an enforced license is the Media Redundancy Protocol (MRP) Client license, which is available on Industrial Ethernet Switches.

- Export-Controlled

Licenses that belong to this enforcement type are export-restricted by U.S. trade-control laws and these licenses require authorization before use. The required authorization code must be installed in the corresponding product instance for these licenses as well. Cisco may pre-install export-controlled licenses when ordered with hardware purchase.

An example of an export-controlled license is the High Security (HSEC) license, which is available on certain Cisco Routers.

SLP Architecture

This section explains the various components that can be part of your SLP implementation.

Product Instance

A product instance is a single instance of a Cisco product, identified by a Unique Device Identifier (UDI).

A product instance records and reports license usage (RUM reports), and provides alerts and system messages about overdue reports, communication failures, etc. The RUM reports and usage data are also stored securely in the product instance.

A Resource Utilization Measurement report (RUM report) is a license usage report, which fulfils reporting requirements as specified by the policy. RUM reports are generated by the product instance and consumed by CSSM. The product instance records license usage information and all license usage changes in an open RUM report. At system-determined intervals, open RUM reports are closed and new RUM reports are opened to continue recording license usage. A closed RUM report is ready to be sent to CSSM.

A RUM acknowledgement (RUM ACK or ACK) is a response from CSSM and provides information about the status of a RUM report. Once the ACK for a report is available on the product instance, it indicates that the corresponding RUM report is no longer required and can be deleted.

CSSM displays license usage information as per the last received RUM report.

Cisco Smart Software Manager (CSSM)

CSSM is a portal that enables you to manage all your Cisco software licenses from a centralized location. CSSM helps you manage current requirements and review usage trends to plan for future license requirements.

You can access CSSM at https://software.cisco.com . Under the License tab, click the Smart Software Licensing link.

In CSSM you can:

- Create, manage, or view virtual accounts.
- Create and manage Product Instance Registration Tokens.
- Transfer licenses between virtual accounts or view licenses.
- Transfer, remove, or view product instances.
- Run reports against your virtual accounts.

- Modify your email notification settings.
- View overall account information.

Prior to using CSSM, please view a short video about how to use the portal found here:

https://www.cisco.com/c/en/us/buy/smart-accounts/software-manager.html

Click on the **View Video** button.

Cisco Smart Licensing Utility (CSLU)

CSLU is a Windows-based reporting utility that provides aggregate licensing work-flows. It helps you administer all your licenses and their associated product instances from your premises instead of having to connect to CSSM.

This utility performs the following key functions:

- Provides the options relating to how work-flows are triggered. The work-flows can be triggered by CSLU or by the product instance
- Collects usage reports from the product instance and upload these usage reports to the corresponding smart account or virtual account – online, or offline, using files. Similarly, the RUM report ACK is collected online, or offline, and provided back to the product instance.
- Sends authorization code requests to CSSM and receives authorization codes1 from CSSM.

CSLU can be part of your SLE topology in the following ways:

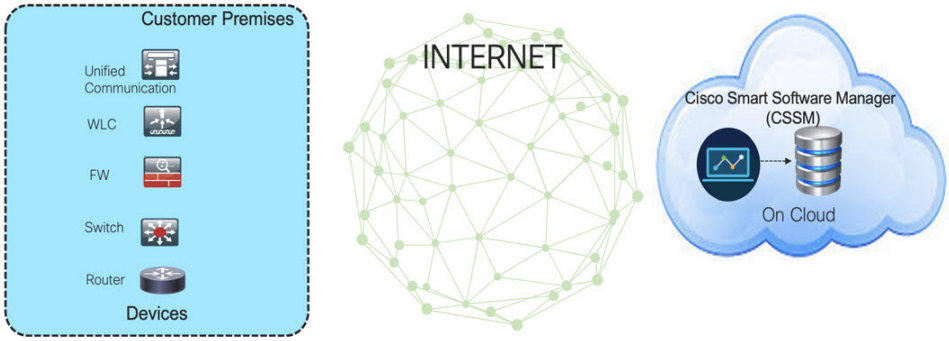
- Install the windows application, to use CSLU as a standalone tool and connect it to CSSM.
- Install the windows application, to use CSLU as a standalone tool and not connect it to CSSM. With this option, the required usage information is downloaded to a file and then uploaded to CSSM. This is suited to air-gapped networks.
- Embed it in a controller such as Cisco DNA Center.

Customer Topologies

IoT Routing platforms use two different topologies.

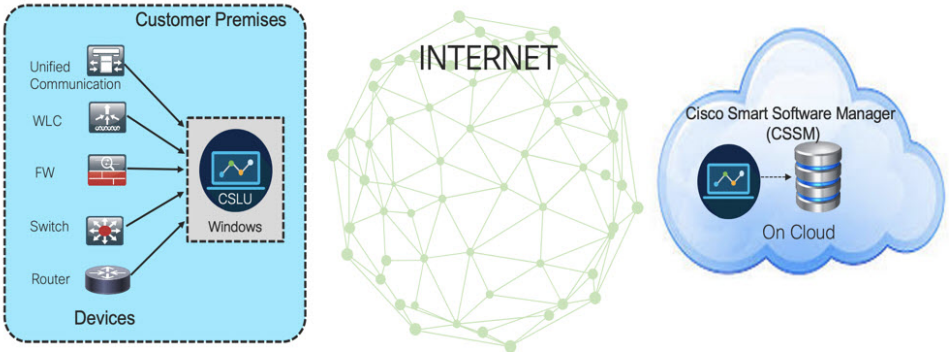
- Full Offline Access
- CSLU has No Access to CSSM

The following figure illustrates the Full Offline Access:



In this topology, devices do not have connectivity to CSSM (software.cisco.com). The user must copy and paste information between Cisco products and CSSM to manually check in and out licenses.

The following figure illustrates the CSLU having No Access to CSSM:



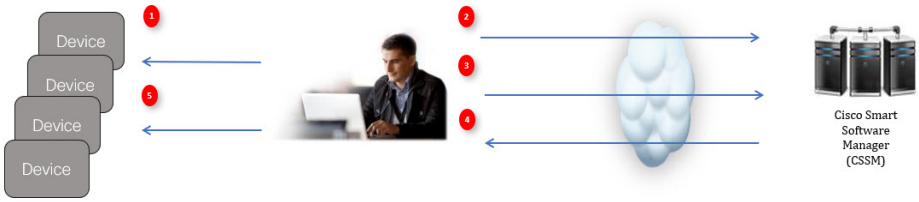
In this topology the devices are connected to the CSLU controller, but there is no connectivity between CSLU and CSSM (Cisco Smart Software Manager – software.cisco.com).

Cisco devices will send usage information to a locally installed CSLU. The user must copy and paste information between the CSLU and CSSM to manually check-in and check-out licenses.

License Installation Procedure - Full Offline Access Topology

This procedure requires a manual exchange of required information between the router and CSSM.

Refer to the following graphic for the flow of information:



1. Generate a License Usage Data file or AuthCode Request.
2. Export to CSSM.

3. Upload License Usage Data or AuthCode Request.
4. Export ACK/AuthRequest file to Router.
5. Upload ACK file or AuthRequestAuthCode

Procedure to Register Product Instance in CSSM

Procedure

Step 1 Generate a license usage file from the Router.

In exec mode, perform the following:

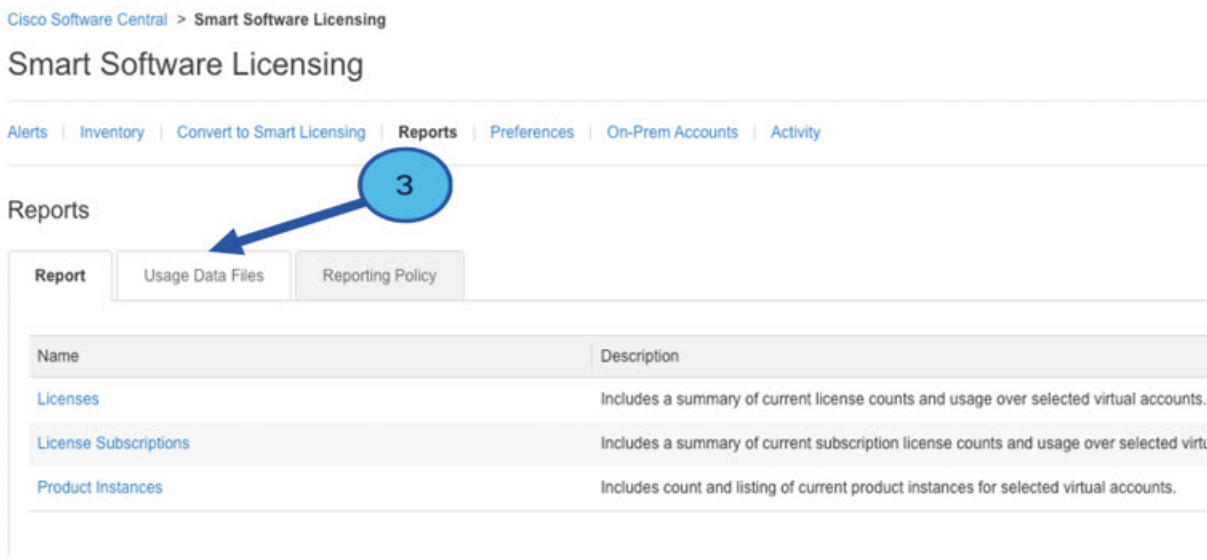
Example:

```
Router# license smart save usage all file flash:sle
```

Step 2 Export the license usage file (sle) to your host laptop/PC.

Step 3 Importing the license usage file to CSSM on Cloud. Click on the **Usage Data Files** tab.

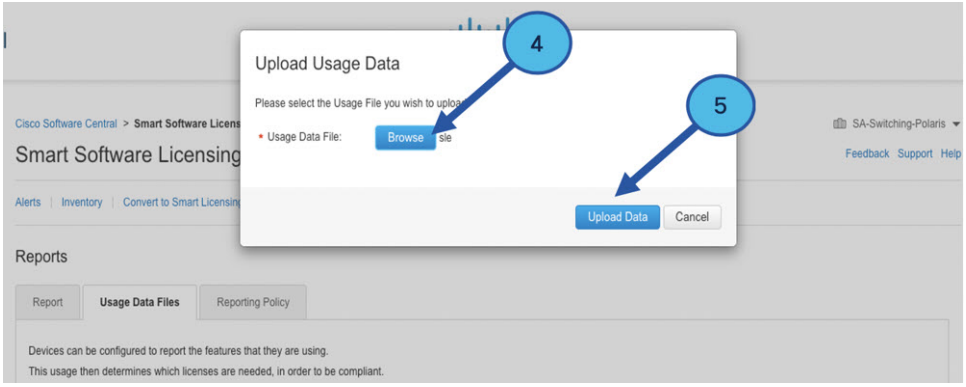
Figure 11: Usage Data File



Step 4 The **Upload Usage Data** window appears. Click **Browse**, and navigate to where the file is.

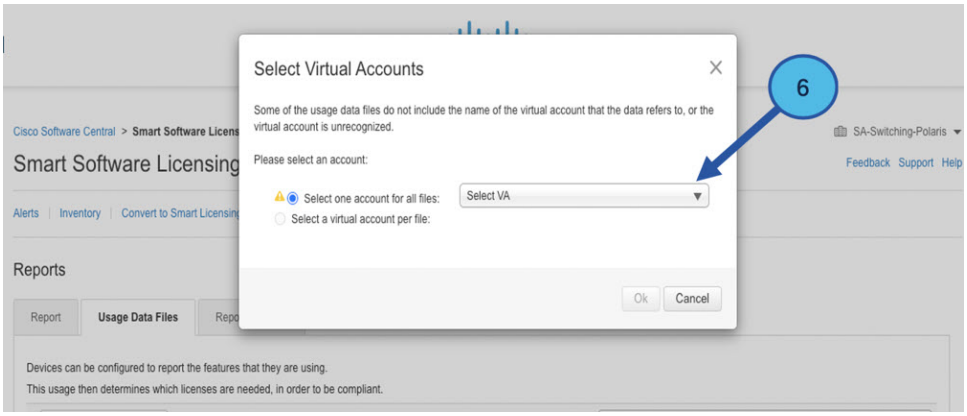
Step 5 Click on **Upload Data**.

Figure 12: Browse and Upload



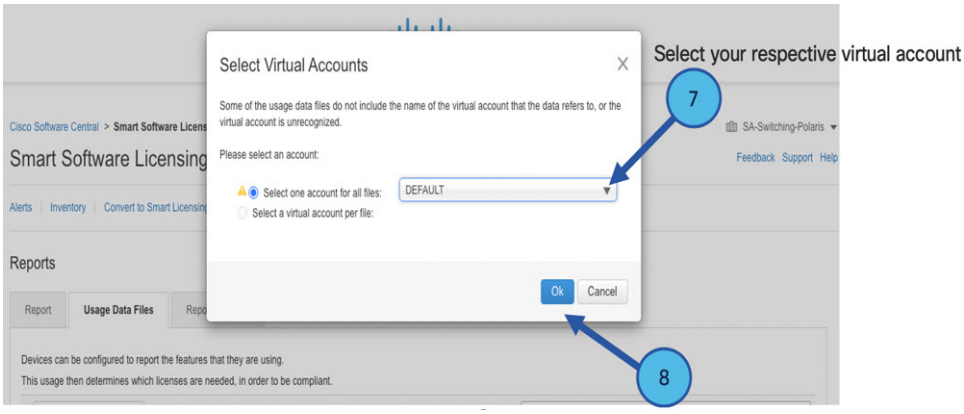
Step 6 Select the Virtual Account.

Figure 13: Select Account



Step 7 From the pull-down, select your respective virtual account.

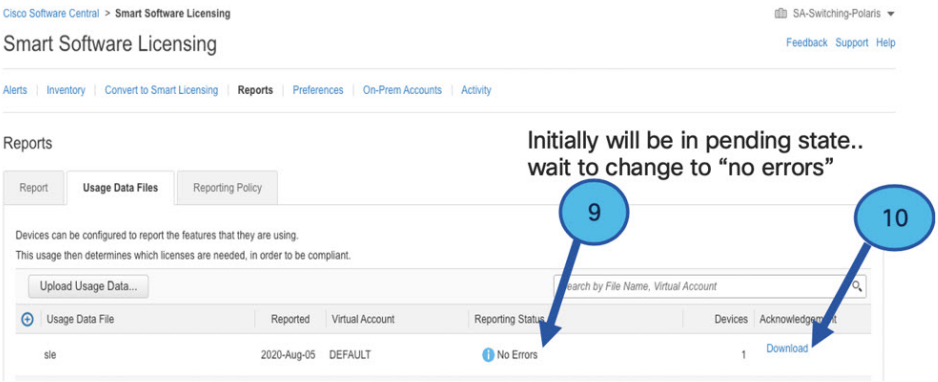
Figure 14: Select Your Account



Step 8 Click **Ok**.

Step 9 Observe the Smart Software Licensing window. Initially, the Reporting Status state will be **Pending**. Wait until the window reflects **No Errors** before continuing.

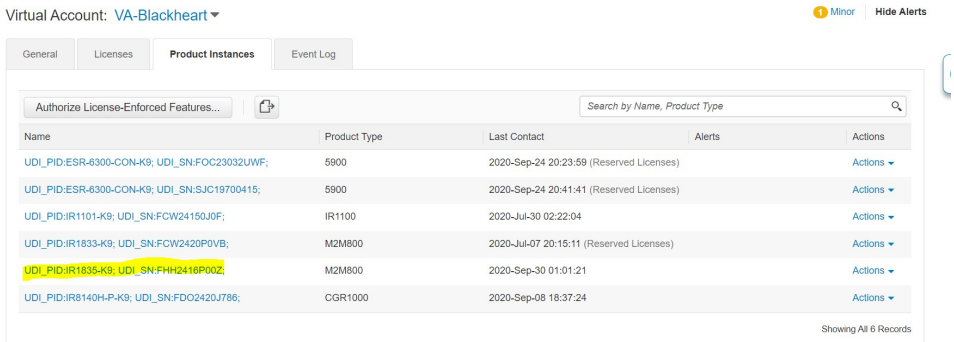
Figure 15: Reporting Status



Step 10 Click **Download** to download the ACK file.

Step 11 Check under the **Product Instances** tab to verify your device is listed.

Figure 16: Product Instances



Step 12 Import the ACK file from CSSM to your device using the command line interface.

Importing the ACK file from CSSM to your Device

Procedure

Step 1 Copy the ACK file from CSSM to your host laptop or usbflash device. In exec mode on the device:

Example:

```
Router#license smart import <flash: | usbflash0:> ACK_sle
Import Data Successful
Router#
*Sep  1 21:12:58.576: %SIP-1-LICENSING: SIP service is Up. License report acknowledged.
```

```
*Sep  1 21:12:58.616: %SMART_LIC-6-POLICY_INSTALL_SUCCESS: A new licensing policy was
successfully installed
```

Step 2 Verify Product Instance has imported the data

Example:

```
Router# show license usage
License Authorization:
  Status: Not Applicable
network-advantage_250M (IR1800_P_250M_A):
  Description: network-advantage_250M
  Count: 1
  Version: 1.0
  Status: IN USE
  Export status: NOT RESTRICTED
  Feature Name: network-advantage_250M
  Feature Description: network-advantage_250M
  Enforcement type: NOT ENFORCED
```

Step 3 Verify the license is in use.

Example:

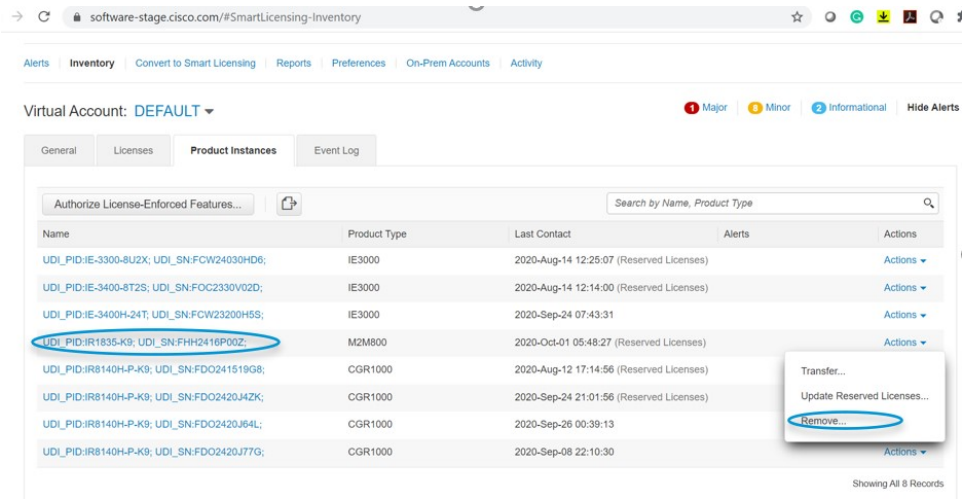
```
Router# show license summary
License Usage:
  License                                     Entitlement tag              Count   Status

  -----------------------------------------------------------------------------------------
  network-advantage_250M  (IR1800_P_250M_A)         1      IN USE

Router#
Router#show license all | beg Usage Reporting:
Usage Reporting:
  Last ACK received: Sep 01 21:12:58 2020 UTC
  Next ACK deadline: <none>
  Reporting Interval: 0 (no reporting)
  Next ACK push check: <none>
  Next report push: <none>
  Last report push: <none>
  Last report file write: <none>
Trust Code Installed: Sep 01 00:28:48 2020 UTC
```

Removing the Device from CSSM

Procedure

Step 1 Navigate back to the product instances tab. Locate your device.

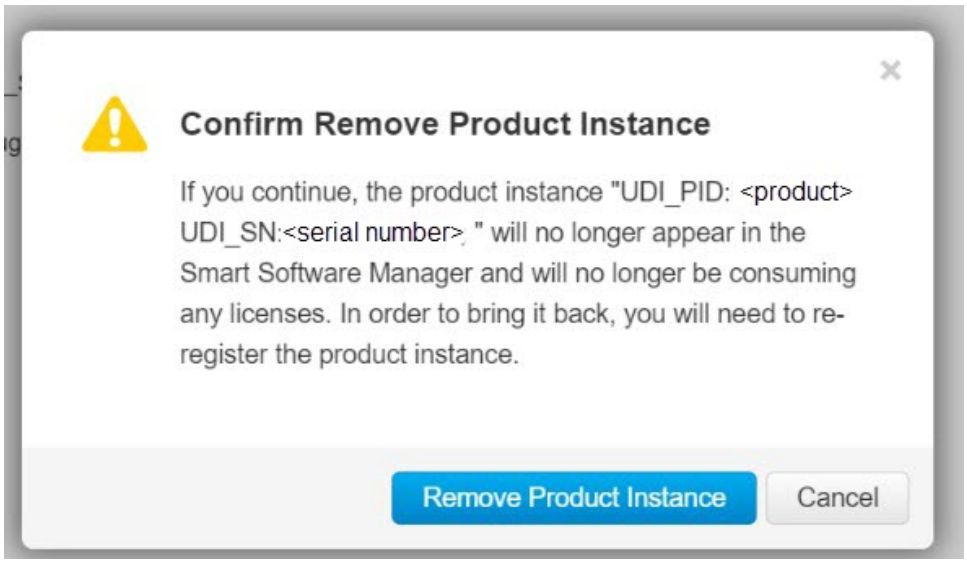
Figure 17: Product Instances



Step 2 Click on **Actions** beside your device, and from those options click **Remove**.

The Confirm Remove Product Instance window appears.

Figure 18: Confirm Remove Product Instance

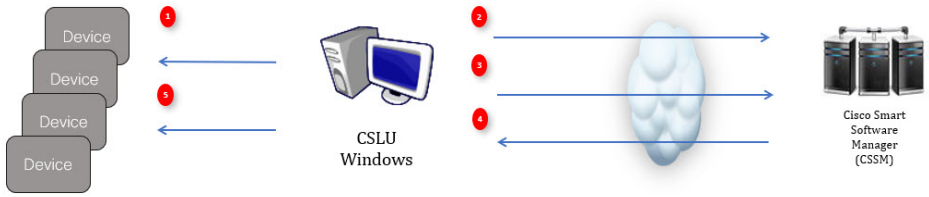


Step 3 Click **Remove Product Instance**.

License Installation Procedure - CSLU has No Access to CSSM

This procedure performs an online exchange of required information between the Router and CSLU.

Refer to the following graphic for the flow of information:



Procedure

Step 1 In CSLU, identify the devices that require an AuthCode, and initiate the request. An AuthCode file is created.

Step 2 Export the AuthCode file to CSSM.

Step 3 Upload the AuthCode to CSSM SA/VA account.

Step 4 Export the AuthRequestAuthcode file to CSLU.

Step 5 Upload ACK file or AuthRequestAuthCode

Procedure when devices are connected to the CSLU

First, perform these steps on the router using the CLI to get a license UDI:

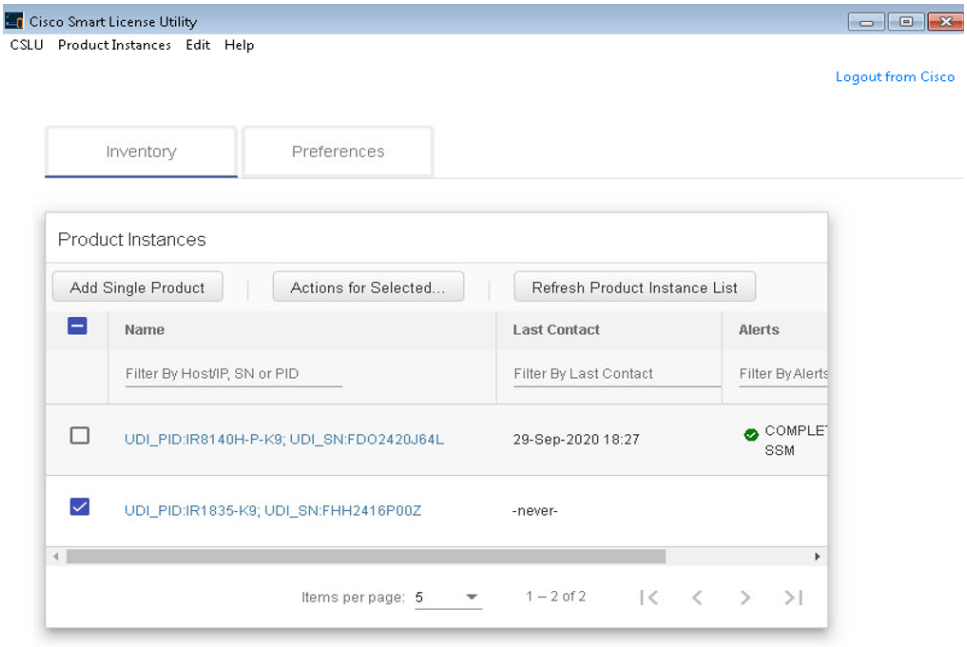
```
Router#show license summary
License Reservation is ENABLED
License Usage:
License Entitlement tag Count Status
-----------------------------------------------------------------------------
network-essentials_250M (IR1800_P_250M_E) 1 IN USE

Router#configure terminal
Enter configuration commands, one per line. End with CNTL/Z.
Router(config)#platform hardware throughput level 2G
% 2G throughput level requires hseck9 license!
Router(config)#end

Router#sh license udi
UDI: PID:IR1835-K9,SN:FHH2416P00Z
```

Procedure

Step 1 Open the Cisco Smart License Utility (CSLU).

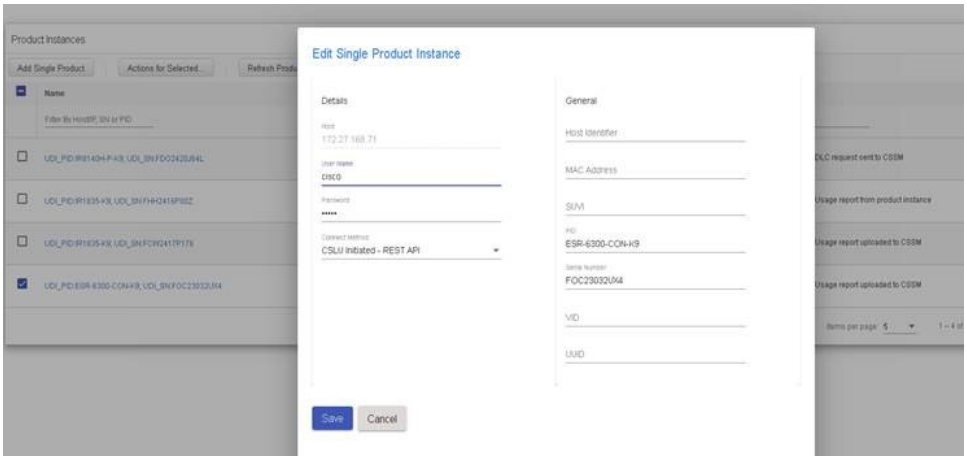
Step 2 Navigate to the **Product Instances** tab, then click on the UDI.

Figure 19: Select UDI



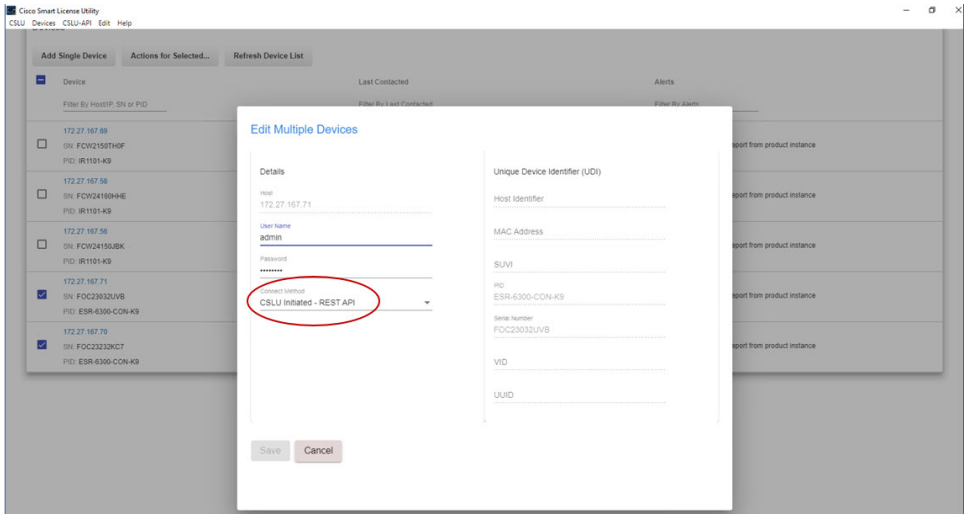
Step 3 The **Edit Single Product Instance** window appears.

Figure 20: Edit Single Product Instance



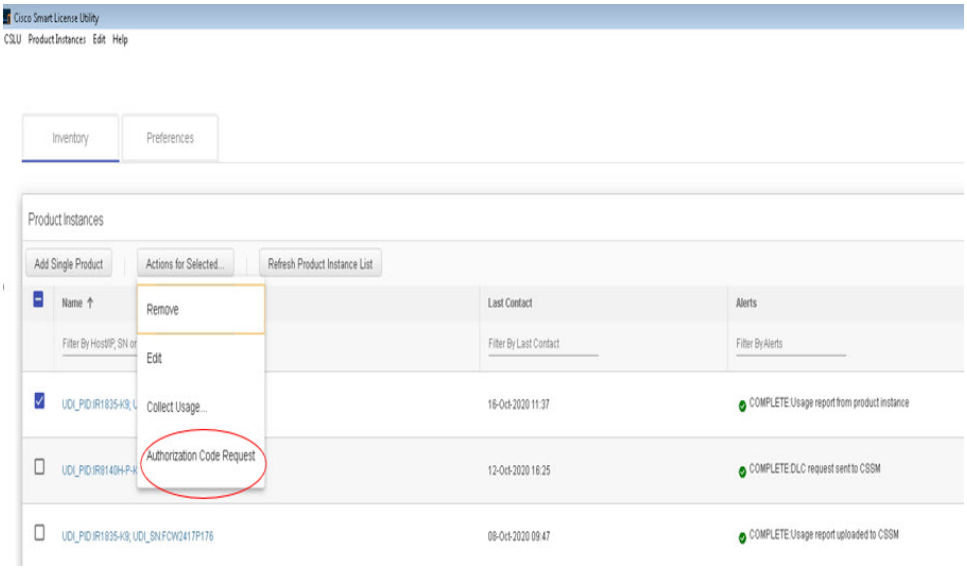
Step 4 The **Edit Multiple Devices** window appears. Supply your account password and click **Save**.

Figure 21: Edit Multiple Devices



Step 5 In the **Product Instances** window, click on the **Actions for Selected Devices** Tab.

Figure 22: Actions for Selected Devices



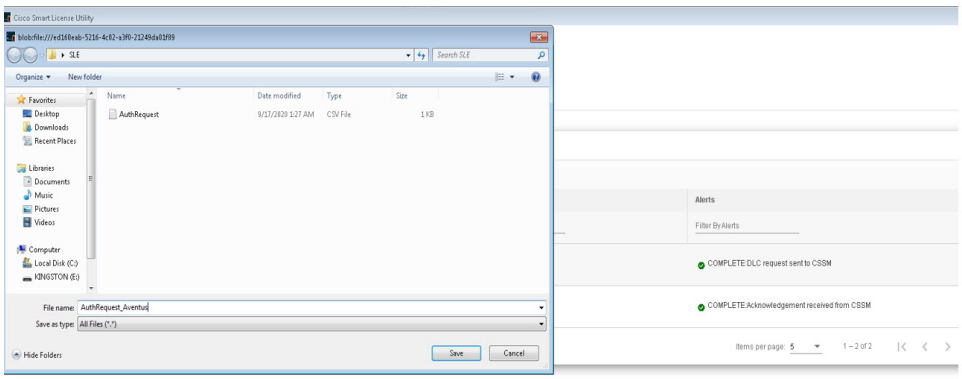
Step 6 Select **Authorization Code Request**.

Step 7 The **Authorization Request Information** window appears. Read the contents and then click **Accept**.

Figure 23: Authorization Request Information

Step 8 The CSLU downloads a Authorization Request file to your laptop. Click **Save**.

Figure 24: Authorization Request File



Exporting the AuthRequest File to CSSM

The next step is to take the Authorization Request file you just saved, and export it into Cisco Smart Software Manager (CSSM).

Launch CSSM.

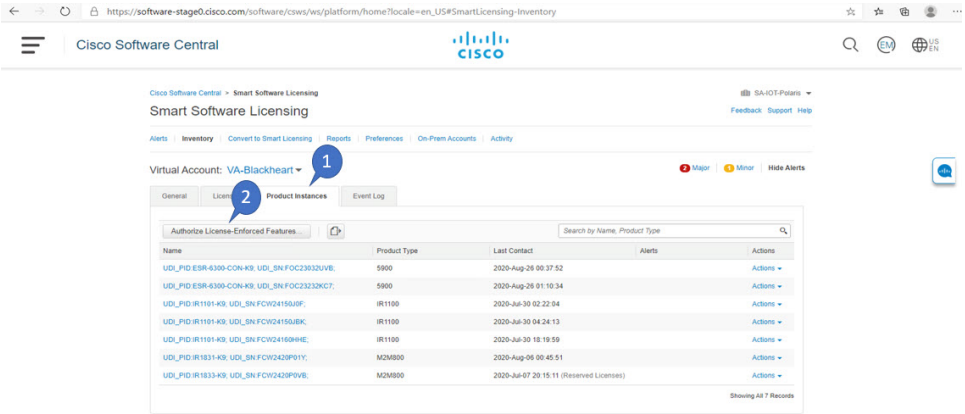
Click on the **Inventory** Tab, select your Virtual Account.

Procedure

Step 1 Click on the **Product Instances** Tab.

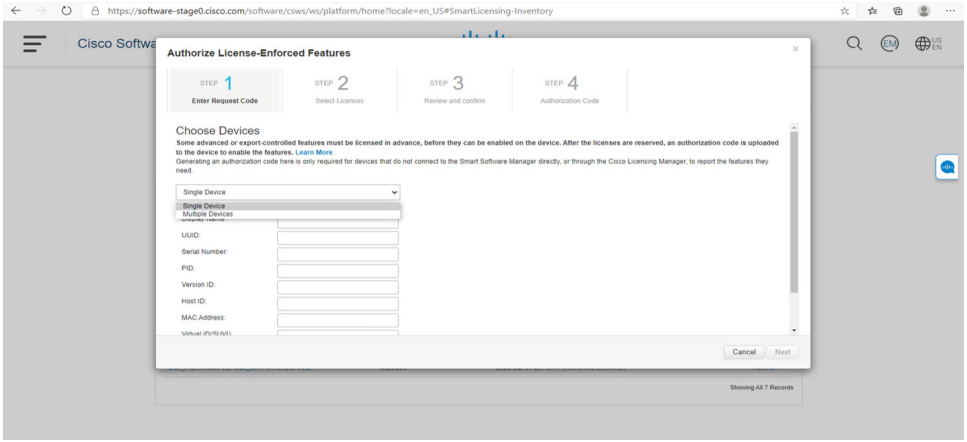
Step 2 Click on **Authorize License-Enforced Features**.

Figure 25: Authorize License-Enforced Features



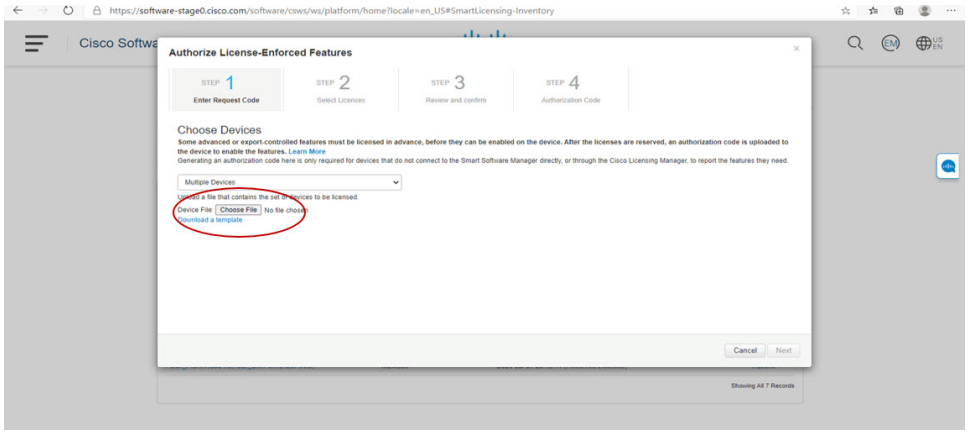
The **Authorize License-Enforced Features** window appears**.**

Figure 26: Authorize License-Enforced Features



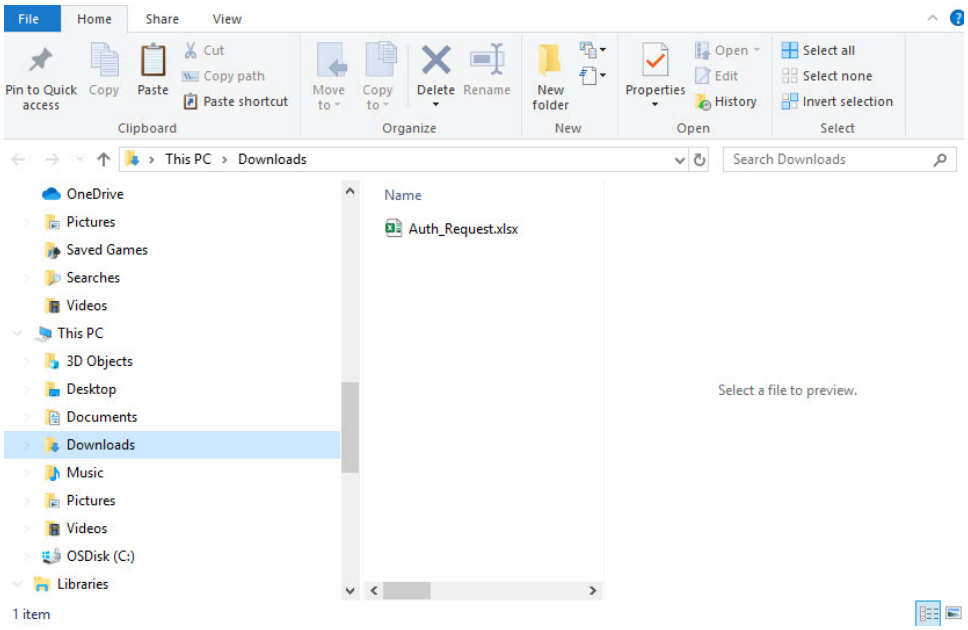
Step 3 Choose **Multiple** or **Single** devices from the pull-down.

Step 4 The window changes to an option to select a device file. Click on **Choose File**.



Step 5 A popup window opens to navigate to where you saved your Authorization Request file on your laptop.

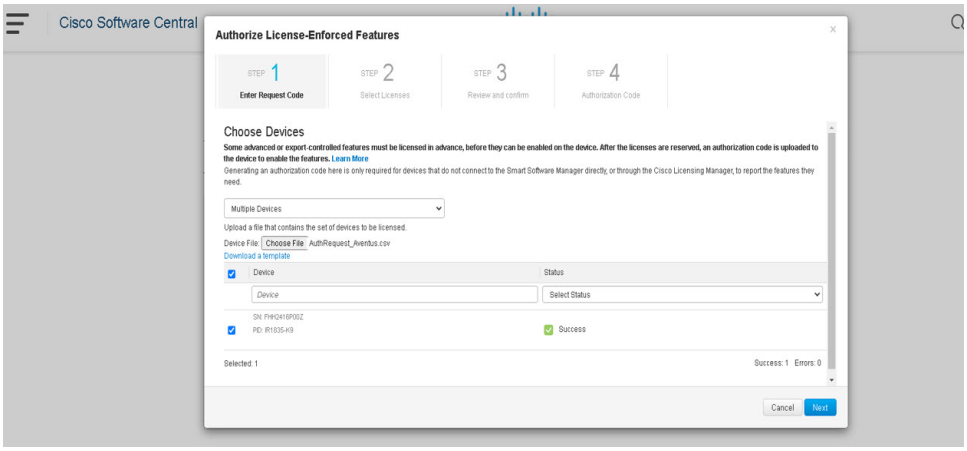
Figure 27: Open File Navigation Window



Step 6 Select your file, and then click **Open**.

Step 7 The authorization file loads, and the window changes to present your devices.

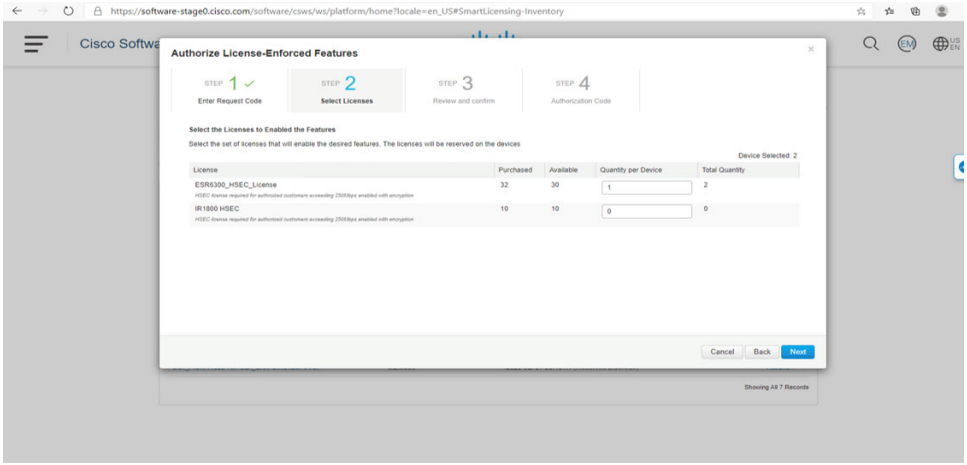
Figure 28: Present Devices



Step 8 When successful, click **Next**.

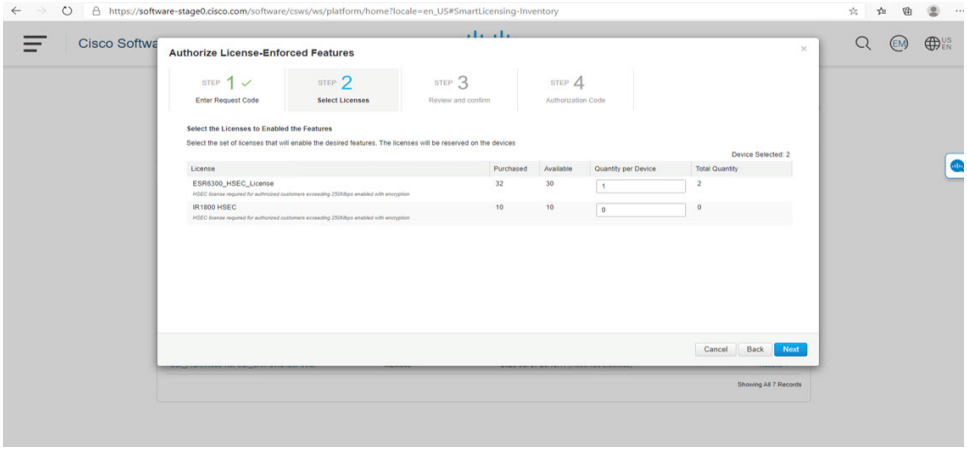
Step 9 The **Select Licenses** Tab opens.

Figure 29: Select Licenses



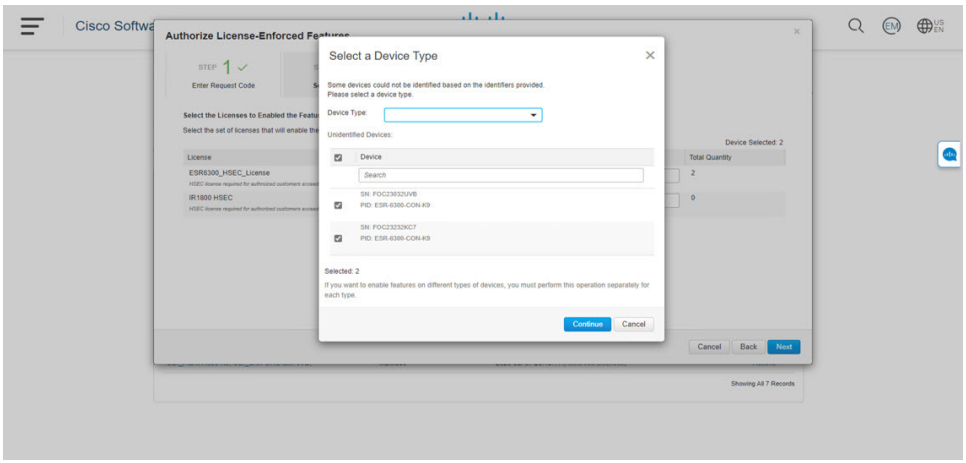
Step 10 Under **Quantity per Device**, enter the number you wish.

Figure 30: Enter Number



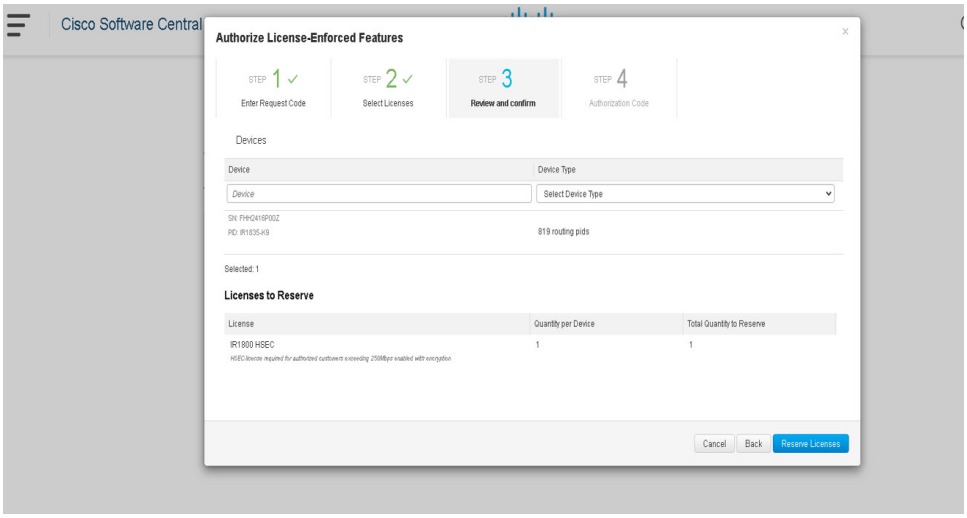
Step 11 If CSSM cannot identify your device from the identifying information, you can select it manually.

Figure 31: Select a Device Type



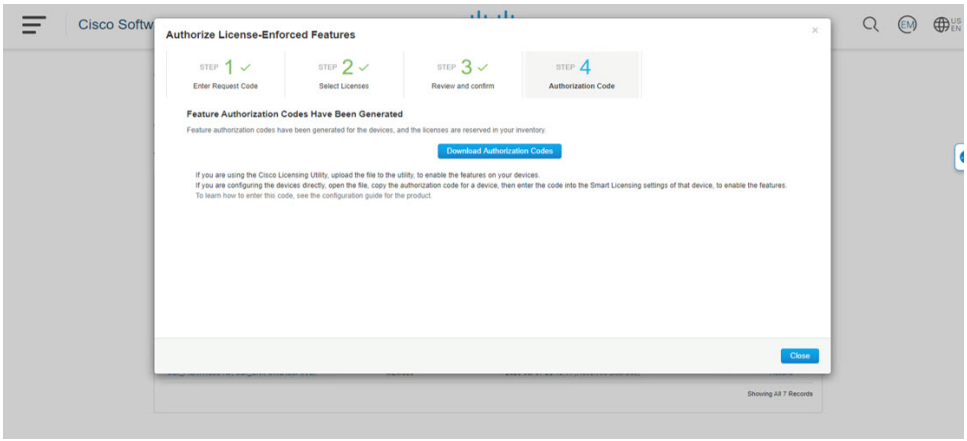
Step 12 Click **Continue**, and the window changes to **Review and Confirm**.

Figure 32: Review and Confirm



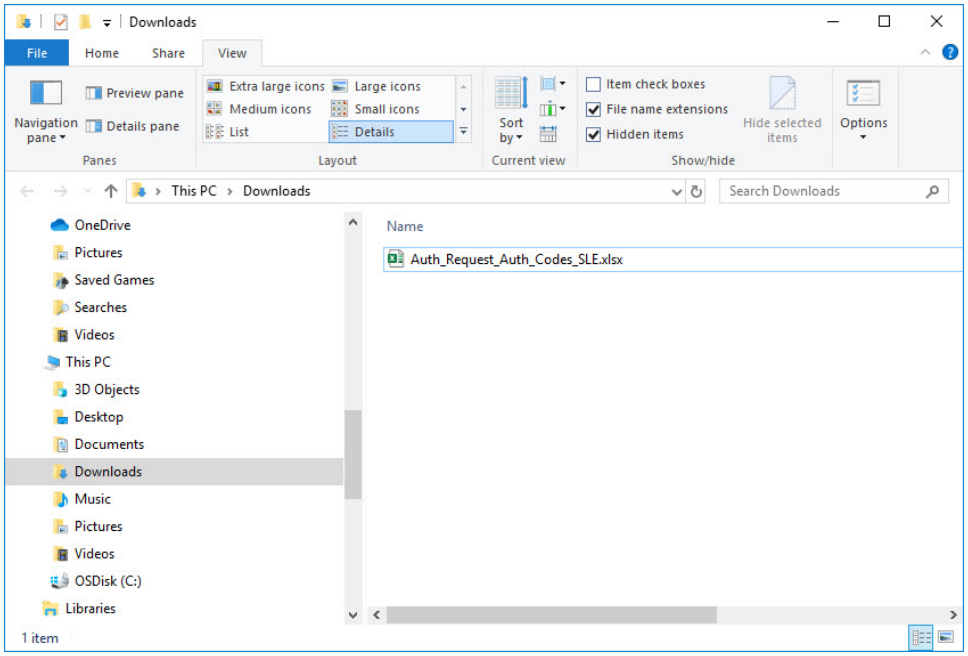
Step 13 Click on **Reserve Licenses**, and CSSM generates feature authorization codes.

Figure 33: Feature Authorization Codes



Step 14 Click **Download Authorization Codes**, and a window opens to navigate to where you wish to save the codes.

Figure 34: Save Authorization Code



Step 15 Click **Ok**.

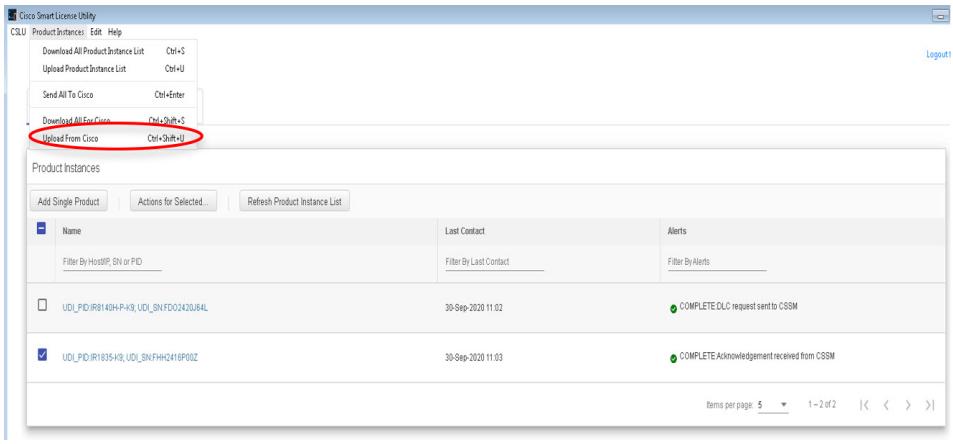
Uploading the Authorization Request Code file into CSLU

Procedure

Step 1 Open the Cisco Smart License Utility (CSLU).

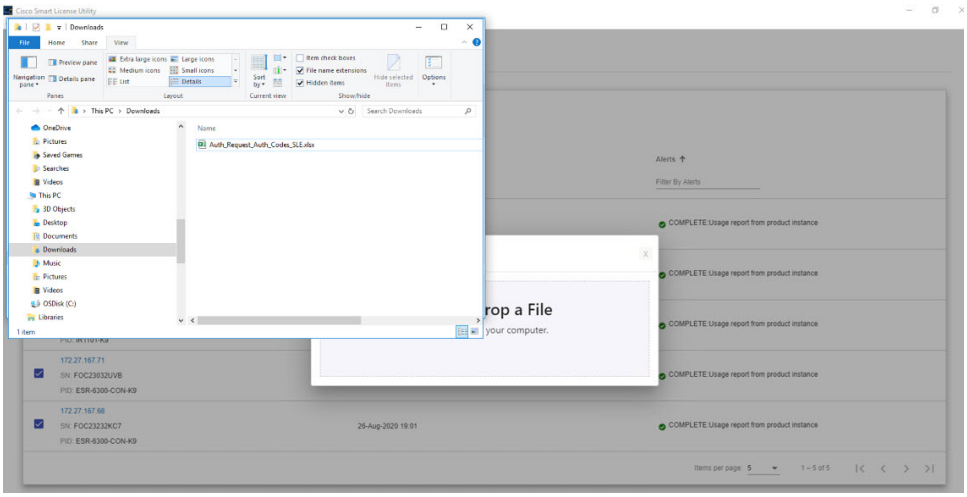
Step 2 Navigate to **Product Instances**, and then select **Upload From Cisco**.

Figure 35: Upload From Cisco



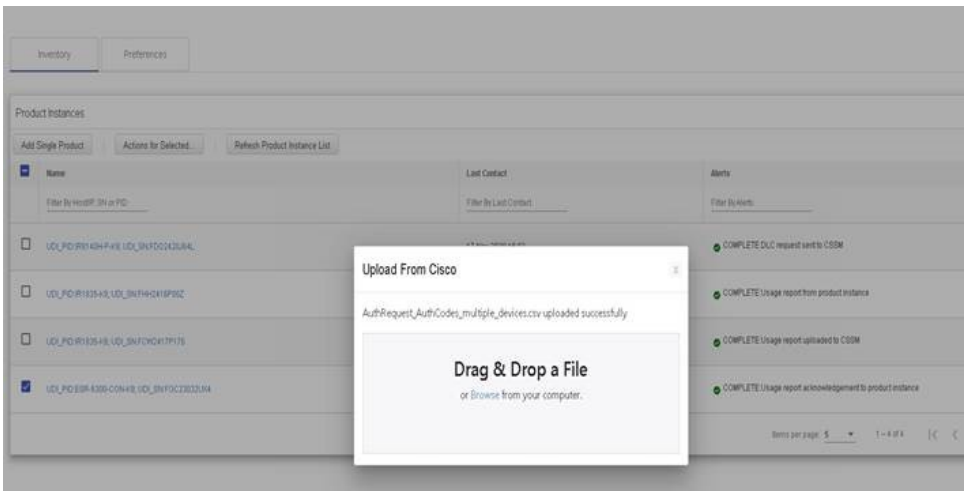
Step 3 There are two options to load your file. **Drag and Drop**, or **Browse** to where you saved your file. This example shows Browse.

Figure 36: Browse to File



Step 4 Select your authorization code file, and then click **Open**. The system uploads the authorization code file, then a successful upload message appears.

Figure 37: Successful Upload



License Installation Process in the Router

Perform the following from the command line interface.

```
Router#show license summary
License Reservation is ENABLED
License Usage:
  License                  Entitlement tag              Count Status
```

```
  ------------------------------------------------------------------------------
  network-essentials_250M (IR1800_P_250M_E)                 1 IN USE
  hseck9                  (IR1800_HSEC)                     1 IN USE
Router#show license usage
License Authorization:
  Status: Not Applicable
network-essentials_250M (IR1800_P_250M_E):
  Description: network-essentials_250M
  Count: 1
  Version: 1.0
  Status: IN USE
  Export status: NOT RESTRICTED
  Feature Name: network-essentials_250M
  Feature Description: network-essentials_250M
  Enforcement type: NOT ENFORCED

hseck9 (IR1800_HSEC):
  Description: hseck9
  Count: 1
  Version: 1.0
  Status: IN USE
  Export status: RESTRICTED - ALLOWED
  Feature Name: hseck9
  Feature Description: hseck9
  Enforcement type: EXPORT RESTRICTED
Router(config)#platform hardware throughput level 2G
% Please write mem and reload
% The config will take effect on next reboot
Router(config)#end
Router#
*Sep 30 18:05:55.654: %SYS-5-CONFIG_I: Configured from console by cisco on console
Router#show license summary
License Reservation is ENABLED
License Usage:
  License                 Entitlement tag               Count Status
  ------------------------------------------------------------------------------
  network-essentials_250M (IR1800_P_250M_E)                 1 IN USE
  hseck9                  (IR1800_HSEC)                     1 IN USE
  network-essentials_2G   (IR1800_P_2G_E)                   1 IN USE
```

CHAPTER 16

Configuring Ethernet Switch Ports

This chapter contains the following sections:

- Configuring VLANs, on page 141
- VLAN Trunking Protocol (VTP), on page 142
- Configuring 802.1x Authentication, on page 142
- Configuring Spanning Tree Protocol, on page 143
- Configuring MAC Address Table Manipulation, on page 145
- Configuring Switch Port Analyzer, on page 146
- Configuring IGMP Snooping, on page 147

Configuring VLANs

A VLAN is a switched network that is logically segmented by function, project team, or application, without regard to the physical locations of the users. VLANs have the same attributes as physical LANs, but you can group end stations even if they are not physically located on the same LAN segment. Any switch port can belong to a VLAN, and unicast, broadcast, and multicast packets are forwarded and flooded only to end stations in the VLAN. Each VLAN is considered a logical network, and packets destined for stations that do not belong to the VLAN must be forwarded through a router.

Note There no support for Jumbo frames on L2 interfaces.

The following is an example of a vlan configuration:

```
IR1800#show vlan
VLAN Name                             Status    Ports
---- -------------------------------- --------- -------------------------------
1    default                          active    Ge0/1/0, Ge0/1/1, Ge0/1/2, Ge0/1/3
1002 fddi-default                     act/unsup
1003 token-ring-default               act/unsup
1004 fddinet-default                  act/unsup
1005 trnet-default                    act/unsup

VLAN Type  SAID       MTU   Parent RingNo BridgeNo Stp  BrdgMode Trans1 Trans2
---- ----- ---------- ----- ------ ------ -------- ---- -------- ------ ------
1    enet  100001     1500  -      -      -        -    -        0      0
1002 fddi  101002     1500  -      -      -        -    -        0      0
1003 tr    101003     1500  -      -      -        -    -        0      0
1004 fdnet 101004     1500  -      -      -        ieee -        0      0
```

```
1005 trnet 101005     1500  -      -      -        ibm  -        0      0

Primary Secondary Type              Ports
------- --------- ----------------- ------------------------------------------

IR1800#
```

You can assign a given port to a vlan by following these steps:

```
interface GigabitEthernet0/1/0
switchport access vlan 4

interface vlan 4
ip v4 address ...
ipv6 address autoconf
```

VLAN Trunking Protocol (VTP)

VTP is a Layer 2 messaging protocol that maintains VLAN configuration consistency by managing the addition, deletion, and renaming of VLANs on a network-wide basis. VTP minimizes misconfigurations and configuration inconsistencies that can cause several problems, such as duplicate VLAN names, incorrect VLAN-type specifications, and security violations.

Before you create VLANs, you must decide whether to use VTP in your network. Using VTP, you can make configuration changes centrally on one or more switches and have those changes automatically communicated to all the other switches in the network. Without VTP, you cannot send information about VLANs to other switches.VTP is designed to work in an environment where updates are made on a single switch and are sent through VTP to other switches in the domain. It does not work well in a situation where multiple updates to the VLAN database occur simultaneously on switches in the same domain, which would result in an inconsistency in the VLAN database.

Further information about configuring VTP can be found here:http://www.cisco.com/c/en/us/td/docs/routers/access/interfaces/software/feature/guide/geshwic_cfg.html#wp1046901

Configuring 802.1x Authentication

IEEE 802.1x port-based authentication defines a client-server-based access control and authentication protocol to prevent unauthorized clients from connecting to a LAN through publicly accessible ports.The authentication server authenticates each client connected to a switch port before allowing access to any switch or LAN services. Until the client is authenticated, IEEE 802.1x access control allows only Extensible Authentication Protocol over LAN (EAPOL), Cisco Discovery Protocol (CDP), and Spanning Tree Protocol (STP) traffic through the port to which the client is connected. After authentication, normal traffic passes through the port.

With IEEE 802.1x authentication, the devices in the network have specific roles:

- Supplicant—Device (workstation) that requests access to the LAN and switch services and responds to requests from the router. The workstation must be running IEEE 802.1x-compliant client software such as that offered in the Microsoft Windows XP operating system. (The supplicant is sometimes called the client.)
- Authentication server—Device that performs the actual authentication of the supplicant. The authentication server validates the identity of the supplicant and notifies the router whether or not the supplicant is authorized to access the LAN and switch services. The Network Access Device transparently passes the authentication messages between the supplicant and the authentication server, and the authentication

process is carried out between the supplicant and the authentication server. The particular EAP method used will be decided between the supplicant and the authentication server (RADIUS server). The RADIUS security system with EAP extensions is available in Cisco Secure Access Control Server Version 3.0 or later. RADIUS operates in a client and server model in which secure authentication information is exchanged between the RADIUS server and one or more RADIUS clients.

- Authenticator—Router that controls the physical access to the network based on the authentication status of the supplicant. The router acts as an intermediary between the supplicant and the authentication server, requesting identity information from the supplicant, verifying that information with the authentication server, and relaying a response to the supplicant. The router includes the RADIUS client, which is responsible for encapsulating and decapsulating the EAP frames and interacting with the authentication server.

For detailed information on how to configure 802.1x port-based authentication, see the following link:

http://www.cisco.com/c/en/us/td/docs/ios-xml/ios/sec_usr_8021x/configuration/15-mt/sec-user-8021x-15-mt-book/config-ieee-802x-pba.html

Example: Enabling IEEE 802.1x and AAA on a Switch Port

This example shows how to configure an IR1800 router as 802.1x authenticator:

```
Router> enable
Router# configure terminal
Router(config)# dot1x system-auth-control
Router(config)# aaa new-model
Router(config)# aaa authentication dot1x default group radius
Router(config)# interface GigabitEthernet 0/1/0
Router(config-if)# switchport mode access
Router(config-if)# access-session port-control auto
Router(config-if)# dot1x pae authenticator
Router(config-if)# access-session closed
Router(config-if)# access-session host-mode single-host
Router(config-if)# end
```

Configuring Spanning Tree Protocol

Spanning Tree Protocol (STP) is a Layer 2 link management protocol that provides path redundancy while preventing loops in the network. For a Layer 2 Ethernet network to function properly, only one active path can exist between any two stations. Multiple active paths among end stations cause loops in the network. If a loop exists in the network, end stations might receive duplicate messages. Switches might also learn end-station MAC addresses on multiple Layer 2 interfaces. These conditions result in an unstable network. Spanning-tree operation is transparent to end stations, which cannot detect whether they are connected to a single LAN segment or a switched LAN of multiple segments.

The STP uses a spanning-tree algorithm to select one switch of a redundantly connected network as the root of the spanning tree. The algorithm calculates the best loop-free path through a switched Layer 2 network by assigning a role to each port based on the role of the port in the active topology:

- Root—A forwarding port elected for the spanning-tree topology
- Designated—A forwarding port elected for every switched LAN segment
- Alternate—A blocked port providing an alternate path to the root bridge in the spanning tree
- Backup—A blocked port in a loopback configuration

The switch that has all of its ports as the designated role or as the backup role is the root switch. The switch that has at least one of its ports in the designated role is called the designated switch.Spanning tree forces redundant data paths into a standby (blocked) state. If a network segment in the spanning tree fails and a redundant path exists, the spanning-tree algorithm recalculates the spanning-tree topology and activates the standby path. Switches send and receive spanning-tree frames, called bridge protocol data units (BPDUs), at regular intervals. The switches do not forward these frames but use them to construct a loop-free path. BPDUs contain information about the sending switch and its ports, including switch and MAC addresses, switch priority, port priority, and path cost. Spanning tree uses this information to elect the root switch and root port for the switched network and the root port and designated port for each switched segment.

When two ports on a switch are part of a loop, the spanning-tree port priority and path cost settings control which port is put in the forwarding state and which is put in the blocking state. The spanning-tree port priority value represents the location of a port in the network topology and how well it is located to pass traffic. The path cost value represents the media speed.

For detailed configuration information on STP see the following link:

http://www.cisco.com/c/en/us/td/docs/routers/access/interfaces/NIM/software/configuration/guide/4_8PortGENIM.html#pgfId-1079138

Important If the router is factory-defaulted, write erased, or config-reset, the vlan database gets deleted. Even though the configuration takes effect, interfaces need to be removed and re-applied.

Example: Spanning Tree Protocol Configuration

The following example shows configuring spanning-tree port priority of a Gigabit Ethernet interface. If a loop occurs, spanning tree uses the port priority when selecting an interface to put in the forwarding state.

```
Router# configure terminal
Router(config)# interface GigabitEthernet 0/1/0
Router(config-if)# spanning-tree vlan 1 port-priority 64
Router(config-if)# end
```

The following example shows how to change the spanning-tree port cost of a Gigabit Ethernet interface. If a loop occurs, spanning tree uses cost when selecting an interface to put in the forwarding state.

```
Router#configure terminal
Router(config)# interface GigabitEthernet 0/1/0
Router(config-if)# spanning-tree cost 18
Router(config-if)# end
```

The following example shows configuring the bridge priority of VLAN 10 to 33792:

```
Router# configure terminal
Router(config)# spanning-tree vlan 10 priority 33792
Router(config)# end
```

The following example shows configuring the hello time for VLAN 10 being configured to 7 seconds. The hello time is the interval between the generation of configuration messages by the root switch.

```
Router# configure terminal
Router(config)# spanning-tree vlan 10 hello-time 7
Router(config)# end
```

The following example shows configuring forward delay time. The forward delay is the number of seconds an interface waits before changing from its spanning-tree learning and listening states to the forwarding state.

```
Router# configure terminal
Router(config)# spanning-tree vlan 10 forward-time 21
Router(config)# end
```

The following example shows configuring maximum age interval for the spanning tree. The maximum-aging time is the number of seconds a switch waits without receiving spanning-tree configuration messages before attempting a reconfiguration.

```
Router# configure terminal
Router(config)# spanning-tree vlan 20 max-age 36
Router(config)# end
```

The following example shows the switch being configured as the root bridge for VLAN 10, with a network diameter of 4.

```
Router# configure terminal
Router(config)# spanning-tree vlan 10 root primary diameter 4
Router(config)# exit
```

Configuring MAC Address Table Manipulation

The MAC address table contains address information that the switch uses to forward traffic between ports. All MAC addresses in the address table are associated with one or more ports. The address table includes these types of addresses:

- Dynamic address: a source MAC address that the switch learns and then drops when it is not in use. You can use the aging time setting to define how long the switch retains unseen addresses in the table.
- Static address: a manually entered unicast address that does not age and that is not lost when the switch resets.

The address table lists the destination MAC address, the associated VLAN ID, and port associated with the address and the type (static or dynamic).

Port security is supported, as is sticky MAC addresses.

See the “Example: MAC Address Table Manipulation” for sample configurations for enabling secure MAC address, creating a statc entry, set the maximum number of secure MAC addresses and set the aging time.

For detailed configuration information on MAC address table manipulation see the following link:

http://www.cisco.com/c/en/us/td/docs/routers/access/interfaces/software/feature/guide/geshwic_cfg.html#wp1048223

Example: MAC Address Table Manipulation

The following example shows creating a static entry in the MAC address table.

```
Router# configure terminal
Router(config)# mac address-table static 0002.0003.0004 interface GigabitEthernet 0/1/0
vlan 3
Router(config)# end
```

The following example shows setting the aging timer.

```
Router# configure terminal
Router(config)# mac address-table aging-time 300
Router(config)# end
```

Configuring Switch Port Analyzer

The Cisco IR1800 supports local SPAN only, and up to one SPAN session. You can analyze network traffic passing through ports by using SPAN to send a copy of the traffic to another port on the switch or on another switch that has been connected to a network analyzer or other monitoring or security device. SPAN copies (or mirrors) traffic received or sent (or both) on source ports to a destination port for analysis. SPAN does not affect the switching of network traffic on the source ports. You must dedicate the destination port for SPAN use. Except for traffic that is required for the SPAN or RSPAN session, destination ports do not receive or forward traffic.

Only traffic that enters or leaves source ports or traffic that enters or leaves source can be monitored by using SPAN; traffic routed to a source cannot be monitored. For example, if incoming traffic is being monitored, traffic that gets routed from another source cannot be monitored; however, traffic that is received on the source and routed to another can be monitored.

For detailed information on how to configure a switched port analyzer (SPAN) session, see the following web link:

http://www.cisco.com/c/en/us/td/docs/switches/lan/catalyst3750/software/release/15-0_2_se/configuration/guide/scg3750/swspan.html

Example: SPAN Configuration

The following example shows how to configure a SPAN session to monitor bidirectional traffic from a Gigabit Ethernet source interface:

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

The following example shows how to configure a gigabit ethernet interface as the destination for a SPAN session:

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

The following example shows how to remove gigabit ethernet as a SPAN source for SPAN session 1:

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

Show Monitor Example

```
Router(config)#monitor session 1 source interface gi0/1/0
Router(config)#monitor session 1 destination interface gi0/1/1
Router#sh monitor session 1
Session 1
---------
Type : Local Session
Source Ports :
```

```
Both : Gi0/1/0
Destination Ports : Gi0/1/1
```

Example of ERSPAN

```
Router#show monitor session 1
Session 1
---------
Type                      : ERSPAN Source Session
Status                    : Admin Disabled
Source Ports              :
    RX Only               : Gi0/0/0
Destination IP Address    : 172.5.5.200
MTU                       : 1464
Destination ERSPAN ID     : 100
Origin IP Address         : 172.5.6.2
IPv6 DSCP                 : 0
IPV6 TTL                  : 0
```

Configuring IGMP Snooping

IGMP snooping constrains the flooding of multicast traffic by dynamically configuring Layer 2 interfaces so that multicast traffic is forwarded to only those interfaces associated with IP multicast devices. As the name implies, IGMP snooping requires the LAN switch to snoop on the IGMP transmissions between the host and the router and to keep track of multicast groups and member ports. When the switch receives an IGMP report from a host for a particular multicast group, the switch adds the host port number to the forwarding table entry; when it receives an IGMP Leave Group message from a host, it removes the host port from the table entry. It also periodically deletes entries if it does not receive IGMP membership reports from the multicast clients.

The multicast router sends out periodic general queries to all VLANs. All hosts interested in this multicast traffic send join requests and are added to the forwarding table entry.

Use the **ip igmp snooping enable** command to configure IGMP Snooping on the IR1800.

By default, IGMP snooping is globally enabled in the IR1800.

MLD snooping is also supported on the IR1800, and further information can be found in this documentation set: https://www.cisco.com/c/en/us/td/docs/switches/lan/catalyst3850/software/release/16-1/configuration_guide/b_161_consolidated_3850_cg/b_161_consolidated_3850_cg_chapter_01100.html

CHAPTER 17

Power Over Ethernet (PoE)

This section contains the following:

- Power over Ethernet Overview, on page 149
- Device Detection and Power Allocation, on page 149
- Command Line Interface, on page 149

Power over Ethernet Overview

Power over Ethernet (PoE) is typically used to power up devices such as Access points, IP Cameras and IP Phones connected to the device's Ethernet ports. The total PoE available power is 30W to be shared by the 4 LAN ports.

The power allocation is as follows:

- 1 x POE+ (AT Type2, Class 4) port (25.5 W)
- 1 x POE (AT Type1 or AF-Class 0/3) ports (15.4 W)
- 4 x POE (AF, Class 1, Class 2) Ports (3.84W or 6.49W)

Device Detection and Power Allocation

The router will detect a Cisco Pre-standard or an IEEE-compliant PD when the PoE is enabled and the connected device is not being powered by an AC adapter.

After device detection, the router will determine the power requirements based on power classification class. Depending on the available power in the power budget, the router determines if a port can be powered. The router initially allocates this power when it detects and powers the device. Power negotiation using CDP/LLDP protocols happens thereafter. Supported protocols for power negotiation are CDP for Cisco PD, and LLDP for non-Cisco PDs. Maximum power budget for 4 LAN ports combined at any time is 30.8W. On reload the PoE ports are powered down until the unit reboots.

Command Line Interface

This section describes the CLI to use for configuring and displaying PoE.

To configure auto or off:

power inline *auto* | *never*

Configuration example:

```
Router#config terminal
Router#interface g0/1/<0,1,2,3>
Router(config-if)#power inline {auto|never}
```

To Verify your configuration:

```
Router#show power inline
Available:30.0(w)  Used:22.5(w)  Remaining:7.5(w)

Interface Admin  Oper       Power   Device              Class Max
                            (Watts)
--------- ------ ---------- ------- ------------------- ----- ----
Gi0/1/0   auto   on         15.4    Ieee PD             4     30.0
Gi0/1/1   auto   off        0.0     n/a                 n/a   30.0
Gi0/1/2   auto   off        0.0     n/a                 n/a   30.0
Gi0/1/3   auto   on         7.1     IP Phone 8845       2     30.0
Router#
```

To show power on a particular interface:

```
Router#show power inline {interface-id}
```

Displays PoE status for a router for the specified interface.

show power inline *interface-id* detail

To show power consumption:

```
Router#show power
Main PSU :
    Total Power Consumed: 20.99 Watts
    Configured Mode : N/A
    Current runtime state same : N/A
    PowerSupplySource : External PS
POE Module :
    Configured Mode : N/A
    Current runtime state same : N/A
    Total power available : 30 Watts
Router#
```

The list of commands for debugging PoE follows:

Command	Description
Debug ilpower controller	Display PoE controller debug messages
Debug ilpower event	Display PoE event debug messages
Debug ilpower port	Display PoE port manager debug messages
Debug ilpower powerman	Display PoE power management debug messages
Debug ilpower cdp	Display PoE CDP debug messages
Debug ilpower registries	Display PoE registries debug messages

Command	Description
Debug ilpower scp	Display PoE scp debug messages

CHAPTER 18

Cellular Pluggable Interface Module Configuration Guide

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.

CHAPTER 19

Cisco Wi-Fi Interface Module (WIM)

This chapter contains the following sections:

- Cisco Wi-Fi Interface Module (WIM) Overview, on page 155
- Hardware Overview, on page 156
- Software Overview, on page 157
- WIM Configuration Overview, on page 158
- Typical Deployment Scenarios, on page 165
- Wi-Fi Mode Conversion, on page 173
- Upgrading the Firmware on the Cisco Wi-Fi Interface Module, on page 175
- Connection to the Ethernet Switch, on page 177
- Factory Reset, on page 182
- Flexible Antenna Port, on page 183
- Related Documentation, on page 185

Cisco Wi-Fi Interface Module (WIM) Overview

This section provides an overview of the Cisco Wi-Fi Interface Module (WIM). The PID is WP-WIFI6-*x*.

Highlights of the WIM are:

- Pluggable 802.11ax module for Cisco Catalyst IR1800 series
- WiFi-6 (802.11ax), 2x2 MIMO with 2 spatial streams
- Supports WGB and uWGB Mode
- EWC Controller Support
- Extended Temperature Range
- Field Replaceable Unit (FRU), however does not support OIR (Online Insertion and Removal)
- Versatile RF coverage with external RP-SMA antenna connectors
- Flexible Antenna Port feature support
- Parity with Cisco AP 9105AXI

Hardware Overview

The following graphic shows the front panel of the WIM.

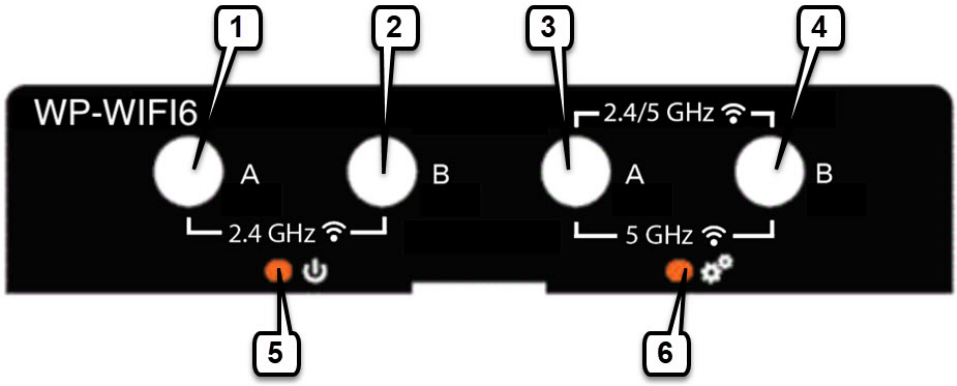


Table 14: WIM Front Panel

Item	Description
1	WLAN ANT 0 2.4 GHz
2	WLAN ANT 1 2.4 GHz
3	WLAN ANT 0 2.4/5 GHz
4	WLAN ANT 1 2.4/5 GHz
5	Enable LED
6	Wi-Fi LED

The following table describes the Enable LED:

LED Status	Description
Off	No Power
Yellow	Power is on, module is not yet functional
Green	Module is fully functional

The following table describes the Wi-Fi LED:

LED Status	Status Type	Description
Solid Green	Association Status	Normal operating condition, but no wireless client associated.
Solid Blue	Association Status	Normal operating condition, at least one wireless client associated.
Solid Green	Boot Loader Status	Executing Boot Loader
Flashing Green	Boot Loader Status	Boot Loader Error, signing verification error.
Flashing Blue	Operating Status	Software upgrade in progress.
Alternate between Green and Red	Operating Status	Discovery/Join process is in progress.
Cycle through Red-Off-Green-Off-Blue-Off	Access Point operating system error	General warning; insufficient inline power.

Software Overview

The WIM is supported on all four models of the IR1800 series.

- The IR1800 must be running IOS-XE version 17.7.1 or greater
- The built-in access point (AP) runs on the image family ap1g8
- The WIM supports 802.11ax, 2x2:2, 1.488Gbps (PHY rate)

The WIM supports all three modes with 17.8.1 IOS-XE release.

- Embedded Wireless Controller (EWC) – scale as C9105 + IR1800
- Control and Provisioning of Wireless Access Points (CAPWAP) - attached to WLC
- Workgroup Bridge (WGB) - Currently cannot support WGB and EWC or CAPWAP concurrently.

The WIM has two different OIDs:

- Cisco Wide Pluggable Form Factor WIFI6 AP Module for Industrial Routers
  - 1.3.6.1.4.1.14179.2.2.1.1.22.116
- Cisco Embedded Wireless Controller on Catalyst Access Point
  - 1.3.6.1.4.1.9.1.3088

For a complete list of supported features, see the following link and search for Cisco Catalyst Series 9105AXI Access Point.

https://www.cisco.com/c/en/us/td/docs/wireless/access_point/feature-matrix/ap-feature-matrix.html

Management Commands

Commands used to view the status of the module are:

- show platform
- show inventory
- show hw-module subslot 0/3 attribute
- show hw-module subslot 0/3 oir
- show hw-module subslot 0/3 tech-support
- show logging

Commands used to configure the module are:

- hw-module subslot 0/3 maintenance enable | disable
- hw-module subslot 0/3 stop | start | reload [force]
- hw-module subslot 0/3 error-recovery password_reset

WIM Configuration Overview

The following are some of the product configuration details:

- The module is fixed to subslot 0/3
- The Wi-Fi interface to communicate with the AP is known as Wl0/1/4
- By default, Wl0/1/4 is in VLAN 1
  - If a DHCP pool is set-up on VLAN1, AP (and associated clients) will get an IP address.
- The module cannot be hot-swapped but is field replaceable.
- The host router must be manually reloaded after the module is inserted.
- Feature set is aligned on AP 9105AXI. See the Feature Matrix for Cisco Wireless Access Points

Wi-Fi landing page feature (Web-based authentication) support. See the Web-Based Authentication chapter in the Cisco Embedded Wireless Controller on Catalyst Access Points Configuration Guide.

The IR1800 Wl0/1/4 Interface

On the IR1800, Wlan-GigabitEthernet0/1/4 is a layer 2 interface that IOX-XE uses to connect to the WP-WiFi6-x module. It supports both access and trunk mode.

```
IR1821#show vlan
VLAN Name                             Status    Ports
---- -------------------------------- --------- ----
1    default                          active    Gi0/1/0, Gi0/1/1, Gi0/1/2
                                                Gi0/1/3, Wl0/1/4
```

Traffic forwarding is performed by the IR1800 ethernet switch.

- Layer 2 forwarding from/to assigned VLAN
- Layer 3 forwarding through associated VLAN x interface is handled by the host CPU. For example: routing, NAT, tunneling etc.

```
IR1831#show inter wl0/1/4
Wlan-GigabitEthernet0/1/4 is up, line protocol is up (connected)
  Hardware is WLAN Gigabit Ethernet, address is c44d.849b.0a8c (bia c44d.849b.0a8c)
  Description: Internal switch interface connecting to the embedded AP
  MTU 1500 bytes, BW 1000000 Kbit/sec, DLY 10 usec,
     reliability 255/255, txload 1/255, rxload 1/255
  Encapsulation ARPA, loopback not set
  Keepalive not supported
  Full-duplex, 1000Mb/s, link type is auto, media type is 10/100/1000BaseTX
  input flow-control is off, output flow-control is unsupported
  ARP type: ARPA, ARP Timeout 04:00:00
  Last input 00:00:03, output 00:00:08, output hang never
  Last clearing of "show interface" counters never
  Input queue: 0/375/0/0 (size/max/drops/flushes); Total output drops: 0
  Queueing strategy: fifo
  Output queue: 0/40 (size/max)
  5 minutes input rate 1000 bits/sec, 1 packets/sec
  5 minutes output rate 0 bits/sec, 0 packets/sec
     4479 packets input, 389481 bytes, 0 no buffer
     Received 1579 broadcasts (2499 multicasts)
     0 runts, 0 giants, 0 throttles
     0 input errors, 0 CRC, 0 frame, 0 overrun, 0 ignored
     0 watchdog, 2499 multicast, 0 pause input
     0 input packets with dribble condition detected
     588 packets output, 74648 bytes, 0 underruns
     Output 13 broadcasts (178 multicasts)
     0 output errors, 0 collisions, 1 interface resets
     111 unknown protocol drops
     0 babbles, 0 late collision, 0 deferred
     0 lost carrier, 0 no carrier, 0 pause output
     0 output buffer failures, 0 output buffers swapped out
```

WIM Baseline Configuration

The IR1800 will need a basic configuration to enable Wi-Fi.

The high level steps are:

1. Check that Wl0/1/4 is assigned to VLAN1 (default)
2. Assign an IP address to interface VLAN1
3. Create a DHCP pool servicing interface VLAN1
4. Once the WP-WiFi6-x boots, check it gets an IP address

Other items to check are:

- Check and modify, if necessary, NAT, ACL, and default route to work with the interface
- Wl0/1/4 interface can be configured with switchport mode access or trunk with associated VLAN(s) number

Configuration Steps for the WIM

To configure the global parameters for your router, follow these steps:

Procedure

	Command or Action	Purpose
Step 1	**ip dhcp pool** *name* **Example:** `Device(config)#ip dhcp pool wireless`	Use this command to create a name for the DHCP server address pool and to enter the DHCP pool configuration mode.
Step 2	**network** *ip address subnet mask* **Example:** `Router(dhcp-config)#network 10.10.10.0 255.255.255.0`	Use this command to create a DHCP pool of IP addresses to be used by the Switched Virtual Interface (SVI) (Refer Step 11 and further for SVI).
Step 3	**default-router** *ip address* **Example:** `Router(dhcp-config)#default-router 10.10.10.1`	Use this command to assign the default gateway to clients of this DHCP pool.
Step 4	**dns-server** *ip address* **Example:** `Router(dhcp-config)#dns-server 192.0.2.1`	Use this command to assign the DNS server IP address to clients in this DHCP pool.
Step 5	**interface Wlan-GigabitEthernet** *slot* **Example:** `Router(config)#interface Wlan-GigabitEthernet0/1/4`	Use the **interface Wlan-GigabitEthernet** command to add the interface and set the IP address.
Step 6	**switchport access vlan** *number* **Example:** `Router(config-if)#switchport access vlan 1`	Use the **switchport access vlan** command to assign the port or range of ports into access ports. **Note** The default VLAN is 1. Depending on your topology or needs, you may have to set up a separate VLAN.
Step 7	**switchport mode access** **Example:** `Router(config-if)#switchport mode access`	Use the **switchport mode access** command to configure the VLAN membership mode.
Step 8	**interface vlan** *number* **Example:**	Use the **interface vlan** *number* command in the configuration mode to create a Switched

	Command or Action	Purpose
	`Router(config)#interface vlan 1`	Virtual Interface (SVI) and enter the interface configuration (VLAN) mode for a specific VLAN or a range of VLANs.
Step 9	**description** *name* **Example:** `Router(config-if)#description Wireless`	Use this command to add a description for the Switched Virtual Interface (SVI).
Step 10	**ip address** *ip-address subnet_mask* **Example:** `Router(config-if)#ip address 10.10.10.1 255.255.255.0`	Use this command to assign an IP address from the DHCP Pool.
Step 11	**ip nat inside** **Example:** `Router(config)#ip nat inside`	Connects the interface to the inside network, which is subject to NAT.
Step 12	**ip nat inside source list** *number* **interface Wlan-GigabitEthernet** *slot* **overload** **Example:** `Router(config)#ip nat inside source list 10 interface Wlan-GigabitEthernet0/1/4 overload`	Use this command to establish dynamic source translation, specifying the access list. **Note** This step is only valid if using WGB mode.
Step 13	**ip route 10.10.10.10 10.10.10.10** *default gateway ip-address* **Example:** `Router(config)#ip route 10.10.10.10 10.10.10.10 192.0.2.1`	Use this command to direct all the traffic to the default gateway of the router.
Step 14	**ip access-list standard** *number* **Example:** `Router(config)#ip access-list standard 10`	Use the ip access-list standard command to filter the traffic based on a set of rules.
Step 15	*number* **permit** *ip_address wildcard_mask* **Example:** `Router(config)#10 permit 10.10.10.0 0.0.0.255`	Use this command to create ACL entries to permit or deny traffic.

Normal Bootup

The Wi-Fi module is powered on as soon as the host router reloads. The Wi-Fi module state turns from 'booting' to 'ok' when the host receives the 'ready' signal from Wi-Fi module. For example:

```
#show platform
Chassis type: IR1835-K9
Slot      Type                State                 Insert time (ago)
--------- ------------------- --------------------- -----------------
0         IR1835-K9           ok                    15:25:47
 0/0      IR1835-1x1GE        ok                    15:23:37
 0/1      IR1835-ES-4         ok                    15:23:36
 0/3      WP-WIFI6-B          ok                    00:00:07
R0        IR1835-K9           ok, active            15:25:47
F0        IR1835-K9           ok, active            15:25:47
P0        PWR-12V             ok                    15:23:59
GE-POE    Unknown             ok                    15:23:59

# show logging
Apr  6 18:05:41.992 CST: %IOSXE_OIR-6-INSSPA: SPA inserted in subslot 0/3
Apr  6 18:05:54.886 CST: new extended attributes received from iomd(slot 0 bay 3 board 0)
Apr  6 18:05:55.226 CST: %SPA_OIR-6-ONLINECARD: SPA (WP-WIFI6) online in subslot 0/3
```

Manual Reload

In some scenarios, you might have to manually reload the WIM. For example, an error in the module or to perform a factory reset. Use the **hw-module subslot 0/3** *stop* | *start* | *reload [force]* command.

The output of the **show platform** command as well as syslog output will show the module booting and then turn “ok”.

```
#show platform
Chassis type: IR1835-K9
Slot      Type                State                 Insert time (ago)
--------- ------------------- --------------------- -----------------
0         IR1835-K9           ok                    15:24:10
 0/0      IR1835-1x1GE        ok                    15:22:00
 0/1      IR1835-ES-4         ok                    15:21:59
 0/3      WP-WIFI6-B          booting               00:00:01
R0        IR1835-K9           ok, active            15:24:10
F0        IR1835-K9           ok, active            15:24:10
P0        PWR-12V             ok                    15:22:2
```

```
#show platform
Chassis type: IR1835-K9
Slot      Type                State                 Insert time (ago)
--------- ------------------- --------------------- -----------------
0         IR1835-K9           ok                    15:25:47
 0/0      IR1835-1x1GE        ok                    15:23:37
 0/1      IR1835-ES-4         ok                    15:23:36
 0/3      WP-WIFI6-B          ok                    00:01:38
R0        IR1835-K9           ok, active            15:25:47
F0        IR1835-K9           ok, active            15:25:47
P0        PWR-12V             ok                    15:23:59
GE-POE    Unknown             ok                    15:23:59

# show logging
Apr  7 09:34:45.143 CST: %IOSXE_OIR-6-SOFT_RELOADSPA: SPA(WP-WIFI6) reloaded on subslot 0/3
```

```
Apr  7 09:34:45.143 CST: %SPA_OIR-6-OFFLINECARD: SPA (WP-WIFI6) offline in subslot 0/3
Apr  7 09:36:25.863 CST: %SPA_OIR-6-ONLINECARD: SPA (WP-WIFI6) online in subslot 0/3
```

Bootloader Mode

In the scenario where the user must enter bootloader mode (for example, to recover from a continuous software crash), best practice is to issue the command **hw-module subslot 0/3 maintenance enable** on the host router. Otherwise, the host router would power cycle the WIM forcefully in 10 minutes.

Router Commands

```
Router# hw-module subslot 0/3 maintenance enable
Router# hw-module subslot 0/3 reload force
Apr  7 09:49:05.700 CST: %IOSXE_OIR-6-SOFT_RELOADSPA: SPA(WP-WIFI6) reloaded on subslot 0/3
Apr  7 09:49:05.700 CST: %SPA_OIR-6-OFFLINECARD: SPA (WP-WIFI6) offline in subslot 0/3

Router#show platform
Chassis type: IR1835-K9
Slot      Type                State    XBase-X SFP Module Plugged in
bcm47622_eth-0
MAC: e8:eb:34:9c:14:c0
Hit ESC key to stop autoboot:  0
u-boot>             Insert time (ago)
--------- ------------------- --------------------- -----------------
0         IR1835-K9           ok                    15:52:35
 0/0      IR1835-1x1GE        ok                    15:50:24
 0/1      IR1835-ES-4         ok                    15:50:24
 0/3      WP-WIFI6-B          booting               00:06:59
Router#
Apr  7 09:56:15.741 CST: %SPA_OIR-6-OFFLINECARD: SPA (WP-WIFI6) offline in subslot 0/3
```

WIM Power Down

The host router will power down the WIM if the WIM reloads 5 times within 20 minutes (for example, a continuous software crash):

```
*Apr  7 10:34:57.412 CST: %SPA_OIR-6-ONLINECARD: SPA (WP-WIFI6) online in subslot 0/3
*Apr  7 10:36:19.021 CST: %SPA_OIR-6-OFFLINECARD: SPA (WP-WIFI6) offline in subslot 0/3
*Apr  7 10:37:59.128 CST: %SPA_OIR-6-ONLINECARD: SPA (WP-WIFI6) online in subslot 0/3
*Apr  7 10:39:18.942 CST: %SPA_OIR-6-OFFLINECARD: SPA (WP-WIFI6) offline in subslot 0/3
*Apr  7 10:41:00.072 CST: %SPA_OIR-6-ONLINECARD: SPA (WP-WIFI6) online in subslot 0/3
*Apr  7 10:42:15.864 CST: %SPA_OIR-6-OFFLINECARD: SPA (WP-WIFI6) offline in subslot 0/3
*Apr  7 10:43:57.507 CST: %SPA_OIR-6-ONLINECARD: SPA (WP-WIFI6) online in subslot 0/3
*Apr  7 10:45:06.049 CST: %SPA_OIR-6-OFFLINECARD: SPA (WP-WIFI6) offline in subslot 0/3
*Apr  7 10:46:46.167 CST: %SPA_OIR-6-ONLINECARD: SPA (WP-WIFI6) online in subslot 0/3
*Apr  7 10:48:12.425 CST: %SPA_OIR-3-SPA_POWERED_OFF: subslot 0/3: SPA WP-WIFI6 powered off
 after 5 failures within 1200 seconds
*Apr  7 10:48:12.425 CST: %SPA_OIR-6-OFFLINECARD: SPA (WP-WIFI6) offline in subslot 0/3


Router# show platform
Chassis type: IR1835-K9
Slot      Type                State                 Insert time (ago)
--------- ------------------- --------------------- -----------------
0         IR1835-K9           ok                    16:45:16
 0/0      IR1835-1x1GE        ok                    16:43:06
 0/1      IR1835-ES-4         ok                    16:43:05
 0/3      WP-WIFI6-B          out of service        00:00:39
```

```
R0        IR1835-K9          ok, active           16:45:16
F0        IR1835-K9          ok, active           16:45:16
P0        PWR-12V            ok                   16:43:28
GE-POE    Unknown            ok                   16:43:28
```

Connect to the Access Point through the Router Console

To connect to the WIM, first establish a connection to the host router through the console, ssh protocol, or telnet protocol.

Then re-direct to the Access Point from the host router. See the following example:

```
Router# hw-module subslot 0/3 ?
  error-recovery  Error Recovery
  maintenance     Maintenance mode
  oir             Spa OIR
  reload          Restart the target subslot
  start           Activate the target subslot
  stop            Deactivate the target subslot
Router#

Router# hw-module session 0/3
Establishing session connect to subslot 0/3
To exit, type ^a^q
picocom v3.1
port is: /dev/ttyWIFI
flowcontrol: none
baudrate is: 9600
parity: none
databits are: 8
stopbits are: 1
escape is: C-a
local echo is: no
noinit is: no
noreset is: no
hangup is: no
nolock is: yes
send_cmd is: sz -vv
receive_cmd is: rz -vv -E
imap is:
omap is:
emap is: crcrlf,delbs,
logfile is: none
initstring: none
exit_after is: not set
exit is: no
Type [C-a] [C-h] to see available commands
Terminal ready
Username:
```

Disconnect from the Access Point by performing the following:

```
issue ^a^q
Username:
Terminating...
Skipping tty reset...
Thanks for using picocom
Router#
```

Typical Deployment Scenarios

Three of the typical scenarios that the WIM can be deployed are described in this section.

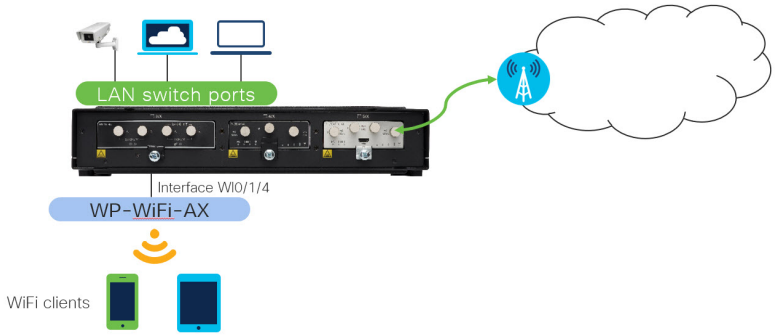
The Wireless Interface Module closely resembles the Cisco Catalyst Series 9105AXI Access Point in functionality.

Embedded Wireless Controller

The first is Embedded Wireless Controller (EWC). This scenario offers:

- Self-Management
- Traffic Local-Switching
- May manage cascaded AP, aligned on C9105 + IR1800 performances
- WebUI management
- Cisco Wireless Mobility management application (iPhone/Android)

EWC mode is typically used for Mass Transit/Transportation Remote & Mobile Assets.



When the Wireless Interface Module is running in EWC mode, it acts as a wireless controller and an Access Point (usually called internal AP). EWC manages other APs in a similar way as dedicated wireless controller (like C9800 series).

In a Cisco EWC network, an Access Point (AP) running the wireless controller function is designated as the active AP. The other access points, which are managed by this active AP, are referred to as subordinate APs.

The image used for EWC mode is C9800-AP-iosxe-wlc.bin.

The active EWC has two roles:

- Functions and operates as a Wireless LAN Controller (WLC) to manage and control the subordinate APs. The subordinate APs operate as lightweight access points to serve clients.
- Operates as an access point to serve clients.

Further information on the Cisco Embedded Wireless Controller can be found in the following:

https://www.cisco.com/c/en/us/products/collateral/wireless/catalyst-9800-series-wireless-controllers/q-and-a-c67-743152.html

https://www.cisco.com/c/en/us/products/collateral/wireless/embedded-wireless-controller-catalyst-access-points/white-paper-c11-743398.html

EWC Baseline Configuration

The following section shows the basic configuration to activate EWC mode.

The high level steps are:

1. Create a VLAN. For this example, VLAN10 is dedicated to Wi-Fi users.

```
interface Wlan-GigabitEthernet0/1/4
 switchport access vlan 10
 switchport mode access

IR1831#show vlan
VLAN Name                             Status    Ports
---- -------------------------------- --------- -------
1    default                          active    Gi0/1/0, Gi0/1/1, Gi0/1/2, Gi0/1/3
10   VLAN0010                          active    Wl0/1/4
```

2. Assign an IP address to interface VLAN10.

```
interface Vlan10
 ip address 192.168.80.1 255.255.255.0
 ip nat inside
```

3. Create a DHCP pool servicing interface VLAN10.

```
ip dhcp pool VLAN10
 network 192.168.80.0 255.255.255.0
 dns-server 8.8.8.8
 default-router 192.168.80.1
```

4. Check and modify, if necessary, NAT, ACL, and default route to work with VLAN10.

Other items to consider are:

- Day 0 configuration of Wi-Fi can be done via console access, or through the Cisco Wireless Mobility application on a mobile phone. Initial configuration dialog is IOS-XE based.
- For day 0 configuration done through the WebUI or Cisco Wireless Mobility application, it is recommended to reload the Wi-Fi module making sure it obtains an IP address from configured VLAN pool, for example, VLAN1.
- For WebUI day 0, it may not work if using an IP address different from the default IP address:192.168.0.1, which also collides with day0 IP address of the IR1800 IOS-XE.

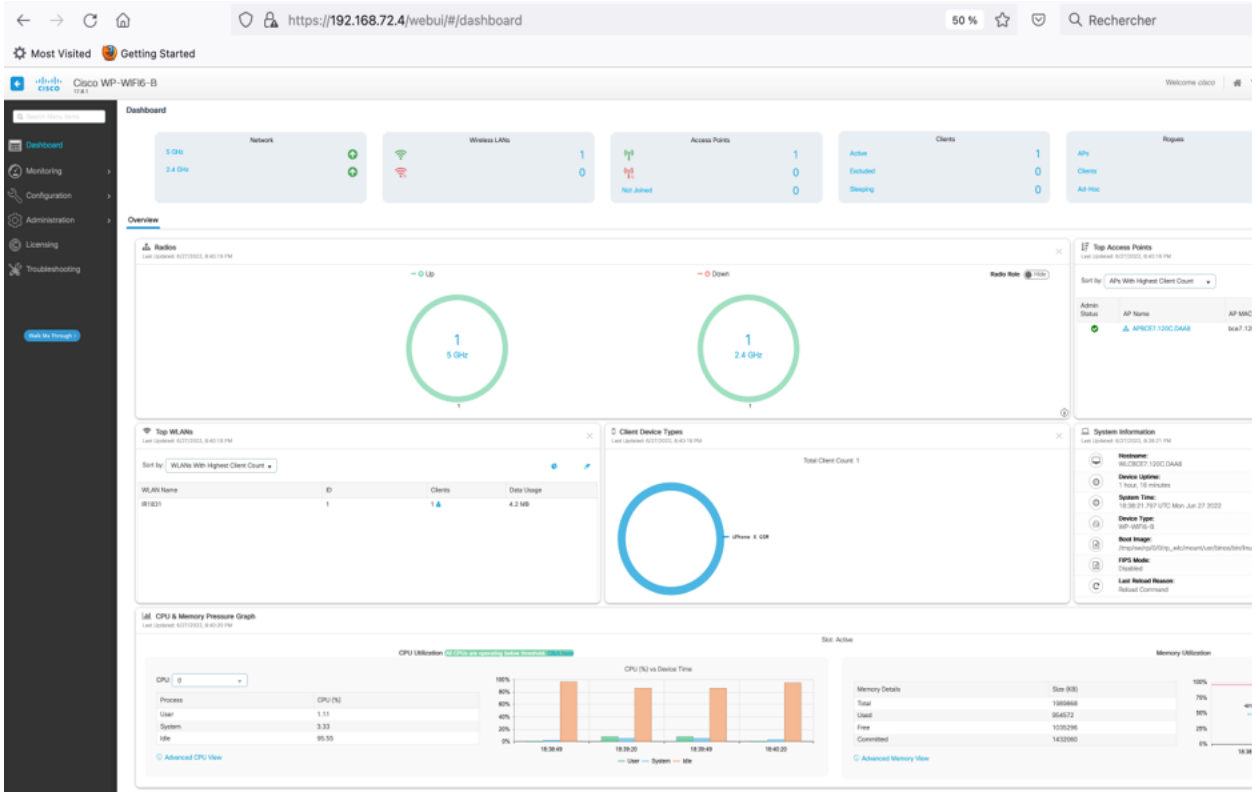
EWC mode – WebUI Management

This section provides steps for configuring the WIM in EWC mode through the WebUI.

Procedure

Step 1 Open the WebUI from a browser. Use the IP address allocated from DHCP.

Step 2 The WebUI dashboard appears.



Step 3 Once connected to the Wireless LAN Controller (WLC), configuration is performed as any other access point. Refer to the following resources for additional information:

Overview of Cisco Embedded Wireless Controller on Catalyst Access Points

Cisco Embedded Wireless Controller on Catalyst Access Points Configuration Guide, IOS XE

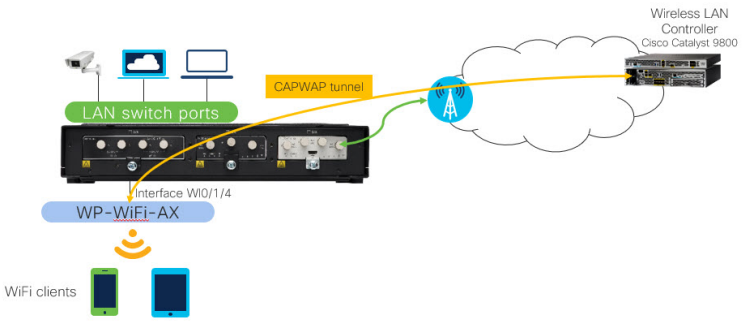
Control And Provisioning of Wireless Access Points (CAPWAP)

The second is Control And Provisioning of Wireless Access Points (CAPWAP). This scenario offers:

- Central-Management
- Flexible Traffic Forwarding Model
- AP Join process – tested with DHCP option 43 – to join specific WLC
- CAPWAP AP modes: Local, FlexConnect, Sniffer, and Monitor.

Note There is no local WebUI support for CAPWAP mode.

CAPWAP mode is typically used for Extended Enterprise Stationary & Mobile Use Cases.



Once an IP address is set on the Wi-Fi, it communicates and is managed through its WLC. The IR1800 is managed by vManage (SDWAN) or IOT OD.

When the Wireless Interface Module is running in CAPWAP mode, it is managed by an external wireless controller, such as a Cisco 9800 series. Cisco lightweight access points use the IETF standard Control and Provisioning of Wireless Access Points protocol (CAPWAP) to communicate between the controller and other lightweight access points on the network.

The image used for CAPWAP mode is ap1g8.

Further information on CAPWAP can be found in the following:

https://www.cisco.com/c/en/us/products/collateral/wireless/catalyst-9800-series-wireless-controllers/guide-c07-743627.html

https://www.cisco.com/c/en/us/td/docs/wireless/controller/9800/17-5/config-guide/b_wl_17_5_cg.html

https://www.cisco.com/c/en/us/support/docs/wireless-mobility/wireless-lan-wlan/97066-dhcp-option-43-00.html

CAPWAP Baseline Configuration

The following section shows the basic configuration on the Wl0/1/4 interface to activate CAPWAP mode.

Note If the AP is already in CAPWAP mode, the AP does not reboot. If the AP is in EWLC or WGB mode, the AP reboots after which the mode is changed to CAPWAP.

The high level steps are:

1. Create and assign an IP address to interface VLAN10, dedicated to Wl0/1/4

```
interface Vlan10
 ip address 192.168.73.1 255.255.255.0
 ip nat inside
```

2. Assign Wl0/1/4 to VLAN10

```
interface Wlan-GigabitEthernet0/1/4
 switchport access vlan 10
 switchport mode access
```

```
IR1831#show vlan
VLAN Name                             Status    Ports
---- -------------------------------- --------- -------
1    default                          active    Gi0/1/0, Gi0/1/1, Gi0/1/2, Gi0/1/3
10   VLAN0010                         active    Wl0/1/4
```

3. Create a DHCP pool servicing interface VLAN10 with DHCP option 43 for WLC address.

```
ip dhcp pool VLAN10
 network 192.168.73.0 255.255.255.0
 dns-server 8.8.8.8
 default-router 192.168.73.1
 option 43 hex f108c0a80a05c0a80a14
 ! With hex value for the IP addresses of the WLC coded in hexadecimal
 ! 192.168.10.5 = %xc0a80a05 and 192.168.10.20 = %xc0a80a14
 ! Prefix f108
```

When DHCP server is programmed to offer WLC IP addresses as Option 43 for other Cisco LAPs, the sub-option TLV block is defined in this way:

- **Type** - 0xf1 (decimal 241
- **Length** - Number of controller IP addresses * 4.
- **Value** - List of the WLC management interfaces, typically translated to hexadecimal values.
  - %x f104xxxxxxxx – single WLC IP address
  - %x f108xxxxxxxxxxxxxxxx – dual WLC IP addresses

The CAPWAP Joining Process

This section describes the joining process of the Cisco Wi-Fi Interface Module when configured in CAPWAP mode.

Refer to this knowledge article for a detailed description of the Cisco AP Join process.

1. The AP boots and gets an IP address from the IR1800 DHCP pool.

```
[* 07/12/2022 15:24:38.0000] AP IPv4 Address updated from 0.0.0.0 to 192.168.73.2
```

2. The AP gets a Wireless LAN Controller IP address from the IR1800 DHCP pool option 43.

```
[* 07/12/2022 15:24:41.3180] Got WLC address 192.168.72.5 from DHCP.
```

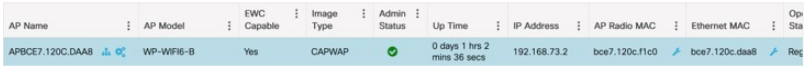
3. The AP connects to the Wireless LAN Controller using the learnt IP address.

```
[*07/12/2022 15:07:32.0120] Discovery Request sent to 192.168.72.5, discovery type DHCP(2)
[*07/12/2022 15:07:32.0160] Discovery Response from 192.168.72.5
```

4. The AP joins the Wireless LAN Controller, potentially getting a SW upgrade, and configuration.

```
[*07/12/2022 15:24:38.0000] CAPWAP State: DTLS Setup
[*07/12/2022 15:24:38.6700]
[*07/12/2022 15:24:38.6700] CAPWAP State: Join
[*07/12/2022 15:24:38.6740] Sending Join request to 192.168.72.5 through port 5270
[*07/12/2022 15:24:38.6830] Join Response from 192.168.72.5
[*07/12/2022 15:25:18.7630] AC accepted previous sent request with result code: 0
[*07/12/2022 15:25:19.0690] Received wlcType 2, timer 30
[*07/12/2022 15:25:19.2220]
[*07/12/2022 15:25:19.2220] CAPWAP State: Image Data
[*07/12/2022 15:25:19.2230] AP image version 17.8.0.144 backup 17.6.0.75, Controller
17.8.0.144
[*07/12/2022 15:25:19.2230] Version is the same, do not need update
```

5. The AP is now managed from the Wireless LAN Controller.

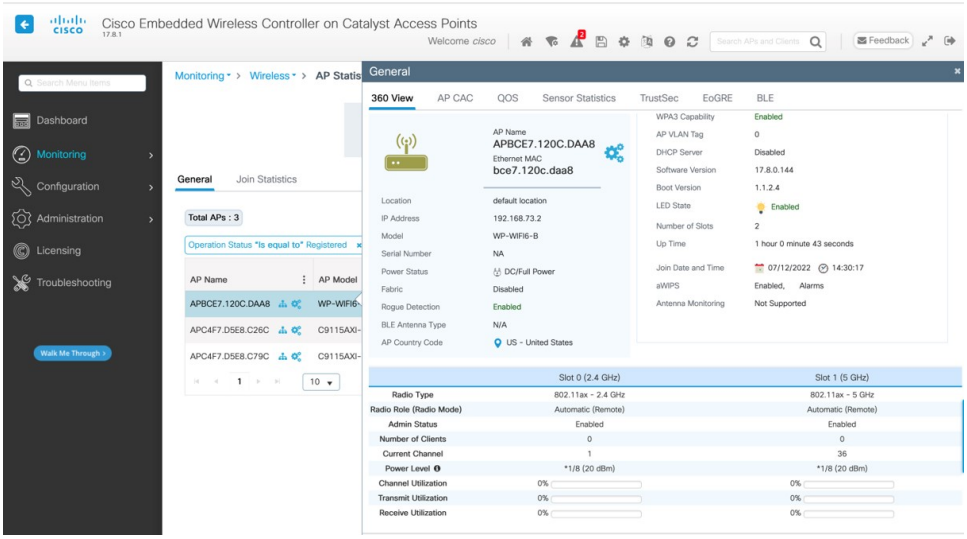


AP Name	AP Model	EWC Capable	Image Type	Admin Status	Up Time	IP Address	AP Radio MAC	Ethernet MAC	Op Sta
APBCE7.120C.DAA8	WP-WIFI6-B	Yes	CAPWAP		0 days 1 hrs 2 mins 36 secs	192.168.73.2	bce7.120c.f1c0	bce7.120c.daa8	Reg

CAPWAP Management from Wireless LAN Controller

The Cisco Wi-Fi Interface Module can be managed in CAPWAP mode from the Wireless LAN Controller. This GUI is the management interface of the WLC managing all CAPWAP APs. In the following example, the WLC is a C9115 AXI.

The following figure shows an example of the WebUI:

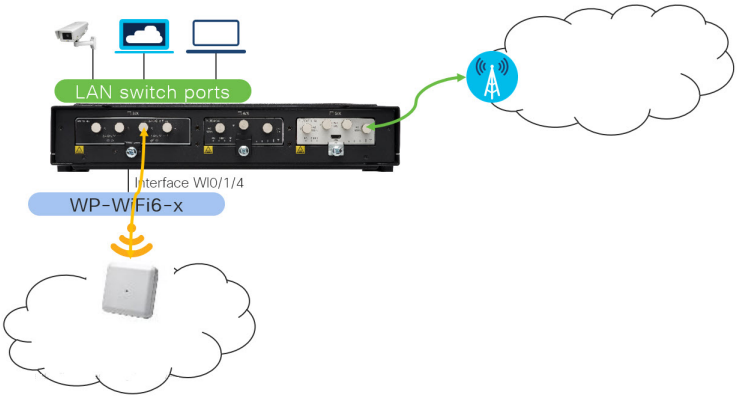


Workgroup Bridge (WGB)

The third scenario is Workgroup Bridge (WGB). WGB offers:

- Low Cost High-Speed Wi-Fi Uplink
- No concurrent mode, for example EWC and WGB
- Only one radio is allowed to operate in uWGB or WGB mode
- IR1800 SDWAN support is available in IOS-XE 17.10.1
- WGB supports up to 20 wired clients
- uWGB supports a single client MAC address, for example, VLAN10 interface in a configuration where Wl0/1/4 is a routed interface for Wi-Fi as backhaul link

WGB mode is typically used for Data Offloading Over Infrastructure Wi-Fi.



Workgroup Bridge mode is a special mode. An Access Point running in this mode works like a wireless station. It is normally used to bridge wired clients (connected to it via its Gigabit port) to a wireless infrastructure.

An example of one usage scenario is to place one WGB on a vehicle and connect devices like a camera and sensor to its Gigabit port. WGB helps bridge all the traffic via its Wi-Fi uplink.

The image used for WGB mode is ap1g8. This is the same as CAPWAP mode, but only one mode is active.

There is much more information available at the following:

https://www.cisco.com/c/en/us/td/docs/wireless/controller/8-10/config-guide/b_cg810/workgroup_bridges.html#info-wgb-on-cos-aps

Look for "Cisco Wave 2 Access Points as Workgroup Bridges".

https://www.cisco.com/c/en/us/td/docs/wireless/outdoor_industrial/iw6300/software/config/guide/b-cos-wgb-deploy-guide.pdf

From Cisco IOS-XE Release 17.8.1, the Universal WGB mode is supported for WIM.

Universal WGB (uWGB) is a complementary mode of WGB feature that acts as a wireless bridge between the wired client connected to uWGB and wireless infrastructure including Cisco and non-Cisco wireless network.

For more information on WGB and uWGB configuration, see the Cisco Industrial Wireless Workgroup Bridge and Universal WGB Deployment Guide.

WGB/uWGB Baseline Configuration

The following section shows the minimum IOS-XE configuration done on the Wl0/1/4 interface.

WP-WiFi6-x acts as a workgroup bridge. Check the **show configuration** once opening session to 0/3.

The 5GHz radio (dot11 radio 1) is set-up to attach to a 3rd party AP.

- 2.4GHz radio (dot11 radio 0) is turned off
- uWGB configuration – to connect 3rd party AP

The high level steps are:

1. Configure an SSID profile.

```
configure ssid-profile Test ssid Free authentication psk cisco12345 key-management wpa2
```

2. Configure radio interface to WGB/uWGB mode and map the SSID profile. Select authentication as dictated by the wireless infrastructure.

```
configure dot11radio 1 mode uwgb c44d.849b.0a8c ssid-profile Test
 configure dot11radio 1 encryption mode ciphers aes-ccm
 configure dot11radio 1 enable
```

3. Configure unused radio as root-AP and off as today, single radio is used by WGB/uWGB.

```
configure dot11radio 0 mode root-ap
 configure dot11radio 0 disable
```

uWGB Show configuration

The following are examples of a uWGB configuration.

```
APBCE7.120C.DAA8#show config
AP Name              : APBCE7.120C.DAA8
AP Mode              : WorkGroupBridge
CDP State            : Enabled
Watchdog monitoring  : Enabled
SSH State            : Disabled
AP Username          : Cisco
Session Timeout      : 300

Radio and WLAN-Profile mapping:-
=====================================
Radio ID    Radio Mode    SSID-Profile      SSID       Authentication
-------------------------------------------------------------------
1           UWGB          Test              Free       PSK
```

Radio Configuration:

```
Radio Id             : 0
   Admin state       : DISABLED
   Mode              : RootAP
   Beacon Period     : 100 mSec
Radio Id             : 1
   Admin state       : ENABLED
   Mode              : UWGB
   Uclient mac       : C44D.849B.0A8C
   Current state     : WGB
   UClient timeout   : 0 Sec
   Dot11 type        : 11ax
   Encryption mode   : AES128

WGB specific configuration:-
=====================================
WGB Radio Id         : NA
  Mode State         : NA
  SSID Profile       : NA
UWGB Radio Id        : 1
  Mode Enable        : Enable
  SSID Profile       : Test
  Uclient MAC Address: C44D.849B.0A8C
```

Check attached wired device on the IR1800:

```
#show wgb bridge
***Client ip table entries***
mac vap              port   vlan_id    seen_ip        confirm_ago  fast_brg
10:DD:B1:CE:B2:E6    0      wired0     192.168.10.25  0.016000     true
```

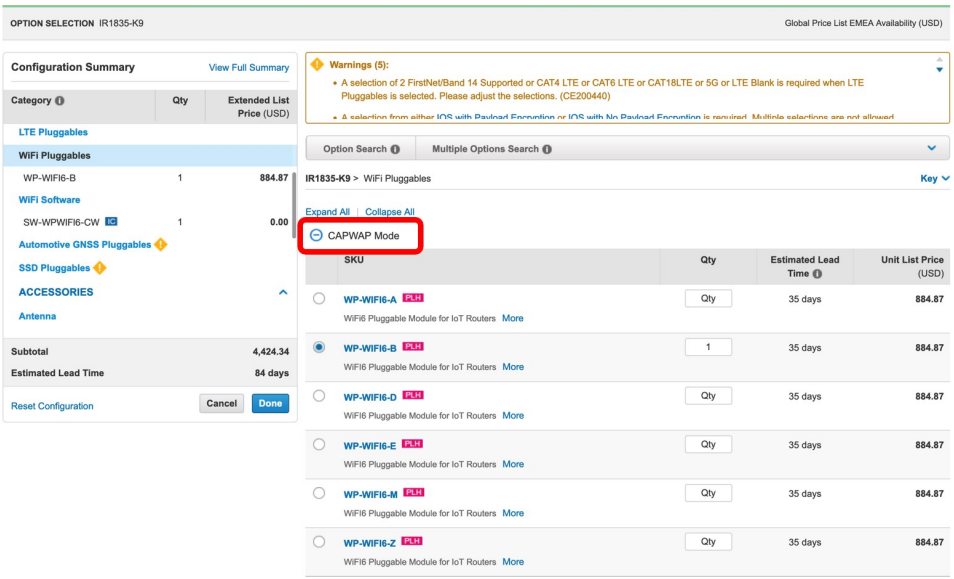
Check the association.

In the following example, the Client must be attached to see uWGB status. Otherwise, it falls back to WGB.

```
#show wgb dot11 associations
Uplink Radio ID             : 1
Uplink Radio MAC            : BC:E7:12:0C:F1:CF
SSID Name                   : Free
Parent AP MAC               : 08:02:8E:8D:52:9A
Uplink State                : CONNECTED
Auth Type                   : PSK
Key management Type         : WPA2
Uclient mac                 : C4:4D:84:9B:0A:8C
Current state               : UWGB
Uclient timeout             : 60 Sec
Dot11 type                  : 11ac
Channel                     : 36
Bandwidth                   : 80 MHz
Current Datarate            : 433 Mbps
Max Datarate                : 1200 Mbps
RSSI                        : 53
IP                          : 0.0.0.0
IPV6                        : ::/128
Assoc timeout               : 5000 Msec
Auth timeout                : 5000 Msec
Dhcp timeout                : 60 Sec
```

Wi-Fi Mode Conversion

When ordering the Wi-Fi pluggable module from Cisco, only the CAPWAP Mode is currently available from Cisco Commerce Workspace (CCW) today. If you wish to use EWC or WGB mode, you will have to convert the mode. A new SKU will be created for EWC mode in the future.



Converting From CAPWAP to EWC Mode

This conversion is required when you have an AP with a CAPWAP image, and you want to use the AP to deploy a embedded wireless controller based network. To do this, you must convert the CAPWAP AP to an embedded wireless controller.

To convert an 802.11ax AP with a CAPWAP image to an embedded wireless controller capable image, either download the controller image based on the automated image download process, use the conversion command, or convert through the WebUI.

Perform the following steps:

1. First check that the AP has the correct image for conversion.

```
APBCE7.120C.DAD8# show version | include AP
Cisco AP Software, (ap1g8), [cheetah-build21:/san1/BUILD/workspace/c176_throttle-ax-bcm32]
APBCE7.120C.DAD8 uptime is 0 days, 0 hours, 5 minutes
Last reload reason : AP type changed from ME to CAPWAP
AP Running Image     : 17.6.2.20
AP Image type     : EWC-AP IMAGE
AP Configuration : NOT ME OR EWC-AP CAPABLE
APBCE7.120C.DAD8#
```

Note If the images are not there, copy them onto the IR1800 bootflash: or on a TFTP server.

2. Use the following command to start the conversion process.

   # **ap-type ewc-ap tftp: tftp:** if the image is available on the module.

   AP-console#**ap-type ewc-ap tftp:**//<*ip-address*>/**ap1g8 tftp:**<*ip-address*>/<*image-name*> if the image is on a TFTP server.

```
AP-console#ap-type ewc-ap tftp://192.168.72.11/ap1g8
tftp://192.168.72.11/C9800-AP-iosxe-wlc.bin
Starting download eWLC image tftp://192.168.72.11/C9800-AP-iosxe-wlc.bin
It may take a few minutes. If longer, please abort command, check network and try again.

##################################################################### 100.0%
```

3. The AP will restart and now the configuration for the new mode must be performed. Refer to the Embedded Wireless Controller.

To use the IR1800 as a TFTP server:

1. Copy the AP files onto the IR1800 bootflash:
2. Configure a TFTP server on IR1800, i.e. AP attached on VLAN1 interface.

```
IR1800# config term
ip tftp source-interface Vlan1
tftp-server bootflash:C9800-AP-iosxe-wlc.bin
tftp-server bootflash:ap1g8
```

Converting From CAPWAP to WGB Mode

In workgroup bridge (WGB) mode, the unit associates to another access point as a client and provides a network connection for the devices connected to its Ethernet port. For example, if you need to provide wireless connectivity for a group of network printers, you can connect the printers to a hub or to a switch, connect the hub or switch to the access point Ethernet port, and configure the access point as a workgroup bridge. The workgroup bridge associates to an access point on your network.

Perform the following steps:

1. Convert the CAPWAP AP to WGB mode.

```
APBCE7.120C.DAA8#ap-type workgroup-bridge
WGB is a wireless client that serve as nonroot ap for wired clients.
AP is the Master/CAPWAP AP, system will need a reboot when ap type is changed to
 WGB. Do you want to proceed? (y/N): y
```

2. The AP will restart and now the configuration for the new mode must be performed. Refer to the Workgroup Bridge (WGB).

Converting From EWC to CAPWAP Mode

This conversion is required if you want to migrate the APs from an embedded wireless controller network to a non-embedded wireless controller network; or if you do not want the APs to participate in the primary AP election process.

To convert an 802.11ax AP from the embedded wireless controller network to a non-embedded wireless controller network, set the AP type to CAPWAP using the conversion command or the WebUI, respectively, and then plug it on to the controller network so that it joins the controller. If the image on the controller is different from the image on the AP, a new CAPWAP image is requested from the controller.

Perform the following steps:

1. Get the name of the AP to convert.

```
IR1831-EWC#show ap sum
Number of APs: 1
AP Name              Slots  AP Model     Ethernet MAC    Radio MAC       Location
       Country  IP Address      State
-------------------------------------------------------------------------------------
APBCE7.120C.DAA8     2      WP-WIFI6-B    bce7.120c.daa8  bce7.120c.f1c0  default
location    US       192.168.72.2    Registered
```

2. Use the following command to start the conversion process.

   #**wireless ewc-ap ap ap-type** *ap-name* **capwap**

```
IR1831-EWC#wireless ewc-ap ap ap-type APBCE7.120C.DAA8 capwap
```

3. The AP will restart and now the configuration for the new mode must be performed. Refer to the Control And Provisioning of Wireless Access Points (CAPWAP) .

Upgrading the Firmware on the Cisco Wi-Fi Interface Module

The firmware can be upgraded from the access point command line interface or the WebUI while in EWC mode.

Firmware Upgrade Using The AP CLI

Perform the following steps to upgrade the firmware:

1. Copy the AP files to the IR1800 bootflash:
2. Configure the IR1800 to act as TFTP or SFTP server.

   TFTP Server:

   ```
   ip tftp source-interface Vlan1   ! VLAN interface to be modified based on the
   configuration
   tftp-server bootflash:C9800-AP-iosxe-wlc.bin
   tftp-server bootflash:ap1g8
   ```

   SFTP Server:

   ```
   ip sftp source-interface Vlan1   ! VLAN interface to be modified based on the
   configuration
   sftp-server bootflash:C9800-AP-iosxe-wlc.bin
   sftp-server bootflash:ap1g8
   ```

3. Upgrade the AP firmware from AP CLI.

   TFTP Server:

   ```
   #ap-type ewc-ap tftp://10.10.10.1/ap1g8 tftp://10.10.10.1/C9800-AP-iosxe-wlc.bin
   ```

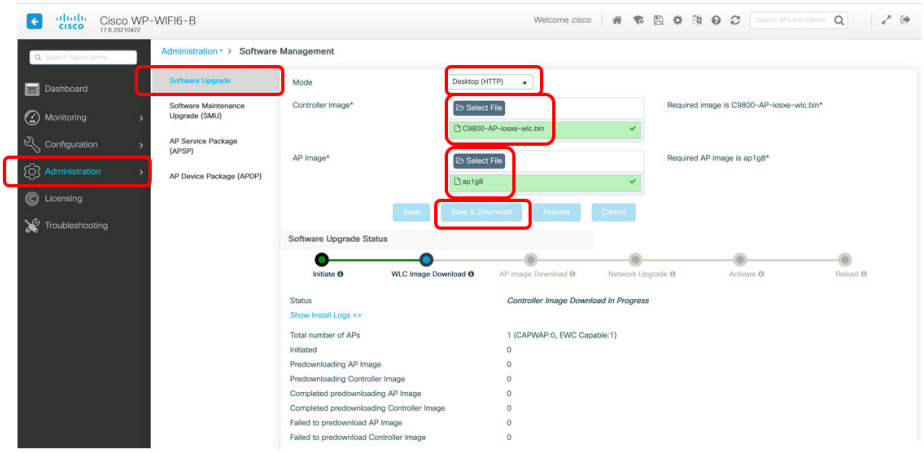
   SFTP Server:

   ```
   #ap-type ewc-ap sftp://10.10.10.1/ap1g8 sftp://10.10.10.1/C9800-AP-iosxe-wlc.bin
   ```

Firmware Upgrade Using The EWC WebUI

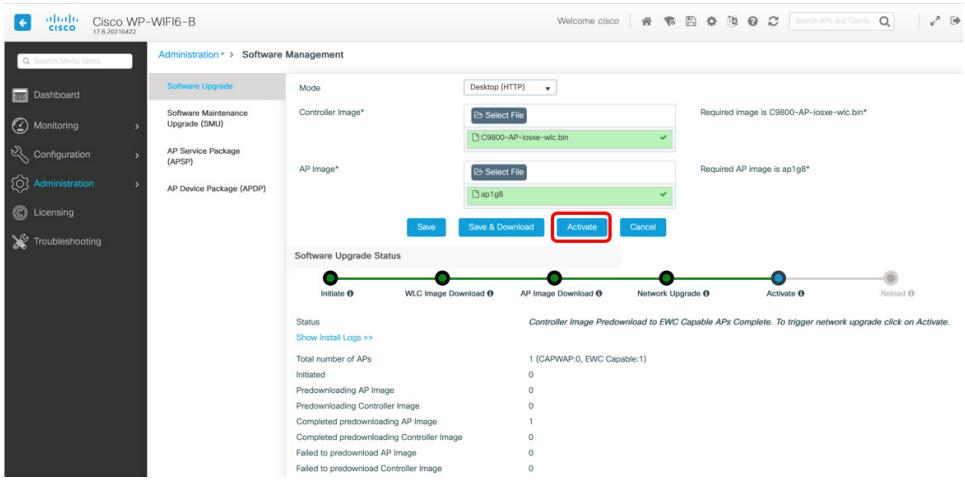
Perform the following steps to upgrade the firmware:

1. Open the EWC WebUI and navigate to **Administration > Software Management > Software Upgrade**.

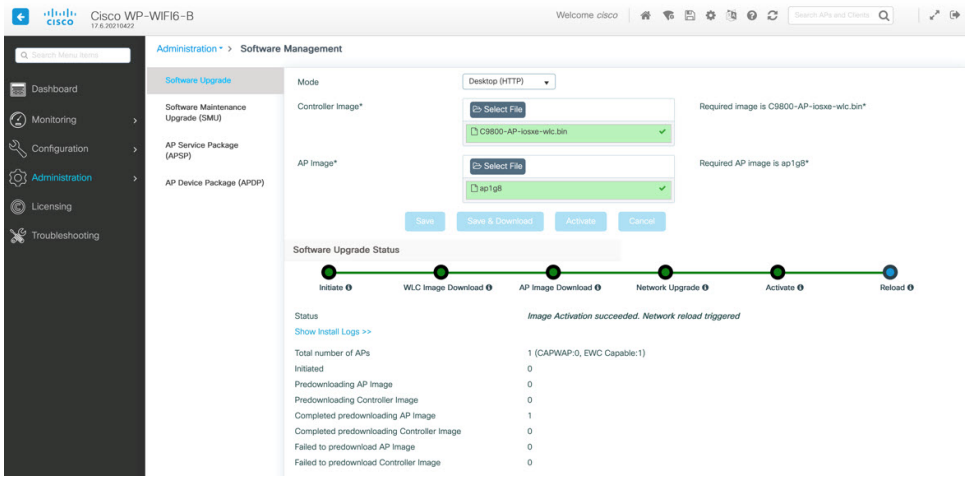


2. Select the following as referenced in the image above:
   - Mode: Desktop (HTTP)
   - Controller Image: C9800-AP-iosxe-wlc.bin
   - AP Image: ap1g8

3. Click on **Save & Download**.
4. Click on **Activate**.



5. The image activates successfully and then the Cisco Wi-Fi Interface Module reloads.



Connection to the Ethernet Switch

The WIM GE driver is directly connected to the host router GE driver. The interface name of the WIM GE port on the router is Wlan-GigabitEthernet0/1/4.

Host Router Configuration

Connection summary:

- Wlan-GigabitEthernet0/1/4 is a layer 2 interface
- The interface is a switchport which supports both access mode and trunk mode
- Traffic forwarding is performed by the ethernet switch

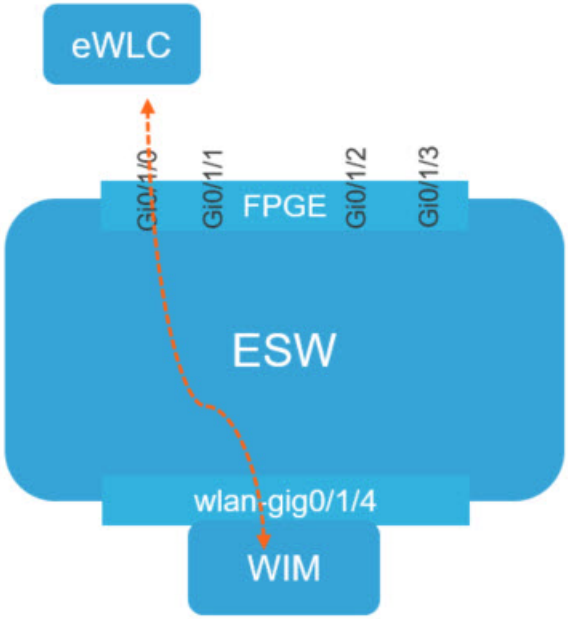
- layer 2 forwarding from/to Front Panel GigabitEthernet
- layer 3 forwarding, handled by the host CPU (routing, NAT, tunneling, and others)

Configuration message:

```
Wlan-GigabitEthernet0/1/4 is up, line protocol is up (connected)
  Hardware is WLAN Gigabit Ethernet, address is c014.fe60.bc8c (bia c014.fe60.bc8c)
  Description: Internal switch interface connecting to the embedded AP
  MTU 1500 bytes, BW 1000000 Kbit/sec, DLY 10 usec,
     reliability 255/255, txload 1/255, rxload 1/255
  Encapsulation ARPA, loopback not set
  Keepalive not supported
  Full-duplex, 1000Mb/s, link type is auto, media type is 10/100/1000BaseTX
  input flow-control is off, output flow-control is unsupported
  ARP type: ARPA, ARP Timeout 04:00:00
  Last input 00:00:14, output 00:00:04, output hang never
  Last clearing of "show interface" counters never
  Input queue: 0/375/0/0 (size/max/drops/flushes); Total output drops: 0
  Queueing strategy: fifo
  Output queue: 0/40 (size/max)
  30 second input rate 0 bits/sec, 0 packets/sec
  30 second output rate 0 bits/sec, 0 packets/sec
     101750 packets input, 19423004 bytes, 0 no buffer
     Received 9550 broadcasts (48415 multicasts)
     0 runts, 0 giants, 0 throttles
     0 input errors, 0 CRC, 0 frame, 0 overrun, 0 ignored
     0 watchdog, 48415 multicast, 0 pause input
     0 input packets with dribble condition detected
     98541 packets output, 11782026 bytes, 0 underruns
     Output 8714 broadcasts (70879 multicasts)
     0 output errors, 0 collisions, 1 interface resets
     5283 unknown protocol drops
     0 babbles, 0 late collision, 0 deferred
     0 lost carrier, 0 no carrier, 0 pause output
     0 output buffer failures, 0 output buffers swapped out
```

Access Mode

When all the control and data traffic are encapsulated in a CAPWAP tunnel, the user can configure wlan0/1/4 in access mode. See the following illustration:

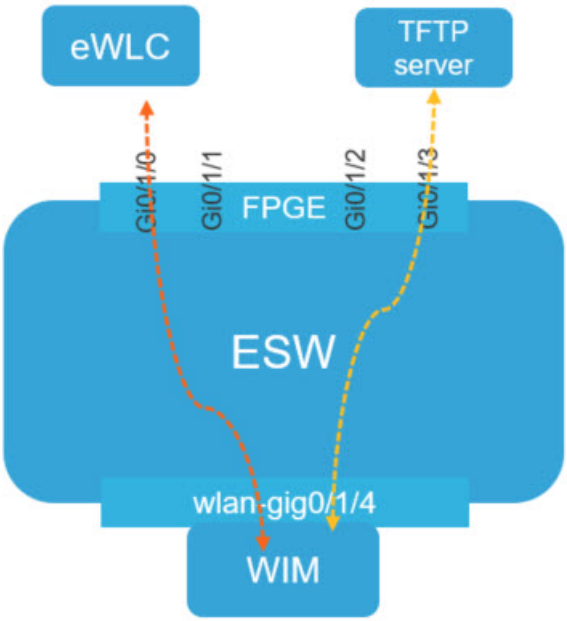


Configuration Example

```
interface GigabitEthernet0/1/0
 switchport access vlan 10
 switchport mode access
interface Wlan-GigabitEthernet0/1/4
 switchport access vlan 10
 switchport mode access
```

Trunk Mode

When all the control and data traffic are encapsulated in a CAPWAP tunnel, the user can configure wlan0/1/4 in trunk mode. See the following illustration:

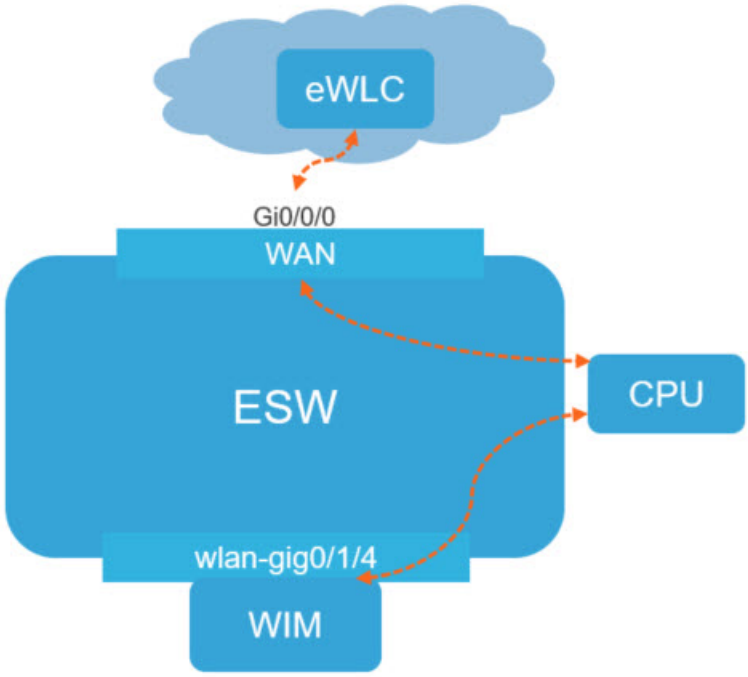


Configuration Example

```
interface GigabitEthernet0/1/0
 description WirelessTraffic
 switchport trunk native vlan 10
 switchport trunk allowed vlan 10,20
 switchport mode trunk
interface GigabitEthernet0/1/3
 description InternalServer
 switchport access vlan 30
 switchport mode access
interface Wlan-GigabitEthernet0/1/4
 switchport trunk native vlan 10
 switchport trunk allowed vlan 10,20,30
 switchport mode trunk
```

Wi-Fi Traffic Behind a NAT

In this configuration, all WLAN traffic is behind a NAT and/or Firewall. See the following illustration:



Configuration Example

```
interface GigabitEthernet0/0/0
 ip address 192.168.4.50 255.255.255.0
 ip nat outside
interface Wlan-GigabitEthernet0/1/4
 switchport access vlan 10
 switchport mode access
interface vlan10
 ip address 10.10.10.1 255.255.255.0
 ip nat inside
ip dhcp pool vlan10
 network 10.10.10.0 255.255.255.0
 default-router 10.10.10.1
 option 43 hex f104.c0a8.04fc
ip access-list extended 100
 1 permit ip 10.10.10.0 0.0.0.255 any
ip nat inside source list 100 interface GigabitEthernet0/0/0 overload
```

```
Router#show ip nat translations
Pro  Inside global         Inside local          Outside local         Outside global
icmp 192.168.4.50:10022    10.10.10.2:10022      192.168.4.252:10022   192.168.4.252:10022
udp  192.168.4.50:5264     10.10.10.2:5264       192.168.4.252:5246    192.168.4.252:5246
Total number of translations: 2
```

Factory Reset

The WIM will follow the host router when performing a factory reset. A factory reset is triggered by holding the Reset button on the router for 4 seconds or longer. If you perform a factory reset on the router, the WIM will also be factory reset.

In most cases, the user will only want to reset the WIM to its factory default condition. (for example, the user forgot the credentials to login to AP console).

The user can execute the following command from the host router:

```
Router# hw-module subslot 0/3 error-recovery password_reset
```

The above command sets the WIM to maintenance mode.

Note When you run the **hw-module subslot 0/3 error-recovery password_reset** command, the AP module automatically reloads to restore the configuration settings and enters the maintenance mode. In the maintenance mode, the AP module is on power on mode. Confirm the module configuration reset through the console or web UI. The user will need to issue the **hw-module subslot 0/3 reload force** command to reload the AP and take it out of maintenance mode.

Factory Reset Example

The following sequence shows the factory reset:

```
Router# hw-module subslot 0/3 error-recovery password_reset
```

The WIM reloads.

The WIM is set to maintenance mode and shows out of service.

```
Router# show platform
Chassis type; IR1835-K9
Slot    Type            State            Insert time (ago)
0       IR1835-K9       ok               00:54:57
 0/0    IR1835-1K1GE    ok               00:52:49
 0/1    IR1835-ES-4     ok               00:52:46
 0/3    WP-WIFI6-B      out of service   00:34:24
R0      IR1835-K9       ok, active       00:54:57
FO      IR1835-K9       ok, active       00:54:57
PO      P-R-12V         ok               00:53:09
GE-P06  Unknown         ok               00:53:09
```

On the WIM, u-boot detects RESET signal. Console prints the following

```
Button is pressed. Configuration reset activated..
Keep the button pressed for > 20 seconds for full reset
Wait for the button to be released ....
Button pressed for 11 seconds
Setting env for full reset..
Erasing SPI flash...Writing to SPI flash...done
```

The user should wait approximately 30 seconds, then use the following command:

```
Router# hw-module subslot 0/3 reload force
```

The WIM reloads.

The WIM quits maintenance mode. Wait for the WIM to turn to the ok state.

```
Router#show platform
Chassis type: IR1835-K9
Slot    Type             State  Insert time (ago)
0       1R1835-K9        ok     00:56:50
 0/0    IR1835-1X1GE     ok     00:54:42
 0/1    IR1835-ES-4      ok     00:54:39
 0/3    KP-WIFI6-B       ok     00:01:36
```

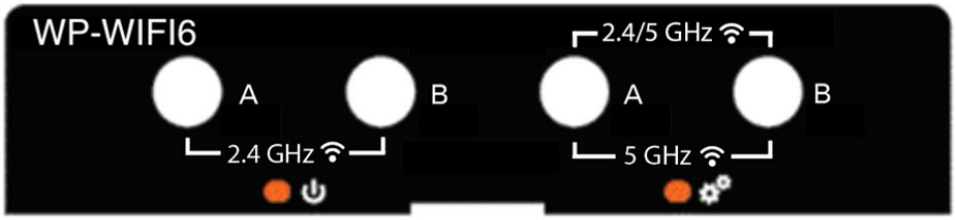
Flexible Antenna Port

The flexible antenna port feature allows customers to customize their radio coverage. The user can configure the feature as dual band mode or single band mode via CLI commands or the WebUI. The default is dual band mode.

In some other IoT products, ports A&B are used for 2.4G single band and C&D are used for 5G single band. The WIM uses a different naming scheme.

For the antenna port labeling, the WIM uses A&B to label both the 2.4G and the 5G posts.

Looking at the front panel of the WIM:



In Dual Band mode, install antennae to A/B ports on the right side.

In Single Band mode, install 2.4G band antennae to A/B ports on the left side, and install 5G band antennae to A/B ports on the right side.

Switching Bands

The CLI for switching band mode on the controller is:

ap name <*Cisco AP*> **antenna band-mode** *[single]|[dual]*

For example:

```
ap name ape8eb-349c-14c0 antenna-band-mode ?
  dual    Dual band mode
  single  Single band mode
```

On the WIM, the band mode is shown by the GPIO value. See the following table:

Antenna Band Mode	Switch Direction	GPIO_34 = ANT_VCONT1	GPIO_35 = ANT_VCONT2
Dual Band (default)	RF1-RFC	1	0
Single Band	RF2-RFC	0	1

Verify the Band Mode

From the wireless controller:

```
show ap name ape8eb-349c-14c0 config general | inc Antenna Band
Antenna Band Mode: Single
```

From the WIM:

```
Access Point#show capwap client config | inc GPIO
GPIO_34: 0
GPIO_35: 1
```

Note GPIO_34:1 & GPIO_35:0 is dual band mode. GPIO_34:0 & GPIO_35:1 is single band mode.

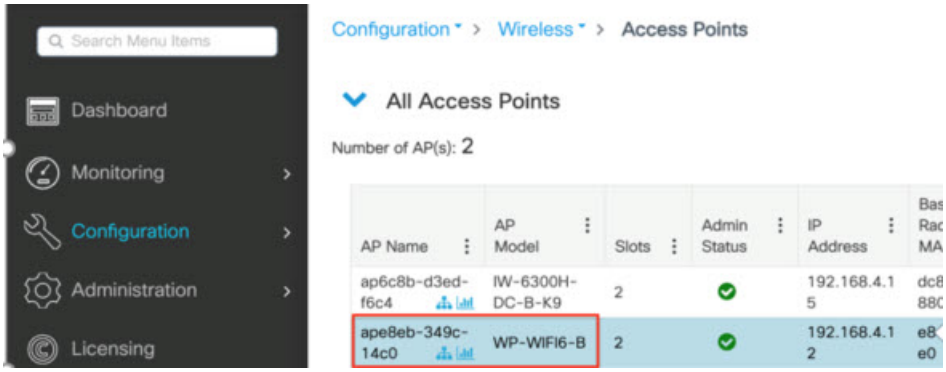
Configuration

Configure from the wireless controller:

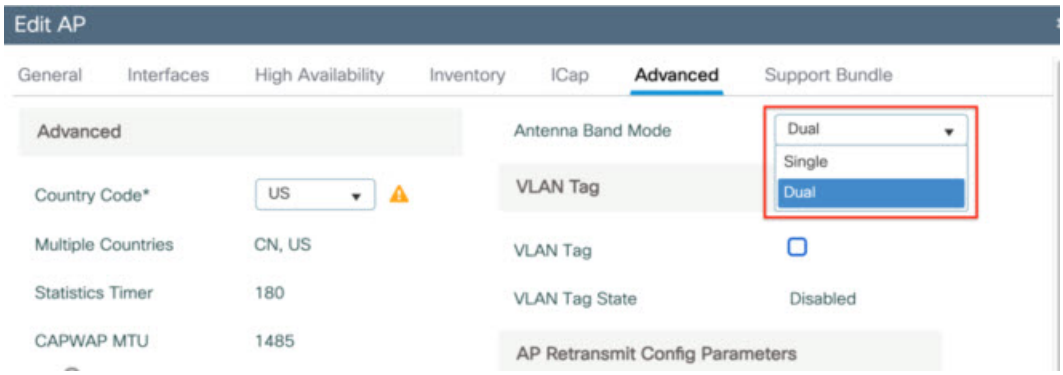
```
ap name ape8eb-349c-14c0 antenna-band-mode ?
  dual    Dual band mode
  single  Single band mode
```

Configure from the WEB UI:

Select the Access Point:



Configure the Access Point:



Verify the Configuration

From the controller:

```
#show ap name ape8eb-349c-14c0 config general | inc Antenna Band
Antenna Band Mode: Single
```

From the WIM:

```
#ape8eb-349c-14c0#show capwap client config | inc GPIO
GPIO_34: 0
GPIO_35: 1
```

Related Documentation

There are many different options that can be configured on the Access Point depending on your installation scenario. Other sources of documentation are available here:

Cisco Catalyst 9100 Family of Access Points

Cisco Wireless Controller Configuration Guide

Cisco Embedded Wireless Controller on Catalyst Access Points FAQ

Cisco Catalyst 9800 Series Configuration Best Practices

Cisco Wave 2 Access Points as Workgroup Bridges

Cisco Industrial Wireless Workgroup Bridge and Universal WGB Deployment Guide

CHAPTER 20

Digital I/O, Ignition, and CAN Bus Connectivity

This section contains the following:

- Overview, on page 187
- Digital IO, on page 187
- Controller Area Network (CAN) Bus, on page 188
- IOx CAN Bus Support, on page 189
- Configuring Digital IO, on page 189
- Ignition Power Management Overview, on page 190
- Features of Ignition Power Management, on page 190
- Ignition Sense Overview, on page 191
- IR1835 Ignition Switch, on page 192
- IR1800 Ignition and Battery Voltage, on page 194
- Command Line Interface (CLI), on page 194
- Default Values, on page 196
- Ignition Power Management Yang Model, on page 196
- Support SNMP MIB for Ignition Power Management , on page 197

Overview

This section covers the Digital I/O configuration and the CAN Bus details.

The IR1835 supports four General-Purpose Input/Output ports (GPIO), also referred to as digital I/O ports. A GPIO port can be configured as input OR output alarm. It can work as dry or wet contact, protected up to +60V .

The Controller Area Network (CAN) Bus enables the ECU (electronic control unit) in a vehicle to communicate with all other ECUs. It consists of two wires, CAN bus High and Low supporting data rate up to 1 Mbs.

The characteristics of the high speed CAN Bus 2.0B are ISO 11898-1 data link layer, ISO 11898-2 and ISO-11898-5 physical layer up to 1Mbs data rate (SW dependent).

Digital IO

A total of four Digital IO with a common return are supported. Digital IO is similar to the ALARM IN and ALARM OUT supported in the IE switches and IR routers. The differences are the ALARM IN is a dedicated

input, the ALARM OUT is a dedicated output whereas the Digital IO can be input or output. ALARM OUT includes a relay to provide the Normally Open (NO) or Normally Close (NC) terminals. Digital IO implements a relay feature similar to Alarm port.

The following configuration commands are available:

```
alarm contact attach-to-iox
alarm contact <1-4> enable enable
alarm contact <1-4> application
alarm contact <1-4> description
alarm contact <1-4> severity
alarm contact <1-4> threshold <1600-2700>
alarm contact <1-4> trigger
alarm contact <1-4> output <1 | 0>
alarm contact <1-4> output relay temperature
alarm contact <1-4> output relay input-alarm <0-4>
```

All configuration commands also come with a **no** prefix to them.

Controller Area Network (CAN) Bus

Details on the CAN Bus and connectivity to the vehicle's On-Board Diagnostic (OBD-II) are covered in the Cisco Catalyst IR1800 Rugged Series Router Hardware Installation Guide.

The CAN Bus interface can be viewed using the command line interface. Some of the CLIs are:

```
IR1800#conf term
Enter configuration commands, one per line.  End with CNTL/Z.
IR1800(config)#canbus baudrate ?
  <125000-1000000>  enter baud rate ranging from 125000 to 1000000

IR1800#show platform hardware canbus ?
  interface  Display CAN Bus interface
  link       Display CAN Bus link

IR1800#show platform hardware canbus link
8: can0: <NOARP,UP,LOWER_UP,ECHO> mtu 16 qdisc pfifo_fast state UP mode DEFAULT group default
 qlen 10
    link/can  promiscuity 0
    can state ERROR-ACTIVE restart-ms 100
          bitrate 125000 sample-point 0.875
          tq 500 prop-seg 6 phase-seg1 7 phase-seg2 2 sjw 1
          mcp251x: tseg1 3..16 tseg2 2..8 sjw 1..4 brp 1..64 brp-inc 1
          clock 10000000
          re-started bus-errors arbit-lost error-warn error-pass bus-off
          0          0          0          0          0          0

numtxqueues 1 numrxqueues 1 gso_max_size 65536 gso_max_segs 65535

    RX: bytes  packets  errors  dropped overrun mcast
    0          0        0       0       0       0
    TX: bytes  packets  errors  dropped carrier collsns
    0          0        0       0       0       0

IR1800#

IR1800#show platform hardware canbus interface
can0      Link encap:UNSPEC  HWaddr 00-00-00-00-00-00-00-00-00-00-00-00-00-00-00-00
          UP RUNNING NOARP  MTU:16  Metric:1
          RX packets:0 errors:0 dropped:0 overruns:0 frame:0
          TX packets:0 errors:0 dropped:0 overruns:0 carrier:0
```

```
          collisions:0 txqueuelen:10
          RX bytes:0 (0.0 B)  TX bytes:0 (0.0 B)
```

IOx CAN Bus Support

A vxcan tunnel is created between Linux and IOx, and then the CAN_GW inside the linux is configured to forward CANBus traffic from/to the real can bus interface (i.e. can0) to/from the end point of the vxcan tunnel.

A vxcan tunnel is established by default, the CANBus traffic will be sent over the tunnel as follows:

CANBus: [Linux] vxcan-ap and vxcan0 [IOx]

Sample output from vxcan0 [IOx]

```
vxcan0    Link encap:UNSPEC  HWaddr 00-00-00-00-00-00-00-00-00-00-00-00-00-00-00-00
          UP RUNNING NOARP  MTU:16  Metric:1
          RX packets:715 errors:0 dropped:0 overruns:0 frame:0
          TX packets:530 errors:0 dropped:0 overruns:0 carrier:0
```

Important Notes

- The feature can support multiple containers, one pair of vxcan will be setup for one container, therefore, two pairs of vxcan will be setup for two containers.
- Within one container, if multiple applications need to access vxcan, it is customer's application responsibility to include CAN_GW in their software.

Configuring Digital IO

To configure the feature, perform the following:

```
Router(config)# alarm contact 2 enable
Router(config)#default alarm contact 2 output
Router(config)#default alarm contact 2 severity
Router(config)#default alarm contact 2 threshold
Router(config)#default alarm contact 2 trigger
Router(config)#default alarm contact 2 application
Router(config)#default alarm contact 2 description
Router(config)#end
```

To view alarm output, perform the following:

```
Router# show alarm | section Digital I/O 2
Digital I/O 2:
Description: External digital I/O port 2
Status: Not Asserted
Application: Dry
Severity: minor
Trigger: Closed
Voltage: 3300mV
Threshold: 1600mV
Mode: Input
Router#
```

Ignition Power Management Overview

This section provides a description and instructions for configuration of the Ignition Power Management feature of the IR1800 router. Ignition Power Management prevents the router from draining the charge of the battery on automotive applications. It also keeps the router up and running while the vehicle is stopped. Therefore, users do not have to wait for routers to reload each time the vehicle is stopped.

When the engine is running, it generates energy and recharges the battery. When the ignition is turned off, the router can remain operational for a pre-determined period of time.

On the IR1800 series, there are two ways to perform Ignition Power Management. All of the routers in the series can use the software based voltage sense controlled by the MCU. The IR1835 can also use the Signal/Ignition signal, available from the Ignition pin in the GPIO 6 pins connector.

Ignition Wiring information is found in the Hardware Installation Guide.

Features of Ignition Power Management

Ignition Power Management is controlled through the MCU built into the router. The MCU provides auto detection of power input to detect if the ignition is on or off. Ignition on is detected by if the ignition signal is on or off, or by sensing the power input level.

Ignition Power Management Cabling and Connector is covered here: https://www.cisco.com/c/en/us/td/docs/routers/access/IR1800/hig/b-ir1800-hig/m-GPIO.html

The system software tries to prevent the discharge of the battery with the following:

- Turning the router off if the vehicle has the ignition off for a period of time (programmable).
- Turning the router off if the battery voltage drops to a certain level (programmable).
- Attempting to protect the router by turning the router off if the battery voltage rises above a certain level (fixed amount of time).

The system software logs the following events to the system log:

- When the user turns on or off the ignition management feature with CLI
- When the ignition is turned on or off
- When the ignition-off timer expires and the system goes off
- When the user enables or disables the feature through the CLI
- Tentatively logs the under-voltage and over-voltage events

All Ignition On, Off, Low and High input thresholds can be stored in non-volatile memory. When the device boots up, the thresholds will be retrieved from the memory to the register. When the CPU detects the front panel push button, the non-volatile memory will reset to default value.

Ignition Sense Overview

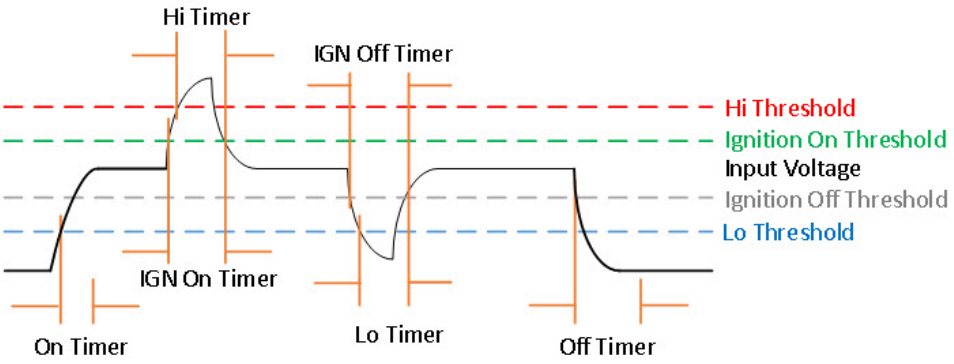
On the IR1800, Ignition Power Management adds ignition sense through software. Ignition sense can be determined by either monitoring the ignition signal pin, or the battery voltage level. The ignition pin and voltage level are continuously monitored. This input will be the main signal to start the state machine. If the ignition signal is not active, then voltage level sense will be used. There is an option for user to disable the voltage level sense by clearing the Ignition Voltage Sense Enable register.

See the following table:

Ignition Bypass	Ignition Voltage Sense Enable	Ignition Sense
0	0	Ignition Signal
0	1	Voltage Level
1	x	Ignition Disabled

The following graphic illustrates Ignition Sense:

Figure 38: Ignition Based on Voltage (Analog Input)



The following tables provide voltage details:

Table 15: Input Voltage (DC)

Minimum	9.6V
Maximum	36V
Nominal	12V or 24V

Table 16: Ignition Sense Voltage

	12V Battery	24V Battery
On	13V + 2%	26V + 2%
Off	13V -2%	26V - 2%

Table 17: Battery Voltage

	12V Battery	24V Battery
Under-Voltage	11.5V	23V
Over-Voltage	36V	36V

See the following command output example:

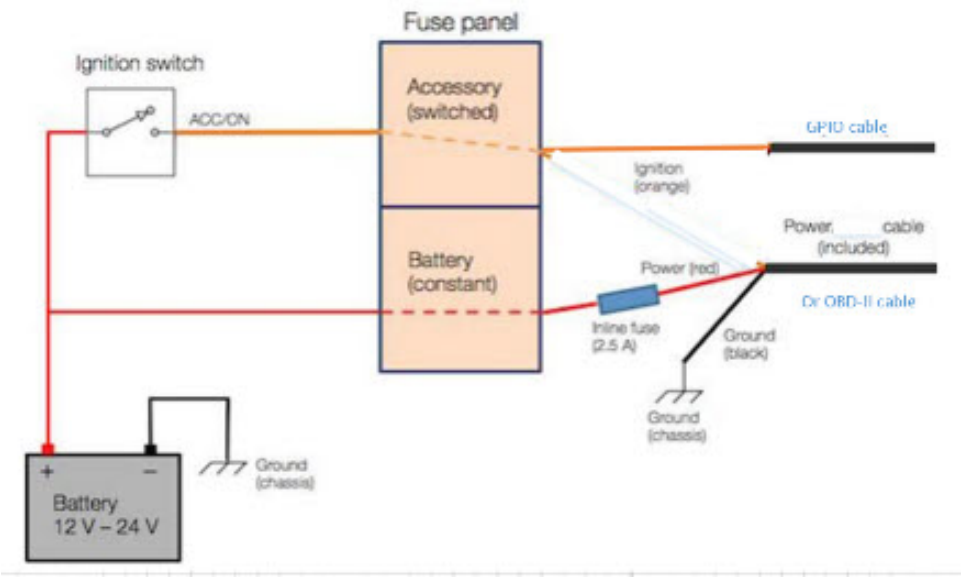
```
IR1800#show run | s ignition
ignition off-timer 300
ignition undervoltage threshold 9 000
ignition battery-type 12v
ignition sense-voltage threshold 13 000
ignition sense
ignition enable

IR1800#show ignition status
Status:
  Ignition management: Enabled
  Input voltage:       11.953 V
  Ignition status:     Timing ignition off shut down
  Ignition Sense:      Enabled
  Shutdown timer:      242.0 s to off [will begin power down at ~100 sec]
  Config-ed battery:   12v
Thresholds:
  Undervoltage:        9.000 V
  Overvoltage:         37.000 V
  Sense on:            13.000 V
  Sense off:           12.800 V
  Undervoltage timer:  20.0 s
  Overvoltage timer:   1.0 s
  Ignition-Off timer:  300.0 s
```

IR1835 Ignition Switch

The IR1835 is the only model in the IR1800 series that offers ignition monitoring via the Signal/Ignition signal, available from the Ignition pin in the GPIO 6 pins connector.

Refer to the following figure:

Figure 39: Ignition based on Signal/Ignition switch(Digital Input)



There are two ways to power the router:

- CANBUS power
- Through the CAB-PWR-15-MF4 cable

There are two thresholds:

- 12V — Defaults ignition sense-voltage threshold 13
- 24V — Defaults ignition sense-voltage threshold 26

See the following command output example:

```
IR1835#show run | s ignition ! Configuring
ignition off-timer 120
ignition undervoltage threshold 9 600
ignition battery-type 12v
ignition sense-voltage threshold 13 000
no ignition sense
ignition enable

IR1835#show ignition Status ! Monitoring
 Ignition management: Enabled
Input voltage:       13.999 V
Ignition status:     Power on
Ignition Sense:      Disabled
 Shutdown timer:      0.0 s to off [will begin power down at ~100 sec]
Config-ed battery:   12v
Thresholds:
Undervoltage:        9.600 V
Overvoltage:         37.000 V
Sense on:            13.200 V
Sense off:           12.800 V
Undervoltage timer:  20.0 s
Overvoltage timer:   1.0 s
Ignition-Off timer:  120.0 s
```

IR1800 Ignition and Battery Voltage

The IR1800 can be set-up with a 12V or 24V battery. The ignition sense voltage level will change accordingly.

- 12Volt Battery
  - If input voltage > 13.200V for more than 1 second, Ignition is **ON**
  - If input voltage < 12.800V for more than 20 seconds, Ignition is **OFF**
- 24Volt Battery
  - If input voltage > 26.200V for more than 1 second, Ignition is **ON**
  - If input voltage < 25.800V for more than 20 seconds, Ignition is **OFF**

Use the folowing command to determine your battery type:

```
IR1800#conf term
Enter configuration commands, one per line.  End with CNTL/Z.
IR1800(config)#ignition battery-type ?
  12v  The battery is 12v
  24v  The battery is 24v
```

Command Line Interface (CLI)

The Ignition Power Management feature of the IR1800 series uses a command line interface.

Enabling Ignition Power Management

The feature is disabled by default, and enabled using the following command:

```
Router(config)#ignition enable
Router(config)#
*Sep 15 16:08:27.697: %IGNITION-5-IGN_ENABLE_CMD: The Ignition Power Management is enabled
Router(config)#end
```

Options to the ignition enable command are:

```
Router(config)#ignition enable ?
  enable        Enable ignition power management feature
  off-timer     Off timer delay
  sense         Enable ignition power sense feature
  undervoltage  Set undervoltage parameters for shutting the system off
```

- Ignition off timer value. After the ignition is turned off the router will stay operational for this amount of time, then it shuts down if the ignition is still off:

```
Router#ignition off-timer <value>
```

- Ignition sense value. Turning this on allows the router to detect if the the ignition is on or off.

```
Router(config)#ignition sense
*Sep 15 16:08:14.391: %IGNITION-5-IGN_SENSE_CMD: The Ignition Voltage Sense is enabled
Router(config)#end
```

- Ignition undervoltage. This command allows you to set parameters for shutting down the router.

```
Router(config)#ignition undervoltage threshold ?
<0-999>  Enter millivolts (mV), if any
```

- Over-voltage threshold. This command allows you to set parameters for shutting down the router.

```
Router#ignition overvoltage threshold <value>
```

Command Examples

Default configuration with no Ignition Management settings:

```
Router#show ignition
Status:
Ignition management: Disabled
Input voltage: 17.672 V
Ignition status: Power on
Ignition Sense: Disabled
Shutdown timer: 0.0 s to off [will begin power down at ~100 sec]
Config-ed battery: 12v
Thresholds:
Undervoltage: 9.000 V
Overvoltage: 37.000 V
Sense on: 13.200 V
Sense off: 12.800 V
Undervoltage timer: 20.0 s
Overvoltage timer: 1.0 s
Ignition-Off timer: 300.0 s
```

Configure Ignition Management:

```
Router(config)#ignition enable
Router(config)#
*Sep 15 16:08:27.697: %IGNITION-5-IGN_ENABLE_CMD: The Ignition Power Management is enabled

Router(config)#ignition sense
*Sep 15 16:08:14.391: %IGNITION-5-IGN_SENSE_CMD: The Ignition Voltage Sense is enabled
Router(config)#end
```

Verify the changes:

```
Router#show ignition
Status:
Ignition management: Enabled
Input voltage: 17.656 V
Ignition status: Power on
Ignition Sense: Enabled
Shutdown timer: 0.0 s to off [will begin power down at ~100 sec]
Config-ed battery: 12v
Thresholds:
Undervoltage: 9.000 V
Overvoltage: 37.000 V
Sense on: 13.200 V
Sense off: 12.800 V
Undervoltage timer: 20.0 s
Overvoltage timer: 1.0 s
Ignition-Off timer: 300.0 s
Router#
```

Show Ignition Status

The following commands are used to show the status of the feature:

```
Router#show ignition
Status:
```

```
Ignition management: Disabled
Input voltage: 17.672 V
Ignition status: Power on
Ignition Sense: Disabled
Shutdown timer: 0.0 s to off [will begin power down at ~100 sec]
Config-ed battery: 12v
Thresholds:
Undervoltage: 9.000 V
Overvoltage: 37.000 V
Sense on: 13.200 V
Sense off: 12.800 V
Undervoltage timer: 20.0 s
Overvoltage timer: 1.0 s
Ignition-Off timer: 300.0 s

Router#show running-config | sec ignition
ignition off-timer 300
ignition undervoltage threshold 9 000
ignition battery-type 12v
ignition sense-voltage threshold 13 000
no ignition sense
no ignition enable
```

Default Values

The following default settings apply to Ignition Power Management:

Setting	Default Value
Ignition Power Management Feature	Disabled
Ignition Sense	Disabled
Off Timer	300 seconds
Under Voltage Threshold	9.000 Volts
Under Voltage Off Timer	20 seconds
Over Voltage Off Timer	1.0 seconds
Ignition Sense On	13.200 volts (26.200 volts)
Ignition Sense Off	12.800 volts (25.800 volts)
Configured Battery	12 volts (24 volts)

Ignition Power Management Yang Model

A Yang Model is available for the Ignition Power Management Configuration Model (config-model) and Ignition Power Management Show Command (oper-model).

The ignition configuration CLI's for the config-model are as follows:

- **[no] ignition enable** – Enable/disable ignition power management.
- **ignition off-timer** <*value*> – After the ignition is turned off, the router will stay operational for this amount of time, then it turns off if the ignition is still off.
- **[no] ignition sense**– Enable/disable voltage sense
- **ignition undervoltage threshold** <*value*> – If the input voltage drops to levels below this threshold, it will cause the router to shut down.

The leaf nodes for this model are as follows:

- enable
- off-timer
- sense
- threshold-value-volt
- threshold-value-milli-volt

A Yang model file, Cisco-IOS-XE-ignition.yang is available for configuration model.

The ignition show CLI's for oper model are as follows:

show ignition – Shows all the ignition-related parameters

A Yang model file, Cisco-IOS-XE-ignition-oper-transform.yang is available for this purpose.

Support SNMP MIB for Ignition Power Management

The following is an example output from the **show ignition** CLI:

```
Status:
Ignition management: Disabled
Input voltage: 17.672 V
Ignition status: Power on
Ignition Sense: Disabled
Shutdown timer: 0.0 s to off [will begin power down at ~100 sec]
Config-ed battery: 12v
Thresholds:
Undervoltage: 9.000 V
Overvoltage: 37.000 V
Sense on: 13.200 V
Sense off: 12.800 V
Undervoltage timer: 20.0 s
Overvoltage timer: 1.0 s
Ignition-Off timer: 300.0 s
```

A MIB file, CISCO-IGNITION-MIB.my, is available to support the **show ignition** CLI.

The MIB file has the following fields:

- IgnitionManagement (1=True; 2=False, Boolean)
- InputVoltage (millivolt, Unsigned Integer)
- IgnitionStatus (Bootloader/Power on/Timing low voltage shut down…, State Index)

  - Bootloader (0)
  - Power On (1)
  - Low Delay (2)
  - Off Delay (3)
  - High Delay (4)
  - On Delay (5)
  - Monitor (6)
  - Sleep (7)
  - Unknown (8)

- IgnitionSense (1=True; 2=False, Boolean)
- ShutdownTimer (milliseconds, Unsigned Integer)
- ConfigBattery (volts, Interger)
- Undervoltage (millivolt, Unsigned Integer)
- Overvoltage (millivolt, Unsigned Integer)
- SenseOn (millivolts, Unsigned Integer)
- SenseOff (millivolts, Unsigned Integer)
- UndervoltageTimer (milliseconds, Unsigned Integer)
- OvervoltageTimer (milliseconds, Unsigned Integer)
- IgnitionOffTimer (milliseconds, Unsigned Integer)

CHAPTER 21

Configuring GPS

This chapter contains the following:

- GPS Overview, on page 199
- Cellular Modem-Based GPS, on page 201
- GPS/Dead Reckoning module (IRM-GNSS-ADR), on page 201
- National Marine Electronics Association (NMEA) IOx Support, on page 210
- NMEA UDP Socket Support, on page 210
- NMEA UDP Configuration with Yang, on page 214
- Yang Data Model Support, on page 216
- Example: Connecting to a Server Hosting a GPS Application, on page 217

GPS Overview

There are two ways to receive GPS information. There is Cellular modem GPS available in the LTE modules that support GPS, and there is a new dedicated GPS/Dead Reckoning module (IRM-GNSS-ADR) which provides more robust capabilities.

The IR1833 and IR1835 have a slot for a dedicated GPS field-replaceable unit (FRU) module, which will be used in addition to the one integrated in the LTE module, for more accurate dead reckoning performance. The part number is IRM-GNSS-ADR.

IRM-GNSS-ADR hardware is capable of supporting various GNSS constellations, however Cisco officially supports GPS only on this module at this time in IOS XE. Please inquire with your sales representative for a roadmap of support of additional constellations on the IRM-GNSS-ADR module.

Modem based GPS cannot provide the coordinates when there are no satellites in line of sight. The GPS module with DR capabilities provide the coordinates even when there are no satellites in line of sight.

With the addition of the dedicated GPS/Dead Reckoning module, along with Cellular module, there will be two sources of GPS location information. These two are independent and they can be retrieved using different CLIs:

- Cellular modem GPS information can be seen using the **show cellular** <*slot number*> **gps** command.
- GPS/DR module GPS info uses the **show platform hardware gps detail** command.

The following table provides a comparison of Modem based GPS and GPS/Dead Reckoning module based GPS.

Parameters	Modem Based GPS	GPS/Dead Reckoning Module
Type	Cellular Modem based GPS	FRU Based GPS
PIDs Supported	All of the IR1800 series	Only on IR1835 and IR1833
Configuration Modes	Standalone mode	No configuration is available to select modes. The device automatically selects either standalone | dead-reckoning mode based on satellites reception.
Number of satellites needed for co-ordinates	Standalone mode – 4	If a signal is received from 4 or more satellites, standalone GPS co-ordinates will provide co-ordinates else dead-reckoning will provide the gps co-ordinates.
Satellites Supported in show command	Co-ordinates seen in show commands output is based only on GPS.	Co-ordinates seen in the output of show commands is based only on GPS satellites. However, nmea traffic will show GPS, Gallileo, and Glonass.
Initial Calibration Required	No	Yes
Co-ordinates in Absence of Satellite	No Co-ordinates will be acquired and it stays in acquiring status.	The device seamlessly shifts to Dead-reckoning mode and provide co-ordinates based on calculation done by the FRU. The FRU takes into account vehicle speed, direction, last acquired coordinates from the GPS satellite, accelerometer and gyroscope. For dead reckoning to work, the device should have acquired the co-ordinates at least once before the loss of signal from satellite after the router boot up.
Device Name of Controller to use for Configuration	**controller cellular** <*slot*>	**controller gps-dr**
CLI to enable feature	**lte gps enable** **lte gps mode standalone** **Note** A modem power cycle is required after enabling the configuration.	**dead-reckoning enable**
CLI to configure nmea	**lte gps nmea**	nmea is automatically enabled once “dead-reckoning enable” is configured
CLI to configure nmea udp socket	**lte gps nmea ip udp** <*source_ip*> <*destination_ip*> <*destination_port*>	**dead-reckoning nmea udp** <*source_ip*> <*destination_ip*> <*destination_port*>
CLI to verify configuration under show running-config	**show run | sec controller cellular**<*slot*>	**show run | sec controller gps-dr**

Parameters	Modem Based GPS	GPS/Dead Reckoning Module
Show commands to verify gps output	**show cellular** <*slot*> **gps** **show controller cellular** <*slot*> **| inc GPS**	**show platform hardware gps detail** **show platform hardware gps mode** **show platform hardware gps status** **show platform hardware dead-reckoning**
Access to GPS nmea traffic on IOx side	Supported	Supported
Debug Command	**debug cellular**<*slot*> **messages gps** **debug cellular**<*slot*> **messages nmea**	**debug platform hardware gps_dr** *all* | *dr* | *gps* | *nmea*
Yang Model Support	Yes	Yes

Cellular Modem-Based GPS

Cellular modem based GPS is covered in the Cellular Pluggable Interface Module Configuration Guide.

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

This section describes the feature when using the GPS/Dead Reckoning-Based GPS Module.

Note GPS dead-reckoning is only available on the GPS Pluggable Module.

GPS Dead Reckoning

The GPS dead-reckoning feature is supported on the IR18xx Pro (1835-K9) and base2 (1833-K9) SKUs.

Dead Reckoning Overview

Dead Reckoning is a GPS fallback feature that provides users with location information during satellite signal interruption by calculating the current position by using a previously determined position, and advancing that position based upon known or estimated speeds over elapsed time and course.

IR18xx 3D Automotive Dead Reckoning (3D ADR) provides automotive-grade GPS services by using intelligent algorithms which combines satellite navigation data with wheel speed, gyroscope, and accelerometer data to deliver accurate positioning, even when satellite signals are partially or completely blocked.

The transition from satellite-based location service to internal-data based location service is transparent and automatic, based on the quality and presence of satellite signals.

The feature is disabled by default and CLIs are provided to enable and configure this feature. Enabling the feature automatically enables GPS, DR and the CAN bus. The feature shows the status, configuration and location data. The location data is streamed from the GPS module and will be forwarded to application via socket.

Command Line Interface

This section provides a description of the different CLIs used with GPS Dead Reckoning.

Command	Description
controller gps-dr	GPS-Dead Reckoning can be configured under controller gps-dr command.
dead-reckoning enable	Enables the GPS-DR feature.
no dead-reckoning enable	Disables the GPS-Dead Reckoning feature.
show platform hardware gps detail	Displays the following output: • Feature is enabled/disabled • GPS coordinates • Timestamps • Satellite information with SNR
show platform hardware gps status	Displays whether the feature is enabled or disabled, and the status of coordinates whether acquired or acquiring.
show platform hardware gps mode	Displays whether the feature is enabled or disabled, and whether Dead Reckoning is in use for location fix or not.
show platform hardware gps dead-reckoning	Displays the following output: • Firmware running on the GPS module • CAN transmit/receive count • Odometer reading • Accelerometer • Gyroscope readings • Whether Dead Reckoning is in use for location fix or not
debug platform hardware gps_dr gps | nmea | dr	Enables the debug logs for GPS | NMEA | DR **Note** The console gets flooded if logs are enabled. Configure **no logging console** and then enable this command to avoid flooding of console. Then perform **show log** to see the output. Perform **undebug all** to disable the debug. Make sure to enable logging on the console once debugging is disabled if needed.

Configuration Commands

To enable the GPS Dead Reckoning feature, perform the following:

```
Router#conf t
```

```
Enter configuration commands, one per line.  End with CNTL/Z.
Router(config)#controller gps-dr

Router(config-controller)#dead-reckoning enable
 Info: []: DR process enabled successfully.
```

To disable the GPS Dead Reckoning feature, perform the following:

```
Router#conf t
Enter configuration commands, one per line.  End with CNTL/Z.
Router(config)#controller gps-dr
 Router(config-controller)#no dead-reckoning enable
 Info: []: GPS/DR feature disabled successfully
```

Debug Commands

The following debugging commands are available:

```
Router#debug platform hardware gps_dr ?
  all   GPS DR all debug
  dr    GPS DR dr debug
  gps   GPS DR gps debug
  nmea  GPS NMEA messages debug
```

Show Commands

Use the following commands to view the status of the module and GPS details:

```
Router#show inventory
++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
INFO: Please use "show license UDI" to get serial number for licensing.
++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
NAME: "Chassis", DESCR: "Cisco Catalyst IR1835 Rugged Series Router"
PID: IR1835-K9         , VID: V00  , SN: FHH2416P00W
NAME: "Power Supply Module 0", DESCR: "Cisco IR1800 DC Power Supply"
PID: PWR-12V           , VID:      , SN:
NAME: "GE-POE Module", DESCR: "POE Module for On Board GE for Cisco IR183X"
PID: IR-183X-POE       , VID:      , SN:
NAME: "module 0", DESCR: "Cisco IR-1835-K9 Built-In NIM controller"
PID: IR-1835-K9        , VID:      , SN:
NAME: "NIM subslot 0/0", DESCR: "Front Panel 1 port Gigabitethernet Module"
PID: IR1835-1x1GE      , VID: V01  , SN:
NAME: "NIM subslot 0/1", DESCR: "IR1835-ES-4"
PID: IR1835-ES-4       , VID: V01  , SN:
NAME: "module F0", DESCR: "Cisco IR1835-K9 Forwarding Processor"
PID: IR1835-K9         , VID:      , SN:
NAME: "Gps-Dr", DESCR: "Dedicated GNSS/GPS/DR module"
PID: IRM-GNSS          , VID:V03 , SN:FOC243645DJ
```

When GPS co-ordinates are acquired from the Satellite, the following is the output from the show commands:

```
Router#show platform hardware gps detail
GPS Feature =  enabled
GPS Status =  GPS coordinates acquired
Latitude =  37 Deg 25 Min 4.7460 Sec North
Longitude =  121 Deg 55 Min 11.1840 Sec West
Timestamp (GMT) =  Tue Nov 24 03:03:55 2020
Fix type index =  0, Height =  40 m
HDOP =  4.1, GPS Mode Used =  GPS standalone
Satellite Info
----------------
```

```
Satellite #30, elevation 72, azimuth 43, SNR 0
Satellite #28, elevation 68, azimuth 277, SNR 0
Satellite #7, elevation 49, azimuth 89, SNR 0
Satellite #13, elevation 37, azimuth 312, SNR 0
Satellite #17, elevation 26, azimuth 185, SNR 25
Satellite #8, elevation 21, azimuth 43, SNR 0
Satellite #9, elevation 15, azimuth 160, SNR 17
Satellite #5, elevation 11, azimuth 260, SNR 26
Satellite #21, elevation 10, azimuth 77, SNR 0
Satellite #19, elevation 7, azimuth 194, SNR 24
Satellite #1, elevation 7, azimuth 103, SNR 0
Satellite #15, elevation 6, azimuth 322, SNR 0
Router#show platform hardware gps dead-reckoning
============================
GPS/DR Vendor Info: TELIT
GPS/DR module FW Version: V33-1.0.5-CLDR-4.7.10-N115R115-003291-3
CAN Bus Status:
  CAN Bus Tx Count: 0
  CAN Bus Rx Count: 0
  CAN NULL packet Bus RX Count: 0
  CAN Bus TX to DR Count: 0
  CAN Bus TX to DR error Count: 0
DR Sample TimeStamp in usec: 0
DR odometer count: 0
DR reverse status: 0
DR in use for location fix: No
time duration for loss of line of sight:
travel distance for loss of line of sight:
travel heading error at exit:
travel yaw error at exit:
travel gyro gain error at exit:
position error at exit:
position error ratio at exit:
position noise error at exit:
Raw Accel Data in X:  -2360
Raw Accel Data in Y:  16130
Raw Accel Data in Z:  0
Raw Gyro Data in X:  38
Raw Gyro Data in Y:  0
Raw Gyro Data in Z:  0
Router#

Router#show platform hardware gps status
GPS Feature =  enabled
GPS Status =  GPS coordinates acquired
Router#

Router#show platform hardware gps mode
GPS Feature =  enabled
DR in use for location fix: No
Router#
```

When the Antenna is not able to receive a satellite signal, it will switch to Dead Reckoning mode. During this mode only the output from the following show commands will change. The rest of the show commands output remains the same.

```
Router#show platform hardware gps mode
GPS Feature =  enabled
DR in use for location fix: Yes
Router#

Router#show platform hardware gps detail
GPS Feature =  enabled
GPS Status =  GPS coordinates acquired
```

```
Latitude =  37 Deg 25 Min 4.7460 Sec North
Longitude =  121 Deg 55 Min 11.1840 Sec West
Timestamp (GMT) =  Tue Nov 24 03:03:55 2020
Fix type index =  0, Height =  40 m
HDOP =  4.1, GPS Mode Used =  DR based GPS

Satellite Info
----------------
Satellite #30, elevation 72, azimuth 43, SNR 0
Satellite #28, elevation 68, azimuth 277, SNR 0
Satellite #7, elevation 49, azimuth 89, SNR 0
Satellite #13, elevation 37, azimuth 312, SNR 0
Satellite #17, elevation 26, azimuth 185, SNR 12
Satellite #8, elevation 21, azimuth 43, SNR 0
Satellite #9, elevation 15, azimuth 160, SNR 14
Satellite #5, elevation 11, azimuth 260, SNR 10
Satellite #21, elevation 10, azimuth 77, SNR 0
Satellite #19, elevation 7, azimuth 194, SNR 8
Satellite #1, elevation 7, azimuth 103, SNR 0
Satellite #15, elevation 6, azimuth 322, SNR 0
Router#
```

When GPS-Dead Reckoning is disabled, the output of the show commands appears as follows:

```
Router#show platform hardware gps detail
GPS Feature =  disabled
GPS Status =  GPS mode not enabled

Router#show platform hardware gps mode
GPS Feature =  disabled

Router#show platform hardware gps status
GPS Feature =  disabled
GPS Status =  GPS mode not enabled

Router#show platform hardware gps dead-reckoning
============================
GPS/DR Vendor Info:
GPS/DR module FW Version:
CAN Bus Status:
  CAN Bus Tx Count: 0
  CAN Bus Rx Count: 0
  CAN NULL packet Bus RX Count: 0
  CAN Bus TX to DR Count: 0
  CAN Bus TX to DR error Count: 0
DR Sample TimeStamp in usec: 0
DR odometer count: 0
DR reverse status: 0
DR in use for location fix: No
time duration for loss of line of sight:
travel distance for loss of line of sight:
travel heading error at exit:
travel yaw error at exit:
travel gyro gain error at exit:
position error at exit:
position error ratio at exit:
position noise error at exit:
Raw Accel Data in X:  0
Raw Accel Data in Y:  0
Raw Accel Data in Z:  0
Raw Gyro Data in X:  0
Raw Gyro Data in Y:  0
Raw Gyro Data in Z:  0
```

Feature Limitations

The following are feature limitations:

- To acquire an initial timestamp, it is required for the antenna to receive a signal from the Satellite when the device is powered on. Once acquired, the timestamp will be updated every second. Additionally, for DR to display coordinates, it is required for the antenna to acquire coordinates from the satellite at least once after the device is powered on, and DR is enabled. If the device had acquired coordinates before a power down, and the device is powered on again later, the device may try to show the coordinates based on the last known location.
- Cisco recommends using this feature only if the vehicle CAN bus is connected to this IR18xx. CAN Bus connection ensures that the GPS module is properly calibrated before the DR feature is fully functional, and location fix can be obtained even without line of sight of satellites.
- As long as sufficient satellite signals are received, the coordinates will be acquired as a standalone GPS module, whether connected to the vehicle via CAN bus or not. However, if there are no sufficient signals (or no signals at all), the location fix using DR will kick in and be accurate only if its CAN interface is connected to the vehicle via the CAN bus and gets all the required vehicle data.
- In the event of a GPS module initial deployment in the field, if CAN bus is not connected and satellite signal is not received, coordinates cannot be acquired.
- In the event of a GPS module has obtained the coordinates before, if Satellite signal is not received, coordinates will be acquired with the previously obtained value through DR, whether the CAN bus is not connected or not. The accuracy depends on whether the location has been moved or not since last location fix with satellite signals.

IR1800 GPS DR Module Calibration

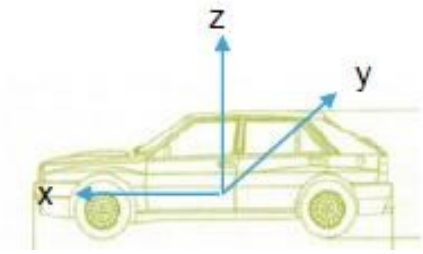
IR1800 GPS DR module provides Automotive Dead Reckoning (ADR) feature that allows the navigation on a automotive platform even when there are not a sufficient number of GPS satellite signals available. This is very common in IoT applications such as in urban canyons, underground tunnels, or any areas where line of sights of satellites are blocked.

To provide such a service in 3-D, the module needs to use the chipset's built-in three-axis gyro and accelerometer sensors to obtain the data for the change of the direction as well as the orientation and elevation of the chipset. In addition, to provide automotive grade service, it also allows the inputs of vehicle speed and direction data obtained directly from the CAN bus interface on IR1800.

In order to allow the module to properly interpret the data received from the three-axis sensors, it is necessary to provide the information describing the orientation of these built-in sensors relative to the vehicle, thus to properly calibrate the module, once the IR1800 is installed in the vehicle.

Calibration Requirements

The orientation of the vehicle is used in the module calibration as the base reference. It is arranged as illustrated in the following graphic:



- X axis: points to the vehicle forward direction
- Y axis: points to the right side of the vehicle when the viewer looks forward
- Z axis: points to upward with respect to the vehicle's motion plane.

The DR calibration process requires obtaining the orientation information of the sensors relative to this base reference. The default orientation for this module, as used by the DR algorithm is described as the following:

- IR1800 is installed and fastened on a stable base. Its front panel faces the right side of the vehicle, for example, 90 degrees clockwise away from the vehicle forward direction. The front panel is the side with the power connector and ethernet ports.
- With the default orientation, there is no need to input any sensor orientation data to the module for the calibration. The default data will be assumed by the module for the entire calibration process.

Release 17.6.1 does not support the calibration with non-default orientation. Cisco will provide such support if any future requirements arise. For now, customers need to follow this instruction for default orientation for the module calibration.

Calibration Process

Begin with the router installed as previously described, and with the GPS module installed. Configure the module from the command line:

Procedure

Step 1 This step is only required if the DR feature is not yet enabled. Otherwise, go directly to step 2.

Example:

```
Router(config)#controller gps
Router(config-controller)#no dead-reckoning enable
Info: []: GPS/DR feature disabled successfully
Router(config-controller)#end
```

Step 2 Enable the DR feature (clear the old calibration data)

Example:

```
Router#conf t
Enter configuration commands, one per line. End with CNTL/Z.
Router(config)#controller gps
Router(config-controller)#dead-reckoning enable
Info:  DR process enabled successfully.
```

Step 3 Set the CAN bus baudrate. Please check the user manual of the vehicle for the baudrate value. Typically, 500kbps is most common, but it can vary among different vehicle manufacturers.

Example:

```
Router(config-controller)#exit
Router(config)#canbus baudrate ?
  <125000-1000000>  enter baud rate ranging from 125000 to 1000000
Router(config)#canbus baudrate <baudrate of the vehicle CAN/OBDII interface>
```

Step 4 Reload or power cycle the router and wait until the router finishes rebooting.

Calibration Execution

Typically, a vehicle driving is needed to calibrate the DR module. This process is only needed once as long as the module orientation relative to the vehicle is not changed. This is true as long as the router or the module is never repositioned inside this vehicle.

Follow these steps for calibration:

Procedure

Step 1 Choose a location where there is open sky over the entire test for a good GPS signal. Calibration will require the vehicle to be moving.

Step 2 Wait approximately 2 minutes in a stopped position with the router and vehicle on. This allows the module to initialize the yaw rate offset with reliable values.

Step 3 Log into the console and run the following command to ensure that the GPS location is acquired:

Example:

```
Router#show platform hardware gps detail
GPS Feature =  enabled
GPS Status =  GPS coordinates acquired
Latitude =  37 Deg 25 Min 5.8200 Sec North
Longitude =  121 Deg 55 Min 9.1020 Sec West
Timestamp (GMT) =  Tue Mar  9 02:36:00 2021
Fix type index =  0, Height =  1 m
HDOP =  1.0, GPS Mode Used =  GPS standalone
Satellite Info
----------------
Satellite #12, elevation 70, azimuth 147, SNR 22
Satellite #25, elevation 63, azimuth 305, SNR 25
Satellite #2, elevation 51, azimuth 43, SNR 23
Satellite #5, elevation 41, azimuth 139, SNR 20
Satellite #29, elevation 33, azimuth 294, SNR 26
Satellite #6, elevation 13, azimuth 46, SNR 18
Satellite #31, elevation 10, azimuth 317, SNR 14
Satellite #24, elevation 7, azimuth 195, SNR 0
Satellite #18, elevation 4, azimuth 230, SNR 0
Satellite #82, elevation 58, azimuth 326, SNR 22
Satellite #80, elevation 57, azimuth 289, SNR 18
Satellite #79, elevation 41, azimuth 190, SNR 22
Satellite #81, elevation 34, azimuth 37, SNR 25
Satellite #83, elevation 28, azimuth 267, SNR 0
Satellite #66, elevation 19, azimuth 79, SNR 0
Satellite #73, elevation 16, azimuth 330, SNR 18
```

```
Satellite #67, elevation 8, azimuth 129, SNR 0
Satellite #65, elevation 4, azimuth 21, SNR 16
```

Step 4 After waiting for 2 minutes, drive in a straight line direction for at least five minutes at a constant speed. The speed should be greater than 35 km/h (approx. 22mph).

Step 5 Following the straight line drive, make several left and right turns of at least 90 degrees, allowing the system to calculate the gyro yaw rate gain.

Important Calibration will be improved with more turns completed. A minimum of 10 turns is recommended. Calibration should be performed in an open sky environment. Avoid urban canyons, tunnels, parking garages, dense foliage, etc.

Step 6 To complete calibration, the vehicle should stop and remain stationary for at least 10 seconds. For a full calibration to be successful, the above procedure must be followed.

Step 7 Use the following command to check if the calibration is done:

Example:

```
Router#show platform hardware gps dead

============================
GPS/DR Vendor Info: TELIT
GPS/DR module FW Version: V33-1.0.5-CLDR-4.7.10-N115R115-003291-3
DR Calibration Status:
  DR is calibrated
  Odometer is calibrated
  Gain is calibrated
  Offset is  calibrated
CAN Bus Status:
  CAN Bus Tx Count: 6856
  CAN Bus Tx error Count: 0
  CAN Bus Rx Count: 12724
  CAN NULL packet Bus RX Count: 0
  CAN Bus Rx unsupported packet Count: 0
  CAN Bus TX to DR Count: 12601
  CAN Bus TX to DR error Count: 123
DR data:
  DR Sample TimeStamp in usec: 0
  DR odometer count received from module: 54597690
  DR odometer count sent to module: 54597697
  DR odometer is  valid from module
  DR odometer delta count from module: 220
  DR reverse status: 0
```

Step 8 To clear calibration (for testing purposes), follow steps 1 and 2 under Calibration Process.

Dead Reckoning for GPS NMEA data streaming

The NMEA data streaming feature allows the user to forward NMEA streams over the Internet to any device running a 3rd party application for GPS location service.

The CLIs for IPv4 UDP sockets will be supported as feature parity with existing 4G modem based GPS functionality. No IPv6 UDP port support for NMEA data streaming is supported in existing 4G modem GPS at this time.

Command Line Interface

```
(config-controller)# dead-reckoning nmea ?
ip NMEA over IP interface

Router(config-controller)# dead-reckoning nmea udp ?
  A.B.C.D  Source address

(config-controller)#dead-reckoning nmea udp 10.3.4.5 ?
 A.B.C.D Destination address

(config-controller)#dead-reckoning nmea udp 10.1.1.1 10.3.4.5 ?
  Destination port

(config-controller)#dead-reckoning nmea udp 10.1.1.1 10.3.4.5 3456
```

National Marine Electronics Association (NMEA) IOx Support

From linux or the IOx container, the following tty is available for NMEA traffic:

- /dev/ttyTun9
- /dev/ttyS2

NMEA UDP Socket Support

In order to configure NMEA UDP socket support, you must enable the dead reckoning feature first, and then configure the NMEA UDP socket support. In order to disable NMEA UDP socket support, you must disable NMEA UDP socket support first, and then disable the dead reckoning feature.

See the following examples.

Enable the Feature

GPS is disabled:

```
Router#show platform hardware gps detail
GPS Feature =  disabled
GPS Status =  GPS mode not enabled

Router#show platform hardware gps status
GPS Feature =  disabled
GPS Status =  GPS mode not enabled

Router#show platform hardware gps mode
GPS Feature =  disabled

Router#show platform hardware gps dead-reckoning
============================
GPS/DR Vendor Info:
GPS/DR module FW Version:
DR Calibration Status:
  DR is not calibrated
  Odometer is not calibrated
  Gain is not calibrated
  Offset is not calibrated
```

```
CAN Bus Status:
  CAN Bus Tx Count: 0
  CAN Bus Tx error Count: 0

  CAN Bus Rx Count: 0
  CAN NULL packet Bus RX Count: 0
  CAN Bus Rx unsupported packet Count: 0

  CAN Bus TX to DR Count: 0
  CAN Bus TX to DR error Count: 0

DR data:
  DR Sample TimeStamp in usec: 0
  DR odometer count received from module: 0
  DR odometer count sent to module: 0
  DR odometer is not valid from module
  DR odometer delta count from module: 0
  DR reverse status: 0
DR in use for location fix: No
time duration for loss of line of sight:
travel distance for loss of line of sight:
travel heading error at exit:
travel yaw error at exit:
travel gyro gain error at exit:
position error at exit:
position error ratio at exit:
position noise error at exit:
Raw Accel Data in X:  0
Raw Accel Data in Y:  0
Raw Accel Data in Z:  0
Raw Gyro Data in X:  0
Raw Gyro Data in Y:  0
Raw Gyro Data in Z:  0
```

Configure GPS Dead Reckoning:

```
Router#config term
Enter configuration commands, one per line.  End with CNTL/Z.

Router(config)#controller Gps-Dr

Router(config-controller)#dead-reckoning enable
Info:  DR process enabled successfully.
```

Configure Dead Reckoning NMEA UDP:

```
Router(config-controller)#dead-reckoning nmea udp 192.0.2.163 192.0.2.240 11111
NMEA UDP Socket connect successful.

Router(config-controller)#end
```

Verify the status:

```
Router#show run | sec controller Gps-Dr
controller Gps-Dr
 dead-reckoning enable
 dead-reckoning nmea udp 192.0.2.163 192.0.2.240 11111
Router#show platform hardware gps detail
GPS Feature =  enabled
NMEA UDP socket is in use
NMEA UDP socket operational status: active
GPS Status =  GPS acquiring
Latitude =  0 Deg 0 Min 0 Sec North
Longitude =  0 Deg 0 Min 0 Sec East
```

```
Timestamp (GMT) =  Sun Jan  6 00:00:00 1980

Fix type index =  0
HDOP =  , GPS Mode Used =  not configured

Satellite Info
----------------

Router#show platform hardware gps status
GPS Feature =  enabled
NMEA UDP socket is in use
NMEA UDP socket operational status: active
GPS Status =  GPS acquiring

Router#show platform hardware gps mode
GPS Feature =  enabled
DR in use for location fix: No

Router#show platform hardware gps dead-reckoning
============================
GPS/DR Vendor Info:
GPS/DR module FW Version:
DR Calibration Status:
  DR is not calibrated
  Odometer is not calibrated
  Gain is not calibrated
  Offset is not calibrated

CAN Bus Status:
  CAN Bus Tx Count: 135
  CAN Bus Tx error Count: 0

  CAN Bus Rx Count: 0
  CAN NULL packet Bus RX Count: 0
  CAN Bus Rx unsupported packet Count: 0

  CAN Bus TX to DR Count: 0
  CAN Bus TX to DR error Count: 0

DR data:
  DR Sample TimeStamp in usec: 0
  DR odometer count received from module: 0
  DR odometer count sent to module: 0
  DR odometer is not valid from module
  DR odometer delta count from module: 0
  DR reverse status: 0

DR in use for location fix: No
time duration for loss of line of sight:
travel distance for loss of line of sight:
travel heading error at exit:
travel yaw error at exit:
travel gyro gain error at exit:
position error at exit:
position error ratio at exit:
position noise error at exit:
Raw Accel Data in X:  0
Raw Accel Data in Y:  0
Raw Accel Data in Z:  0
Raw Gyro Data in X:  0
Raw Gyro Data in Y:  0
Raw Gyro Data in Z:  0
```

Disable the Feature

Reverse the procedure to disable NMEA UDP Support. See the following examples:

```
Router#config term
Enter configuration commands, one per line.  End with CNTL/Z.

Router(config)#controller Gps-Dr

Router(config-controller)#no dead-reckoning nmea udp 192.0.2.163 192.0.2.240 11111
NMEA UDP Socket is disabled successfully.

Router(config-controller)#no dead-reckoning enable
Info:  GPS/DR feature disabled successfully

Router(config-controller)#end
```

Verify the status:

```
Router#show platform hardware gps detail
GPS Feature =  disabled
GPS Status =  GPS mode not enabled

Router#show platform hardware gps status
GPS Feature =  disabled
GPS Status =  GPS mode not enabled

Router# show platform hardware gps mode
GPS Feature =  disabled
Router#

Router# show platform hardware gps dead-reckoning
============================
GPS/DR Vendor Info:
GPS/DR module FW Version:
DR Calibration Status:
  DR is not calibrated
  Odometer is not calibrated
  Gain is not calibrated
  Offset is not calibrated

CAN Bus Status:
  CAN Bus Tx Count: 0
  CAN Bus Tx error Count: 0

  CAN Bus Rx Count: 0
  CAN NULL packet Bus RX Count: 0
  CAN Bus Rx unsupported packet Count: 0

  CAN Bus TX to DR Count: 0
  CAN Bus TX to DR error Count: 0

DR data:
  DR Sample TimeStamp in usec: 0
  DR odometer count received from module: 0
  DR odometer count sent to module: 0
  DR odometer is not valid from module
  DR odometer delta count from module: 0
  DR reverse status: 0

DR in use for location fix: No
time duration for loss of line of sight:
travel distance for loss of line of sight:
travel heading error at exit:
travel yaw error at exit:
```

```
travel gyro gain error at exit:
position error at exit:
position error ratio at exit:
position noise error at exit:
Raw Accel Data in X:  0
Raw Accel Data in Y:  0
Raw Accel Data in Z:  0
Raw Gyro Data in X:  0
Raw Gyro Data in Y:  0
Raw Gyro Data in Z:  0
Router#
```

NMEA UDP Configuration with Yang

The Yang Model can be used to enable the feature in the same way as the command line. The same rules apply:

In order to configure NMEA UDP socket support, you must enable the dead reckoning feature first, and then configure the NMEA UDP socket support. In order to disable NMEA UDP socket support, you must disable NMEA UDP socket support first, and then disable the dead reckoning feature.

Enable Dead Reckoning:

```
<rpc xmlns="urn:ietf:params:xml:ns:netconf:base:1.0" message-id="101">
  <edit-config>
    <target>
      <running/>
    </target>
    <config>
      <native xmlns="http://cisco.com/ns/yang/Cisco-IOS-XE-native">
        <controller>
          <Gps-Dr xmlns="http://cisco.com/ns/yang/Cisco-IOS-XE-controller">
            <dead-reckoning>
              <enable/>
            </dead-reckoning>
          </Gps-Dr>
        </controller>
      </native>
    </config>
  </edit-config>
</rpc>
```

Enable UDP Socket:

```
<rpc xmlns="urn:ietf:params:xml:ns:netconf:base:1.0" message-id="101">
  <edit-config>
    <target>
      <running/>
    </target>
    <config>
      <native xmlns="http://cisco.com/ns/yang/Cisco-IOS-XE-native">
        <controller>
          <Gps-Dr xmlns="http://cisco.com/ns/yang/Cisco-IOS-XE-controller">
            <dead-reckoning>
              <nmea>
                <udp>
                  <source-address>172.27.169.162</source-address>
                  <destination-address>172.27.169.140</destination-address>
                  <destination-port>11111</destination-port>
                </udp>
              </nmea>
```

```
            </dead-reckoning>
          </Gps-Dr>
        </controller>
      </native>
    </config>
  </edit-config>
</rpc>
```

Get Status:

```
<rpc xmlns="urn:ietf:params:xml:ns:netconf:base:1.0" message-id="101">
  <get>
    <filter>
      <gnss-dr-oper-data xmlns="http://cisco.com/ns/yang/Cisco-IOS-XE-gnss-dr-oper">
        <gnss-dr-data/>
      </gnss-dr-oper-data>
    </filter>
  </get>
</rpc>
```

Delete UDP Socket:

```
<rpc xmlns="urn:ietf:params:xml:ns:netconf:base:1.0" message-id="101">
  <edit-config>
    <target>
      <running/>
    </target>
    <config>
      <native xmlns="http://cisco.com/ns/yang/Cisco-IOS-XE-native">
        <controller>
          <Gps-Dr xmlns="http://cisco.com/ns/yang/Cisco-IOS-XE-controller">
            <dead-reckoning>
              <nmea>
              <udp xmlns:nc="urn:ietf:params:xml:ns:netconf:base:1.0" nc:operation="delete">

                  <source-address>172.27.169.162</source-address>
                  <destination-address>172.27.169.240</destination-address>
                  <destination-port>11111</destination-port>
                </udp>
              </nmea>
            </dead-reckoning>
          </Gps-Dr>
        </controller>
      </native>
    </config>
  </edit-config>
</rpc>
```

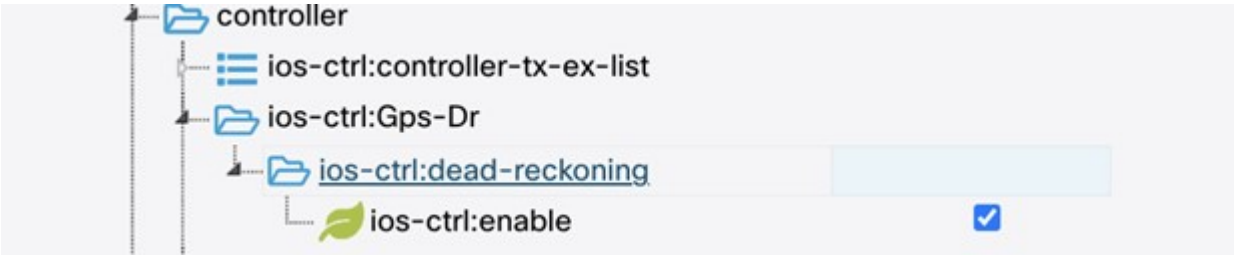
Delete Dead Reckoning Configuration:

```
<rpc xmlns="urn:ietf:params:xml:ns:netconf:base:1.0" message-id="101">
  <edit-config>
    <target>
      <running/>
    </target>
    <config>
      <native xmlns="http://cisco.com/ns/yang/Cisco-IOS-XE-native">
        <controller>
          <Gps-Dr xmlns="http://cisco.com/ns/yang/Cisco-IOS-XE-controller">
            <dead-reckoning>
              <enable xmlns:nc="urn:ietf:params:xml:ns:netconf:base:1.0"
nc:operation="delete"/>
            </dead-reckoning>
          </Gps-Dr>
        </controller>
      </native>
```

```
    </config>
  </edit-config>
</rpc>
```

Yang Data Model Support

Controller yang model is present under Cisco-IOS-XE-controller - Cisco-IOS-XE-native:



The following is an XML example to enable the GPS-Dead Reckoning feature:

```
<rpc xmlns="urn:ietf:params:xml:ns:netconf:base:1.0" message-id="101">
  <edit-config>
    <target>
      <running/>
    </target>
    <config>
      <native xmlns="http://cisco.com/ns/yang/Cisco-IOS-XE-native">
        <controller>
          <Gps-Dr xmlns="http://cisco.com/ns/yang/Cisco-IOS-XE-controller">
            <dead-reckoning>
              <enable/>
            </dead-reckoning>
          </Gps-Dr>
        </controller>
      </native>
    </config>
  </edit-config>
</rpc>
```

The following is an XML example to disable the GPS-Dead Reckoning feature:

```
<rpc xmlns="urn:ietf:params:xml:ns:netconf:base:1.0" message-id="101">
  <edit-config>
    <target>
      <running/>
    </target>
    <config>
      <native xmlns="http://cisco.com/ns/yang/Cisco-IOS-XE-native">
        <controller>
          <Gps-Dr xmlns="http://cisco.com/ns/yang/Cisco-IOS-XE-controller">
            <dead-reckoning>
              <enable xmlns:nc="urn:ietf:params:xml:ns:netconf:base:1.0"
nc:operation="delete"/>
            </dead-reckoning>
          </Gps-Dr>
        </controller>
      </native>
    </config>
  </edit-config>
</rpc>
```

GPS-Dead Reckoning oper commands are present under Cisco-IOS-XE-gnss-dr-oper model:

The following is an example of the XML for the oper command yang model of GPS-Dead Reckoning:

```
<rpc xmlns="urn:ietf:params:xml:ns:netconf:base:1.0" message-id="101">
  <get>
    <filter>
      <gnss-dr-oper-data xmlns="http://cisco.com/ns/yang/Cisco-IOS-XE-gnss-dr-oper">
        <gnss-dr-data/>
      </gnss-dr-oper-data>
    </filter>
  </get>
</rpc>
```

YANG operational and configuration data support will be provided for the previously mentioned CLIs as well.

The YANG model will allow the user to configure the UDP socket (ip address, UDP port etc, under XE-controller), as well as display the operational status of this feature (under XE-gnss-dr-oper), via YANG application software.

Example: Connecting to a Server Hosting a GPS Application

You can feed the NMEA data to a remote server that hosts the GPS application. The server can be connected to the router either directly using an Ethernet cable or through a LAN or WAN network. If the application supports serial port, run a serial port emulation program to create a virtual serial port over the LAN or WAN connection.

Note Microsoft Streets & Trips is a licensed software that you can download from the Microsoft website.

To connect a Cisco 4G LTE-Advanced through IP to a PC running Microsoft Streets & Trips, perform the following steps:

1. Connect the PC to the router using an Ethernet cable.
2. Ensure that the PC and router can ping.
3. Launch the serial port redirector on the PC.

4. Create a virtual serial port that connects to the NMEA port on the router.
5. Launch **Microsoft Streets & Trips** on your PC.
6. Select the GPS Menu.
7. Click Start Tracking.
8. If you have acquired a location fix from the **show cellular 0/3/0 gps** command output on the router, the current location is plotted on the graph, and a reddish brown dotted cursor with a circle around it is seen on the map.

Note If you have not acquired a location fix, the Microsoft application times out and disconnects.

CHAPTER 22

Information About SCADA

This section contains the following topics:.

- Supervisory Control And Data Acquisition (SCADA) Overview, on page 219
- Configuring Protocol Translation, on page 222
- Configuring the T101 Protocol Stack, on page 224
- Configuring the T104 Protocol Stack, on page 226
- Configuration Example, on page 229
- Configuring the DNP3 Protocol Stacks, on page 231
- Starting and Stopping the Protocol Translation Engine, on page 235
- Verifying Configuration, on page 235
- Debug Commands, on page 236

Supervisory Control And Data Acquisition (SCADA) Overview

SCADA refers to a control and management system employed in industries such as water management, electric power, and manufacturing. A SCADA system collects data from various types of equipment within the system and forwards that information back to a Control Center for analysis. Generally, individuals located at the Control Center monitor the activity on the SCADA system and intervene when necessary.

The Remote Terminal Unit (RTU) acts as the primary control system within a SCADA system. RTUs are configured to control specific functions within the SCADA system, which can be modified as necessary through a user interface.

On the IR1800, line is 0/2/0 or 0/2/1, same as the Async interface.

Role of the IR1800

In the network, the Control Center always serves as the master in the network when communicating with the IR1800. The IR1800 serves as a proxy master station for the Control Center when it communicates with the RTU.

The IR1800 provides protocol translation to serve as a SCADA gateway to do the following:

- Receive data from RTUs and relay configuration commands from the Control Center to RTUs.
- Receive configuration commands from the Control Center and relay RTU data to the Control Center
- Terminate incoming requests from the Control Center, when an RTU is offline

The IR1800 performs Protocol Translation for the following protocols:

- IEC 60870 T101 to/from IEC 60870 T104.
- DNP3 serial to DNP3 IP

Key Terms

The following terms are relevant when you configure the T101 and T104 protocol stacks on the IR1800:

- Channel–A channel is configured on each IR1800 serial port interface to provide a connection to a single RTU for each IP connection to a remote Control Center. Each connection transports a single T101 (RTU) or T104 (Control Center) protocol stack.
- Link Address–Refers to the device or station address.
- Link Mode (Balanced and Unbalanced)–Refers to the modes of data transfer.
  - An Unbalanced setting refers to a data transfer initiated from the master.
  - A Balanced setting can refer to either a master or slave initiated data transfer.
- Sector–Refers to a single RTU within a remote site.
- Sessions–Represents a single connection to a remote site.

The following terms are relevant when you configure the DNP3 protocol stacks on the on the IR1800:

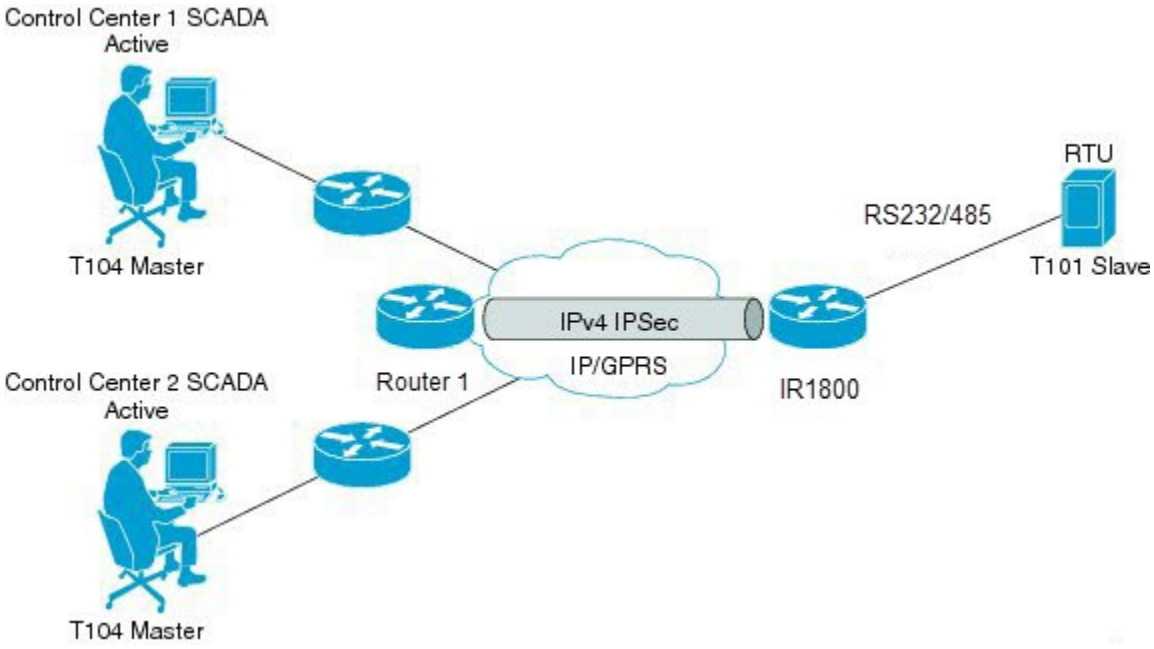
- Channel–A channel is configured on the IR1800 serial port interface to provide a connection to a single RTU for each IP connection to a remote Control Center. Each connection transports a single DNP3 serial (RTU) or DNP3 IP (Control Center) protocol stack.
- Link Address–Refers to the device or station address.
- Sessions–Represents a single connection to a remote site.

Protocol Translation Application

In Figure 40: Routers Within a SCADA System, on page 221 the IR1800 (installed within a secondary substation of the Utility Network) employs Protocol Translation to provide secure, end-to-end connectivity between Control Centers and RTUs within a SCADA System.

The IR1800 connects to the RTU (subordinate) through a RS232/RS485 connection. To protect the traffic when forwarded over public infrastructures (for example, cellular), the IR1800 forwards SCADA data from the RTU to the Control Center in the SCADA system through an IPSec tunnel (FlexVPN site-to-site or hub and spoke). The IPSec tunnel protects all traffic between the IR1800 and the Head-end aggregation router. SCADA traffic can be inspected through an IPS device positioned in the path of the SCADA traffic before it is forwarded to the proper Control Center.

Figure 40: Routers Within a SCADA System



Prerequisites

RTUs must be configured and operating in the network.

For each RTU that connects to the IR1800, you will need the following information for T101/T104:

- Channel information
  - Channel name
  - Connection type: serial
  - Link transmission procedure setting: unbalanced or balanced
  - Address field of the link (number expressed in octets)
- Session information
  - Session name
  - Size of common address of Application Service Data Unit (ASDU) (number expressed in octets)
  - Cause of transmission (COT) size (number expressed in octets)
  - Information object address (IOA) size (number expressed in octets)
- Sector information
  - Sector name
  - ASDU address, (number expressed in octets)

For each RTU that connects to the IR1800, you will need the following information for DNP3:

- Channel information
  - Channel name
  - Connection type: serial
  - Link address
- Session information
  - Session name

Guidelines and Limitations

Each channel supports only one session.

Each sessions supports only one sector.

Default Settings

T101/T104 Parameters	Default
Role for T101	Master
Role for T104	Slave

DNP3 Parameters	Default
Unsolicited Response (DNP3-serial)	Not Enabled
Send Unsolicited Message (DNP3-IP)	Enabled

Configuring Protocol Translation

This section includes the following topics:

Note Before making any configuration changes to a IR1800 operating with Protocol Translation, please review the section on Starting and Stopping the Protocol Translation Engine, on page 235.

Enabling the IR1800 Serial Port and SCADA Encapsulation

Before you can enable and configure Protocol Translation on the IR1800, you must first enable the serial port on the IR1800 and enable SCADA encapsulation on that port.

Before you begin

Determine the availability of a serial port on the IR1800.

Procedure

	Command or Action	Purpose
Step 1	**configure terminal**	Enters the global configuration mode.
Step 2	**interface async** *slot/port/interface*	Enters the interface command mode for the async slot/port/interface. *slot* –value of 0 *port* –value of 2 *interface* –value of 0 or 1
Step 3	**no shutdown**	Brings up the port, administratively.
Step 4	**encapsulation scada**	Enables encapsulation on the serial port for protocol translation and other SCADA protocols.

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 224
- Configuring the T104 Protocol Stack, on page 226
- Starting and Stopping the Protocol Translation Engine, on page 235

Prerequisites

Ensure that you have gathered all the required configuration information.

Enable the serial port and SCADA encapsulation.

Configuring the T101 Protocol Stack

Configure the channel, session, and sector parameters for the T101 protocol stack.

Procedure

	Command or Action	Purpose
Step 1	**configure terminal**	Enters global configuration mode.
Step 2	**scada-gw protocol t101**	Enters the configuration mode for the T101 protocol.
Step 3	**channel** *channel_name*	Enters the channel configuration mode for the T101 protocol. *channel_name* –Identifies the channel on which the serial port of the IR1800 communicates to the RTU. **Note** When the entered channel name does not already exist, the router creates a new channel. Entering the **no** form of this command deletes an existing channel. However, all sessions must be deleted before you can delete a channel.
Step 4	**role master**	Assigns the master role to the T101 protocol channel (default).
Step 5	**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 6	**link-addr-size {none | one | two}**	Defines the link address size in octets.
Step 7	**bind-to-interface async** *slot/port/interface*	Defines the IR1800 serial interface on which the system sends its T101 protocol traffic. *slot* –Value of 0 *port* –Value of 2 *interface* –Value of 0 or 1
Step 8	**exit**	Ends configuration of the channel and exits the channel configuration mode. Saves all settings.

	Command or Action	Purpose
Step 9	**session** *session_name*	Enters the session configuration mode and assigns a name to the session.
Step 10	**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 11	**common-addr-size** *{one | two | three}*	Defines the common address size in octets.
Step 12	**cot size** *{one | two | three}*	Defines the cause of transmission such as spontaneous or cyclic data schemes in octets.
Step 13	**info-obj-addr-size** *{one | two | three}*	Defines the information object element address size in octets.
Step 14	**link-addr-size** *{one | two | three}*	Defines the link address size in octets.
Step 15	**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 16	**exit**	Exits the session configuration mode.
Step 17	**sector** *sector_name*	Enters the sector configuration mode and assigns a name to the sector for the RTU. *sector_name* –Identifies the sector.
Step 18	**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 19	**asdu-addr** *asdu_address*	Refers to the ASDU structure address of the RTU.
Step 20	**exit**	Exits the sector configuration mode.
Step 21	**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)# role master
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 221)

Enable the serial port and SCADA encapsulation. (See Enabling the IR1800 Serial Port and SCADA Encapsulation, on page 222)

Procedure

	Command or Action	Purpose
Step 1	**configure terminal**	Enters configuration mode.
Step 2	**scada-gw protocol t104**	Enters the configuration mode for the T104 protocol.
Step 3	**channel** *channel_name*	Enters the channel configuration mode for the T104 protocol. *channel_name* –Identifies the channel on which the router communicates with the Control Center. **Note** When the entered channel name does not already exist, the router creates a new channel. Entering the **no** form of this command deletes an existing channel. However, all sessions must be deleted before you can delete a channel.

	Command or Action	Purpose
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	**cot size** *{one | two | three}*	Defines the cause of transmission (cot), such as spontaneous or cyclic data schemes in octets.
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 Control Center.
Step 17	**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 18	**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 19	**map-to-sector** *sector_name*	Maps the Control Center (T104) sector to the RTU (T101) sector.
Step 20	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)# cot-size two
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)# cot-size two
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)#
```

Configuration Example

The following example shows how to configure the serial port interface for T101 connection, configure T101 and T104 protocol stacks, and starts the Protocol Translation Engine on the IR1800.

```
router# configure terminal
router(config)# interface async 0/2/0
router (config-if)# no shutdown
router (config-if)# encapsulation scada
router (config-if)# exit
router(config)# scada-gw protocol t101
router(config-t101)# channel rtu_channel
router(config-t101-channel)# role master
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)# cot-size two
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)# cot-size two
router(config-t104-session)# exit
router(config-t104)# sector cc_master2-sector
router(config-t104-sector)# attach-to-session cc_master2
router(config-t104-sector)# asdu-adr 3
router(config-t104-sector)# map-to-sector rtu_sector
router(config-t104-sector)# exit
router(config-t104)# exit
router(config)# scada-gw enable
```

This example configures end-to-end communication between Control Centers and RTUs within a SCADA system using the DNP3 protocol stacks and starts the Protocol Translation Engine on the IR1800:

```
router# configure terminal
router(config)# interface async 0/2/0
router (config-if)# no shutdown
router (config-if)# encapsulation scada
router (config-if)# exit
router(config)# scada-gw protocol dnp3-serial
router(config-dnp3s)# channel rtu_channel
router(config-dnp3s-channel)# bind-to-interface async 0/2/0
router(config-dnp3s-channel)# link-addr source 3
router(config-dnp3s-channel)# unsolicited-response enable
router(config-dnp3s-channel)# exit
router(config-dnp3s)# session rtu_session
router(config-dnp3s-session)# attach-to-channel rtu_channel
router(config-dnp3s-session)# link-addr dest 3
router(config-dnp3s-session)# exit
router(config-dnp3s)# exit
router(config)# scada-gw protocol dnp3-ip
router(config-dnp3n)# channel cc_channel
```

```
router(config-dnp3n-channel)# link-addr dest 3
router(config-dnp3n-channel)# tcp-connection local-port default remote-ip any
router(config-dnp3n-channel)# exit
router(config-dnp3n)# session cc_session
router(config-dnp3n-session)# attach-to-channel cc_channel
router(config-dnp3n-session)# link-addr source 3
router(config-dnp3n-session)# map-to-session rtu_session
router(config-dnp3n)# exit
router(config)# exit
router(config)# scada-gw enable
```

Configuring the DNP3 Protocol Stacks

You can configure the DNP3 serial and DNP3 IP protocol stacks, which allow end-to-end communication between Control Centers and RTUs within a SCADA system.

Configuring DNP3 Serial

Configure the channel and session parameters for the DNP serial communication with an RTU.

Procedure

	Command or Action	Purpose
Step 1	**configure terminal**	Enters global configuration mode.
Step 2	**scada-gw protocol dnp3-serial**	Enters configuration mode for the DNP3 serial protocol.
Step 3	**channel** *channel_name*	Enters channel configuration mode for the DNP3 serial protocol. *channel_name* –Identifies the channel on which the router serial port communicates to the RTU. **Note** When the entered channel name does not already exist, the router creates a new channel Entering the **no** form of this command deletes an existing channel. However, all sessions must be deleted before you can delete a channel.
Step 4	**bind-to-interface async0/2/0**	Defines the router async interface on which the system sends its DNP3 protocol traffic.
Step 5	**link-addr source** *source_address*	Refers to the link address of the master. *source_address* –Range of values from 1 to 65535.
Step 6	**unsolicited-response enable**	(Optional) Allows unsolicited responses.

	Command or Action	Purpose
		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
```

Starting and Stopping the Protocol Translation Engine

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

Enabling the IR1800 Serial Port and SCADA Encapsulation, on page 222

Configuring T101 and T104 Protocol Stacks, on page 223

You must start the Protocol Translation Engine to use Protocol Translation on the IR1800.

Starting– After enabling SCADA encapsulation on the IR1800 serial port and configuring the T101 and T104 protocols on the IR1800, you can start the Protocol Translation Engine.

Stopping– Before you can make any configuration changes to Protocol Translation on the IR1800 with an active Protocol Translation Engine, you must stop the engine.

Procedure

	Command or Action	Purpose
Step 1	**configure terminal**	Enters global configuration mode.
Step 2	**scada-gw enable**	Starts (**scada-gw enable**) or stops (**no scada-gw enable**) the Protocol Translation Engine on the IR1800.

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.

Command	Purpose
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 18: 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 23

Raw Socket Transport

This section contains the following topics:

- Overview, on page 237
- Information About Raw Socket Transport, on page 237
- Prerequisites, on page 239
- Guidelines and Limitations, on page 240
- Default Settings, on page 240
- Configuring Raw Socket Transport, on page 240
- Verifying Configuration, on page 246
- Configuration Example, on page 246

Overview

Raw Socket Transport transports streams of characters from one serial interface to another over an IP network for utility applications.

This document describes Raw Socket Transport for the IR1800 and provides a reference section describing the Raw Socket Transport commands.

Information About Raw Socket Transport

Raw Socket is a method for transporting serial data through an IP network. The feature can be used to transport Supervisory Control and Data Acquisition (SCADA) data from Remote Terminal Units (RTUs). This method is an alternative to the Block Serial Tunnel (BSTUN) protocol.

Raw Socket Transport supports TCP or UDP as the transport protocol. An interface can be configured to use either protocol but not both at the same time. TCP transport is suitable for applications such as control applications that require acknowledged and sequenced delivery of data. For latency-sensitive applications such as line SEL relays, UDP transport provides faster transport of serial data than TCP.

Raw Socket Transport supports the following for the asynchronous serial interface:

- TCP as the transport protocol, with built-in auto TCP connection retry mechanism.
- Up to 32 TCP sessions.
- Interface configuration as a server, client, or a combination of both.

- One server interface, but multiple clients.
- VRF-awareness, which enables the router to send Raw Socket Transport traffic to a server host connected through a Virtual Private Network (VPN) Virtual Routing and Forwarding (VRF) interface.

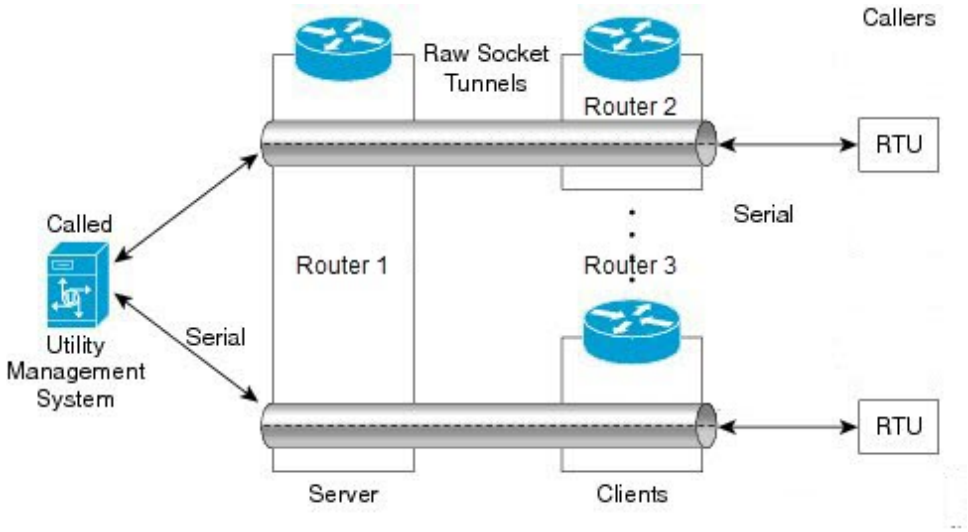
This section includes the following topics:

TCP Transport

TCP Raw Socket transport uses a client-server model. At most one server and multiple clients can be configured on a single asynchronous serial line. In client mode, the IR1800 can initiate up to 32 TCP sessions to Raw Socket servers, which can be other IR1800 routers or third-party devices.

The following figure shows a sample Raw Socket TCP configuration. In this example, serial data is transferred between RTUs and a utility management system across an IP network that includes several IR1800 routers. One IR1800 router (Router 1) acts as a Raw Socket server, listening for TCP connection requests from the other IR1800 routers (Router 2 and Router 3), which are configured as Raw Socket clients.

A Raw Socket client receives streams of serial data from the RTUs and accumulates this data in its buffer, then places the data into packets, based on user-specified packetization criteria. The Raw Socket client initiates a TCP connection with the Raw Socket server and sends the packetized data across the IP network to the Raw Socket server, which retrieves the serial data from the packets and sends it to the serial interface, and on to the utility management system.



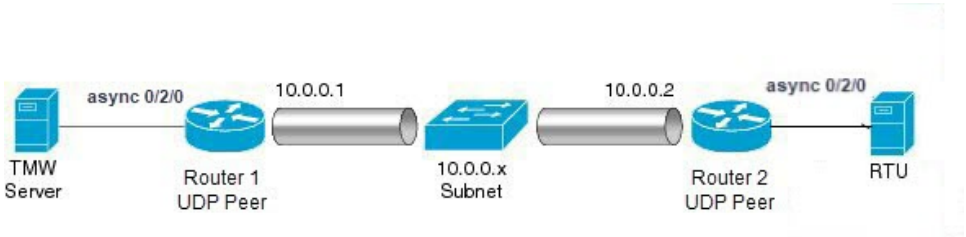
Note When you configure the serial link interface on the router as a server, the interface's peer is the serial link interface on the client router and vice versa.

UDP Transport

UDP transport uses a peer-to-peer model. Multiple UDP connections can be configured on an asynchronous serial line.

The following figure shows a sample Raw Socket UDP configuration. In this example, serial data is transferred between RTUs and a utility management system across an IP network that includes two routers (Router 1 which is an IR1800 and Router 2 which is an IR807) that are configured as Raw Socket UDP peers.

In this example, the Raw Socket UDP peer receives streams of serial data from the RTUs and accumulates this data in its buffer, then places the data into packets, based on user-specified packetization criteria. The Raw Socket UDP peer sends the packetized data across the IP network to the Raw Socket peer at the other end, which retrieves the serial data from the packets and sends it to the serial interface, and on to the utility management system.



Serial Data Processing

When the default serial protocol, Asynchronous Communication Protocol, is used, the streams of serial data received by a Raw Socket peer can be packetized based on the following criteria:

- **Packet length** – You can specify a packet length that triggers the IR1800 to transmit the serial data to the peer. Once the IR1800 collects this much data in its buffer, it packetizes the accumulated data and forwards it to the Raw Socket peer.
- **Packet-timer value** – The packet timer specifies the amount of time the IR1800 waits to receive the next character in a stream. If a character is not received by the time the packet timer expires, the data the IR1800 has accumulated in its buffer is packetized and forwarded to the Raw Socket peer.
- **Special character** – You can specify a character that will trigger the IR1800 to packetize the data accumulated in its buffer and send it to the Raw Socket peer. When the special character (for example, a CR/LF) is received, the IR1800 packetizes the accumulated data and sends it to the Raw Socket peer.

See the "Configuring Common Raw Socket Line Options" procedure on page 6 for information about configuring the processing options.

VRF-Aware Raw Socket

The VRF-aware Raw Socket Transport feature enables you to isolate Raw Socket traffic using a VRF for efficient management and control of serial data. After configuring a VRF, you can associate the serial interface configured for Raw Socket Transport with the VRF. See the Raw Socket VRF, on page 248 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

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.

Default Settings

Feature	Default Setting
Raw Socket Transport	Disabled.
Packet length	No packet length is configured.
Serial Protocol	Asynchronous Communication Protocol
Packet timeout	15 ms.
Special character	No special character is configured.
Raw Socket mode	Best-effort mode is off, not supported on the IR1801.
TCP idle timeout	5 minutes.

Configuring Raw Socket Transport

This section includes the following topics:

Enabling Raw Socket Transport on the Serial Interface

To enable Raw Socket Transport on the IR1800 router, you must first enable an asynchronous serial port and enable Raw Socket TCP or UDP encapsulation for that port.

Procedure

	Command or Action	Purpose
Step 1	**configure terminal**	Enters global configuration mode.
Step 2	**interface async0** */slot /port*	Enters the interface command mode for the async slot/port.
Step 3	**no ip address**	Disables IP processing on the interface.
Step 4	Do one of the following: • **encapsulation raw-tcp** • **encapsulation raw-udp**	Enables Raw Socket TCP encapsulation or UDP encapsulation for the serial port.

EXAMPLE

This example shows how to enable serial port 0/2/0 and how to enable Raw Socket TCP encapsulation on that port.

```
router# configure terminal
router(config)# interface async0/2/0
router(config-if)# no ip address
router(config-if)# encapsulation raw-tcp
router(config-if)# exit
```

Configuring Common Raw Socket Line Options

You can configure options common to all connections on a line. The common options apply to both TCP and UDP.

Before you begin

Enable Raw Socket Transport as described in Enabling Raw Socket Transport on the Serial Interface, on page 240.

Procedure

	Command or Action	Purpose
Step 1	**configure terminal**	Enters global configuration mode.
Step 2	**line 0** */slot /port*	Enters line command mode for the serial slot/port.
Step 3	**raw-socket packet-length** *length*	Specifies the packet size that triggers the IR1800 to transmit the data to the peer. When the IR1800 accumulates this much data in its buffer, it packetizes the data and forwards it to the Raw Socket peer. *length*— 2 to 1400 bytes. By default, the packet-length trigger is disabled.
Step 4	**raw-socket packet-timer** *timeout*	Specifies the maximum time in milliseconds the IR1800 waits to receive the next character in a stream. If a character is not received by the time the packet-timer expires, the accumulated data is packetized and forwarded to the Raw Socket peer. *timeout* —3 to 1000 ms. The default is 15 ms.
Step 5	**raw-socket spec-char** *ascii_char*	Specifies a character that will trigger the IR1800 to packetize the data accumulated in its buffer and send it to the Raw Socket peer. *ascii_char*— 0 to 255.

	Command or Action	Purpose
		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

```
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 240.

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 IR1800 listens for incoming connection requests from Raw Socket clients. *port* – Port number the server listens on. *ip_address* – (Optional) Local IP address on which the server listens for connection requests.
Step 4	**raw-socket tcp idle-timeout** *session_timeout*	Sets the Raw Socket Transport TCP session timeout for the asynchronous line interface. If no data is transferred between the client and server over this interval, then the TCP session closes. The client then automatically attempts to reestablish the TCP session with the server.

	Command or Action	Purpose
		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

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 240.

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.

	Command or Action	Purpose
		local_ip_address – (Optional) Local IP address that the client can also bind to. *local_port* – (Optional) Local port number that the client can also bind to.
Step 4	**raw-socket tcp idle-timeout** *session_timeout*	Sets the Raw Socket Transport TCP session timeout for the asynchronous line interface. If no data is transferred between the client and server over this interval, then the TCP session is closed. The client then automatically attempts to reestablish the TCP session with the server. This timeout setting applies to all Raw Socket Transport TCP sessions under this particular line. *session_timeout* –Currently configured session idle timeout in minutes. The default is 5 minutes.
Step 5	**tcp keepalive** *interval*	Sets the Raw Socket Transport TCP session keepalive interval for the asynchronous line interface. The router sends keepalive messages based on the configured interval. You may need to configure this interval, for example, when sending raw TCP traffic over a cellular interface. *interval* – Currently configured keepalive interval in seconds. Range is 1-864000 seconds. The default is 1 second.

What to do next

To remove a Raw Socket TCP client, use the **no raw-socket tcp client** command.

EXAMPLE

This example shows how to configure a Raw Socket TCP client for an asynchronous serial line. The IR1800 (router), serving as a Raw Socket client, initiates TCP sessions with a Raw Socket server and forwards packetized serial data to it. The router collects streams of serial data in its buffer; when it accumulates 827 bytes in its buffer, the router packetizes the data and forwards it to the Raw Socket server. If the router and the Raw Socket server do not exchange any data for 10 minutes, then the TCP session with the Raw Socket server closes, and the router attempts to reestablish the session with the Raw Socket server.

```
router# configure terminal
router(config)# line 0/2/0
router(config-line)# raw-socket tcp client 10.0.0.1 4000
router(config-line)# raw-socket packet-length 827
router(config-line)# raw-socket tcp idle-timeout 10
router(config-line)# exit
router(config)#
```

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 240.

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

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)#
```

Verifying Configuration

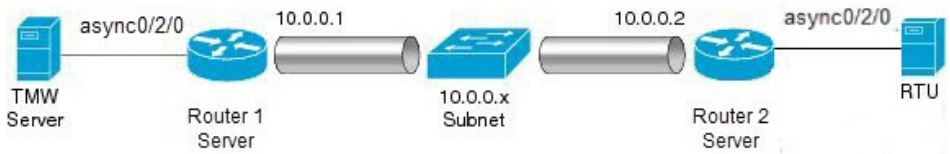
Command	Purpose
show running-config	Shows the configuration of the IR1800, including those features that are active and their settings.
show raw-socket tcp detail	Displays information about Raw Socket Transport TCP activity.
show raw-socket tcp sessions	Displays information about Raw Socket Transport TCP sessions.
show raw-socket tcp statistics	Displays Raw Socket Transport TCP statistics for each asynchronous serial line.
show raw-socket udp detail	Displays information about Raw Socket Transport UDP activity.
show raw-socket udp sessions	Displays information about Raw Socket Transport UDP sessions.
show raw-socket udp statistics	Displays Raw Socket Transport UDP statistics for each asynchronous serial line.
clear raw-socket statistics	Clears Raw Socket Transport statistics for a specific TTY interface or for all asynchronous serial lines.

Configuration Example

The following sections include Raw Socket Transport configuration examples:

Raw Socket TCP

The following example shows a Raw Socket Transport configuration in which an IR1800 router (Router 1) acts as the server, and another IR809 (Router 2) acts as the client.



The following table displays the configuration of the server and client IR1800s highlighted in the above figure:

IR1800 Server Configuration	IR807 Client Configuration
... interface async0/2/0 no ip address **encapsulation raw-tcp** ! ... line 0/2/0 **raw-socket tcp server 5000 10.0.0.1** **raw-socket packet-timer 3** **raw-socket tcp idle-timeout 5** ...	... interface async0 no ip address **encapsulation raw-tcp** ! interface async1 no ip address **encapsulation raw-tcp** ! ... line 1 **raw-socket tcp client 10.0.0.1 5000 10.0.0.2 9000** **raw-socket packet-length 32** **raw-socket tcp idle-timeout 5** line 2 **raw-socket tcp client 10.0.0.1 5000 10.0.0.2 9001** **raw-socket packet-length 32** **raw-socket tcp idle-timeout 5**

Raw Socket UDP

This example shows the configuration for a Raw Socket UDP connection between two IR1800 routers:

From Router1

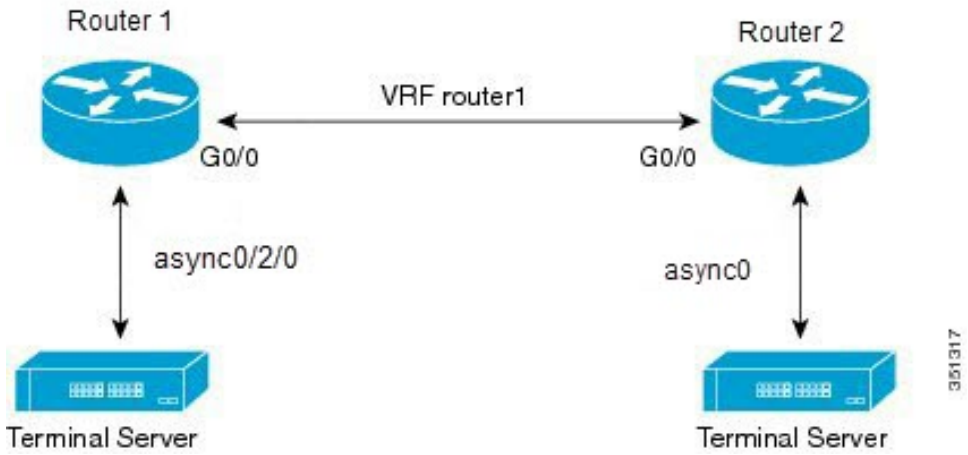
```
interface GigabitEthernet0/1
ip address 192.168.0.8 255.255.255.0
duplex auto
speed auto
interface async0/2/0
no ip address
encapsulation raw-udp
line 0/2/0
raw-socket udp connection 192.168.0.2 2 2
```

From Router2

```
interface GigabitEthernet0/1
ip address 192.168.0.2 255.255.255.0
load-interval 60
duplex auto
speed auto
no keepalive
interface async0/2/0
no ip address
encapsulation raw-udp
line 0/2/0
raw-socket udp connection 192.168.0.8 2 2
```

Raw Socket VRF

The following example shows a Raw Socket VRF configuration in which two routers, configured for Raw Socket Transport, connect through a VRF. Router1 is an IR1800, serves as the Raw Socket TCP server, and Router2 is an IR807 serves as the Raw Socket TCP client.



Following are the configurations of Router1 and Router2 as shown in the above figure:

Router1 Configuration

Defining VRF on the router:

```
vrf definition router1
 rd 100:1
 route-target export 100:3
 route-target import 100:3
 !
 address-family ipv4
 exit-address-family
```

Applying VRF configuration on the interface:

```
interface GigabitEthernet0/0
 vrf forwarding router1
 ip address 100.100.100.2 255.255.255.0
 duplex auto
 speed auto
```

Applying raw-tcp on the serial interface:

```
interface async0/2/0
 vrf forwarding router1
 no ip address
 encapsulation raw-tcp
```

Applying raw-tcp on the line:

```
line 0/2/0
 raw-socket tcp server 5000 4.4.4.4
```

Router2 Configuration

Defining VRF on the router:

```
vrf definition router1
 rd 100:1
 route-target export 100:3
 route-target import 100:3
 !
 address-family ipv4
 exit-address-family
```

Applying VRF configuration on the interface:

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

Applying raw-tcp on the serial interface:

```
interface async0
 vrf forwarding router1
 no ip address
 encapsulation raw-tcp
```

Applying raw-tcp on line:

```
line 1
  raw-socket tcp client 4.4.4.4 5000
```

CHAPTER 24

System Messages

This chapter contains the following sections:

- Information About Process Management, on page 251
- How to Find Error Message Details, on page 251

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 25

Environmental Monitoring

This chapter contains the following sections:

- Environmental Monitoring and Reporting Functions, on page 257
- Environmental Monitoring Functions, on page 257
- Environmental Reporting Functions, on page 258
- Additional References, on page 265
- Technical Assistance, on page 266

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 257
- Environmental Reporting Functions, on page 258

Environmental Monitoring Functions

The router provides a robust environment-monitoring system with several sensors that monitor the system temperatures. The following are some of the key functions of the environmental monitoring system:

- Monitoring temperature of CPUs and Motherboard
- Recording abnormal events and generating notifications
- Monitoring Simple Network Management Protocol (SNMP) traps
- Generating and collecting Onboard Failure Logging (OBFL) data
- Sending call home event notifications
- Logging system error messages
- Displaying present settings and status

Environmental monitoring functions use sensors to monitor the temperature of the cooling air as it moves through the chassis.

The router is expected to meet the following environmental operating conditions

- Non-operating Temperature: -40°F to 158°F (-40°C to 70°C)
- Non-operating Humidity: 5 to 95% relative humidity (non-condensing)
- Operating Temperature:

  -40° to 140°F (-40° to 60°C) in a sealed NEMA cabinet with no airflow

  -40° to 158°F (-40° to 70°C) in a vented cabinet with 40 lfm of air

  -40° to 167°F (-40° to 75°C) in a forced air enclosure with 200 lfm of air
- Operating Humidity: 10% to 95% relative humidity (non-condensing)
- Operating Altitude: -500 to 5,000 feet. Derate max operating temperature 1.5°C per 1000 feet.

The following table displays the levels of status conditions used by the environmental monitoring system.

Table 19: Levels of Status Conditions Used by the Environmental Monitoring System

Status Level	Description
Normal	All monitored parameters are within normal tolerance.
Warning	The system has exceeded a specified threshold. The system continues to operate, but operator action is recommended to bring the system back to a normal state.
Critical	An out-of-tolerance temperature or voltage condition exists. Although the system continues to operate, it is approaching shutdown. Immediate operator action is required.

The environmental monitoring system sends system messages to the console, for example, when the conditions described here are met:

Temperature and Voltage Exceed Max/Min Thresholds

The following example shows the warning messages indicating the maximum and minimum thresholds of the temperature or voltage:

```
Warnings :
--------
For all the temperature sensors (name starting with “Temp:”) above,
the critical warning threshold is 100C (100C and higher)
the warning threshold is 80C (range from 80C to 99C)
the low warning threshold is 1C (range from -inf to 1C).

For all voltage sensors (names starting with "V:"),
the high warning threshold starts at that voltage +10%. (voltage + 10% is warning)
the low warning threshold starts at the voltage -10%. (voltage - 10% is warning)
```

Environmental Reporting Functions

You can retrieve and display environmental status reports using the following commands:

- **show diag all eeprom**
- **show environment**
- **show environment all**
- **show inventory**
- **show platform**
- **show platform diag**
- **show platform software status control-processor**
- **show diag slot R0 eeprom detail**
- **show version**
- **show power**

These commands show the current values of parameters such as temperature and voltage.

The environmental monitoring system updates the values of these parameters every 60 seconds. Brief examples of these commands are shown below:

show diag all eeprom

```
Router# show diag all eeprom
MIDPLANE EEPROM data:

Product Identifier (PID) : IR1800-K9
Version Identifier (VID) : V00
PCB Serial Number : FOC21482ZQF
PCB Serial Number : FOC214822CK
PCB Serial Number : FOC21482SY7
Top Assy. Part Number : 68-6479-01
Top Assy. Revision : 13
Hardware Revision : 0.2
Asset ID :
CLEI Code : UNASSIGNED
Power/Fan Module P0 EEPROM data is not initialized

Power/Fan Module P1 EEPROM data is not initialized

Slot R0 EEPROM data:

Product Identifier (PID) : IR1800-K9
Version Identifier (VID) : V00
PCB Serial Number : FOC21482ZQF
PCB Serial Number : FOC214822CK
PCB Serial Number : FOC21482SY7
Top Assy. Part Number : 68-6479-01
Top Assy. Revision : 13
Hardware Revision : 0.2
CLEI Code : UNASSIGNED
Slot F0 EEPROM data:

Product Identifier (PID) : IR1800-K9
Version Identifier (VID) : V00
PCB Serial Number : FOC21482ZQF
PCB Serial Number : FOC214822CK
PCB Serial Number : FOC21482SY7
```

```
Top Assy. Part Number : 68-6479-01
Top Assy. Revision : 13
Hardware Revision : 0.2
CLEI Code : UNASSIGNED
Slot 0 EEPROM data:

Product Identifier (PID) : IR1800-K9
Version Identifier (VID) : V00
PCB Serial Number : FOC21482ZQF
PCB Serial Number : FOC214822CK
PCB Serial Number : FOC21482SY7
Top Assy. Part Number : 68-6479-01
Top Assy. Revision : 13
Hardware Revision : 0.2
CLEI Code : UNASSIGNED
SPA EEPROM data for subslot 0/0:

Product Identifier (PID) : IR1800-ES-5
Version Identifier (VID) : V01
PCB Serial Number :
Top Assy. Part Number : 68-2236-01
Top Assy. Revision : A0
Hardware Revision : 2.2
CLEI Code : CNUIAHSAAA
SPA EEPROM data for subslot 0/1 is not available

SPA EEPROM data for subslot 0/2 is not available

SPA EEPROM data for subslot 0/3 is not available

SPA EEPROM data for subslot 0/4 is not available

SPA EEPROM data for subslot 0/5 is not available

Router#
```

show environment:

```
Router# show environment
Number of Critical alarms: 0
Number of Major alarms: 0
Number of Minor alarms: 0

Slot Sensor Current State Reading Threshold(Minor,Major,Critical,Shutdown)
---------- ------------- --------------- ------------ ---------------------------------------
R0 Temp: LM75BXXX Normal 43 Celsius (75 ,80 ,90 ,na )(Celsius)

Router#
```

show environment all:

```
Router# show environment all
Sensor List: Environmental Monitoring
Sensor Location State Reading
Temp: LM75BXXX R0 Normal 48 Celsius
```

show inventory:

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

NAME: "Chassis", DESCR: "Cisco Catalyst IR1833 Rugged Series Router"
PID: IR1833-K9 , VID: V00 , SN: FCW2447P0EB

NAME: "Power Supply Module 0", DESCR: "Cisco IR1800 DC Power Supply"
PID: PWR-12V , VID: , SN:

NAME: "GE-POE Module", DESCR: "POE Module for On Board GE for Cisco IR183X"
PID: IR1800-I-POE , VID: V00 , SN: FOC24382K4W

NAME: "module 0", DESCR: "Cisco IR-1833-K9 Built-In NIM controller"
PID: IR1833-K9 , VID: , SN:

NAME: "NIM subslot 0/3", DESCR: "Cisco Wide Pluggable Form Factor WIFI6 AP Module"
PID: WP-WIFI6-B , VID: V00 , SN: FOC24490FEP

NAME: "NIM subslot 0/4", DESCR: "P-LTEA-LA Module"
PID: P-LTEA-LA , VID: V01 , SN: FOC22287JMC

NAME: "Modem on Cellular0/4/0", DESCR: "Sierra Wireless EM7430"
PID: EM7430 , VID: 1.0 , SN: 355813070165276

NAME: "NIM subslot 0/5", DESCR: "P-LTEA-LA Module"
PID: P-LTEA-LA , VID: V01 , SN: FOC22287JLZ

NAME: "Modem on Cellular0/5/0", DESCR: "Sierra Wireless EM7430"
PID: EM7430 , VID: 1.0 , SN: 355813070165524

NAME: "NIM subslot 0/0", DESCR: "Front Panel 1 port Gigabitethernet Module"
PID: IR1833-1x1GE , VID: V01 , SN:

NAME: "NIM subslot 0/1", DESCR: "IR1833-ES-4"
PID: IR1833-ES-4 , VID: V01 , SN:

NAME: "module R0", DESCR: "Cisco IR1833-K9 motherboard"
PID: IR1833-K9 , VID: V00 , SN: FOC24384177

NAME: "module F0", DESCR: "Cisco IR1833-K9 Forwarding Processor"
PID: IR1833-K9 , VID: , SN:
```

show platform:

```
Router# show platform
Chassis type: IR1833-K9

Slot Type State Insert time (ago)
--------- ------------------- --------------------- -----------------
0 IR1833-K9 ok 00:04:03
0/0 IR1833-1x1GE ok 00:01:22
0/1 IR1833-ES-4 ok 00:01:22
0/3 WP-WIFI6-B ok 00:01:22
0/4 P-LTEA-LA ok 00:01:21
0/5 P-LTEA-LA ok 00:01:21
```

```
R0 IR1833-K9 ok, active 00:04:03
F0 IR1833-K9 ok, active 00:04:03
P0 PWR-12V ok 00:02:00
GE-POE IR1800-I-POE ok 00:02:00
```

show platform diag:

```
Router# show platform diag
Chassis type: IR1833-K9

Slot: 0, IR1833-K9
Running state : ok
Internal state : online
Internal operational state : ok
Physical insert detect time : 00:01:09 (00:04:38 ago)
Software declared up time : 00:03:19 (00:02:28 ago)
CPLD version :
Firmware version : 3.9(REL)

Sub-slot: 0/0, IR1833-1x1GE
Operational status : ok
Internal state : inserted
Physical insert detect time : 00:03:51 (00:01:57 ago)
Logical insert detect time : 00:03:51 (00:01:57 ago)

Sub-slot: 0/1, IR1833-ES-4
Operational status : ok
Internal state : inserted
Physical insert detect time : 00:03:51 (00:01:57 ago)
Logical insert detect time : 00:03:51 (00:01:57 ago)

Sub-slot: 0/3, WP-WIFI6
Operational status : ok
Internal state : inserted
Physical insert detect time : 00:03:51 (00:01:57 ago)
Logical insert detect time : 00:03:51 (00:01:57 ago)

Sub-slot: 0/4, P-LTEA-LA
Operational status : ok
Internal state : inserted
Physical insert detect time : 00:03:51 (00:01:56 ago)
Logical insert detect time : 00:03:51 (00:01:56 ago)

Sub-slot: 0/5, P-LTEA-LA
Operational status : ok
Internal state : inserted
Physical insert detect time : 00:03:51 (00:01:56 ago)
Logical insert detect time : 00:03:51 (00:01:56 ago)

Slot: R0, IR1833-K9
Running state : ok, active
Internal state : online
Internal operational state : ok
Physical insert detect time : 00:01:09 (00:04:38 ago)
Software declared up time : 00:01:09 (00:04:38 ago)
CPLD version : 00000000
Firmware version : 3.9(REL)

Slot: F0, IR1833-K9
Running state : ok, active
```

```
Internal state : online
Internal operational state : ok
Physical insert detect time : 00:01:09 (00:04:38 ago)
Software declared up time : 00:03:30 (00:02:17 ago)
Hardware ready signal time : 00:00:00 (never ago)
Packet ready signal time : 00:03:36 (00:02:11 ago)
CPLD version : 00000000
Firmware version : 3.9(REL)

Slot: P0, PWR-12V
State : ok
Physical insert detect time : 00:03:13 (00:02:35 ago)

Slot: GE-POE, IR1800-I-POE
State : ok
Physical insert detect time : 00:03:13 (00:02:35 ago)
```

show platform software status control-processor:

```
Router# show platform software status control-processor
RP0: online, statistics updated 9 seconds ago
Load Average: healthy
1-Min: 0.32, status: healthy, under 5.00
5-Min: 0.33, status: healthy, under 5.00
15-Min: 0.35, status: healthy, under 5.00
Memory (kb): healthy
Total: 3959840
Used: 2894588 (73%), status: healthy
Free: 1065252 (27%)
Committed: 2435656 (62%), under 90%
Per-core Statistics
CPU0: CPU Utilization (percentage of time spent)
User: 0.50, System: 0.91, Nice: 0.00, Idle: 98.07
IRQ: 0.40, SIRQ: 0.10, IOwait: 0.00
CPU1: CPU Utilization (percentage of time spent)
User: 0.81, System: 0.30, Nice: 0.00, Idle: 98.48
IRQ: 0.20, SIRQ: 0.20, IOwait: 0.00
CPU2: CPU Utilization (percentage of time spent)
User: 0.81, System: 2.65, Nice: 0.00, Idle: 95.41
IRQ: 1.12, SIRQ: 0.00, IOwait: 0.00
CPU3: CPU Utilization (percentage of time spent)
User: 7.66, System: 17.05, Nice: 0.00, Idle: 70.58
IRQ: 4.59, SIRQ: 0.10, IOwait: 0.00
Router#
```

show diag slot R0 eeprom detail:

```
Router# show diag slot R0 eeprom detail
Slot R0 EEPROM data:
EEPROM version : 4
Compatible Type : 0xFF
Controller Type : 3457
Hardware Revision : 0.2
PCB Part Number : 73-18820-03
Board Revision : 02
Deviation Number : 0
Fab Version : 02
PCB Serial Number : FOC22106KKH
Top Assy. Part Number : 68-6479-03
Top Assy. Revision : 04
```

```
Chassis Serial Number : FCW2213TH07
Deviation Number : 0
RMA Test History : 00
RMA Number : 0-0-0-0
RMA History : 00
Product Identifier (PID) : IR1800-K9
Version Identifier (VID) : V00
CLEI Code : UNASSIGNED
Manufacturing Test Data : 00 00 00 00 00 00 00 00
Field Diagnostics Data : 00 00 00 00 00 00 00 00
Chassis MAC Address : 682c.7b4d.7880
MAC Address block size : 128
Asset ID :
Asset Alias :
PCB Part Number : 73-18821-03
Board Revision : 03
Deviation Number : 0
Fab Version : 02
PCB Serial Number : FOC22106KHD
PCB Part Number : 73-19117-02
Board Revision : 02
Deviation Number : 0
Fab Version : 01
PCB Serial Number : FOC22106KJ9
Asset ID :
Router#
```

show version:

```
Router# show version
Cisco IOS XE Software, Version 17.06.01prd23
Cisco IOS Software [Bengaluru], ISR Software (ARMV8EL_LINUX_IOSD-UNIVERSALK9_IOT-M), Version
 17.6.1prd23, RELEASE SOFTWARE (fc1)
Technical Support: http://www.cisco.com/techsupport
Copyright (c) 1986-2021 by Cisco Systems, Inc.
Compiled Tue 20-Jul-21 02:28 by mcpre

Cisco IOS-XE software, Copyright (c) 2005-2021 by cisco Systems, Inc.
All rights reserved. Certain components of Cisco IOS-XE software are
licensed under the GNU General Public License ("GPL") Version 2.0. The
software code licensed under GPL Version 2.0 is free software that comes
with ABSOLUTELY NO WARRANTY. You can redistribute and/or modify such
GPL code under the terms of GPL Version 2.0. For more details, see the
documentation or "License Notice" file accompanying the IOS-XE software,
or the applicable URL provided on the flyer accompanying the IOS-XE
software.

ROM: 3.9(REL)

IR1833 uptime is 13 hours, 6 minutes
Uptime for this control processor is 13 hours, 9 minutes
System returned to ROM by Firmware Upgrade
System image file is "bootflash:ir1800-universalk9.17.06.01prd23.SPA.bin"
Last reload reason: Reload Command

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

```
to comply with U.S. and local laws, return this product immediately.

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

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

Technology Package License Information:
-----------------------------------------------------------------
Technology Type Technology-package Technology-package
Current Next Reboot
-----------------------------------------------------------------
Smart License Perpetual network-advantage network-advantage
Smart License Subscription None None

The current throughput level is 50000 kbps
Smart Licensing Status: Registration Not Applicable/Not Applicable
cisco IR1833-K9 (1RU) processor with 470123K/6147K bytes of memory.
Processor board ID FCW2447P0EB
Router operating mode: Autonomous
MCU bootloader version: 0x22
MCU application version: 0x4d
1 Virtual Ethernet interface
6 Gigabit Ethernet interfaces
2 Serial interfaces
1 terminal line
4 Cellular interfaces
32768K bytes of non-volatile configuration memory.
3988088K bytes of physical memory.
7475198K bytes of Bootflash at bootflash:.
Configuration register is 0x2102
```

show power:

```
Router# show power
Main PSU :
Total Power Consumed: 8.16 Watts
Router#
```

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 26

IOx Application Hosting

This section contains the following topics:

- Application Hosting, on page 267
- Information About Application Hosting, on page 267
- Application Hosting on the IR1800 Industrial Integrated Services Router, on page 269
- How to Configure Application Hosting, on page 271
- Installing and Uninstalling Apps, on page 274
- Overriding the App Resource Configuration, on page 274
- **Verifying the Application Hosting Configuration**, on page 276
- Configuration Examples for Application Hosting, on page 277
- Native docker support, on page 278
- Digital IO for IOx container applications, on page 279
- Signed Application Support, on page 280

Application Hosting

A hosted application is a software as a service solution, and it can be run remotely using commands. Application hosting gives administrators a platform for leveraging their own tools and utilities.

This module describes the Application Hosting feature and how to enable it.

Information About Application Hosting

This section contains the following:

Need for Application Hosting

The move to virtual environments has given rise to the need to build applications that are reusable, portable, and scalable. Application hosting gives administrators a platform for leveraging their own tools and utilities. An application, hosted on a network device, can serve a variety of purposes. This ranges from automation, configuration management monitoring, and integration with existing tool chains.

Cisco devices support third-party off-the-shelf applications built using Linux tool chains. Users can run custom applications cross-compiled with the software development kit that Cisco provides.

IOx Overview

IOx is a Cisco-developed end-to-end application framework that provides application hosting capabilities for different application types on Cisco network platforms.

IOx architecture for the IR1800 is different compared to other Cisco platforms that use the hypervisor approach. In other platforms, IOx runs as a virtual machine. IOx is running as a process on the IR1800.

Cisco Application Hosting Overview

The IR1800 enables the user to deploy the application using the app-hosting CLIs. These app-hosting CLIs are not avaiable on the other older platforms. There are additional ways to deploy the applications using the Local Manager and Fog Director.

Application hosting provides the following services:

- Launches designated applications in containers.
- Checks available resources (memory, CPU, and storage), and allocates and manages them.
- Provides support for console logging.
- Provides access to services via REST APIs.
- Provides a CLI endpoint.
- Provides an application hosting infrastructure referred to as Cisco Application Framework (CAF).
- Helps in the setup of platform-specific networking (packet-path) via VirtualPortGroup and management interfaces.

The container is referred to as the virtualization environment provided to run the guest application on the host operating system. The Cisco IOS-XE virtualization services provide manageability and networking models for running guest applications. The virtualization infrastructure allows the administrator to define a logical interface that specifies the connectivity between the host and the guest. IOx maps the logical interface into the Virtual Network Interface Card (vNIC) that the guest application uses.

Applications to be deployed in the containers are packaged as TAR files. The configuration that is specific to these applications is also packaged as part of the TAR file.

The management interface on the device connects the application hosting network to the IOS management interface. The Layer 3 interface of the application receives the Layer 2 bridged traffic from the IOS management interface. The management interface connects through the management bridge to the container/application interface. The IP address of the application must be on the same subnet as the management interface IP address.

IOXMAN

IOXMAN is a process that establishes a tracing infrastructure to provide logging or tracing services for guest applications, except Libvirt, that emulates serial devices. IOXMAN is based on the lifecycle of the guest application to enable and disable the tracing service, to send logging data to IOS syslog, to save tracing data to IOx tracelog, and to maintain IOx tracelog for each guest application.

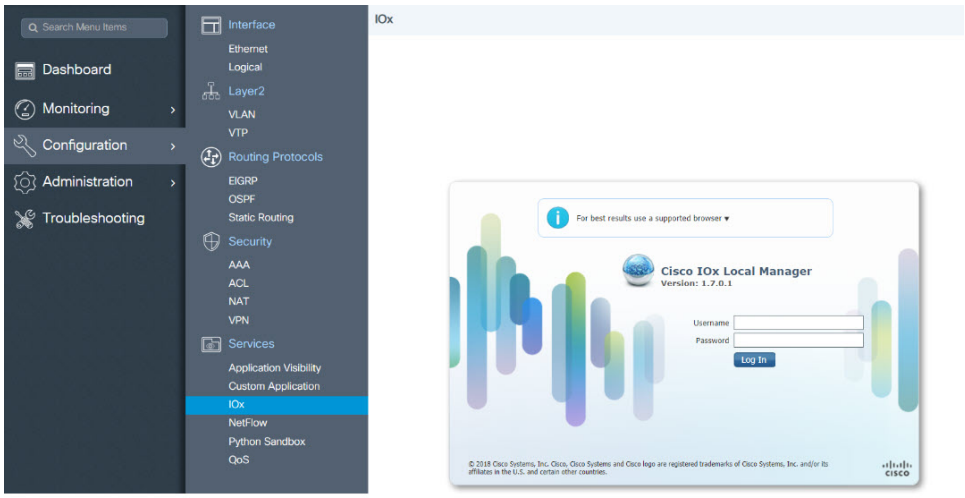
Application Hosting on the IR1800 Industrial Integrated Services Router

This section describes the application-hosting characteristics specific to the IR1800 Industrial Integrated Services Router.

Note The IR1800 CPU is not based on x86 architecture like other Routers. Therefore, this requires the application to comply with the ARM 64-bits architecture.

Application hosting can be achieved using the app-hosting cli's as well using the Local Manager and Fog Director. Application hosting using Local Manager is done through the WebUI. In order to deply the applications using Local Manager, WebUI should be enabled and then login to the Local Manager.

Figure 41: Local Manager



1. From the WebUI, click on **Configuration > Services > IOx**

2. Login using the username and password configured.

3. Follow the steps for the application lifecycle in the **Cisco IOx Local Manager Reference Guide** using this link: https://www.cisco.com/c/en/us/td/docs/routers/access/800/software/guides/iox/lm/reference-guide/1-7/b_iox_lm_ref_guide_1_7/b_iox_lm_ref_guide_1_7_chapter_011.html

The next section explains the deployment of an application using the app-hosting cli's.

VirtualPortGroup

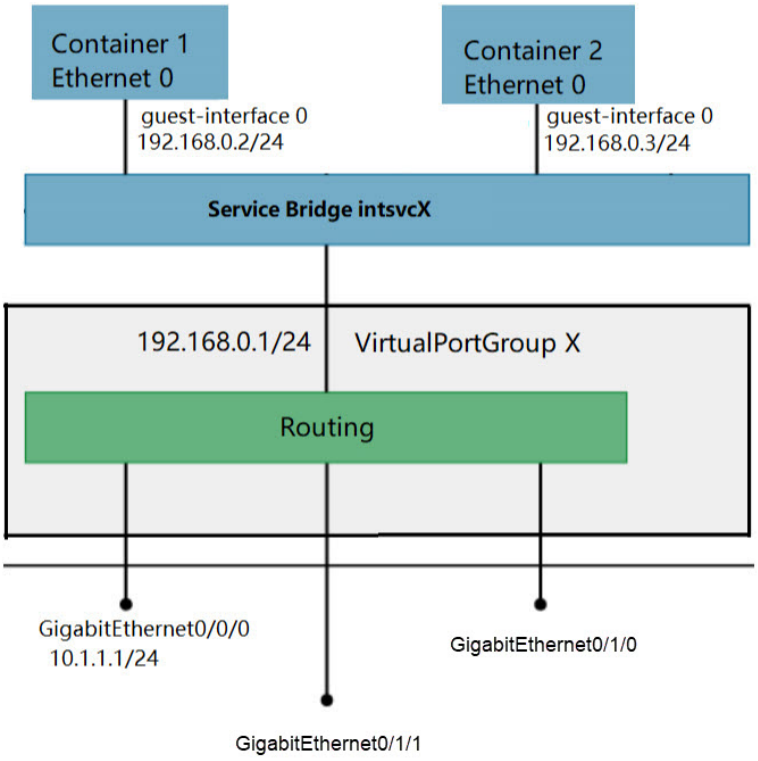
The VirtualPortGroup is a software construct on Cisco IOS that maps to a Linux bridge IP address. As such, the VirtualPortGroup represents the switch virtual interface (SVI) of the Linux container. Each bridge can contain multiple interfaces; each mapping to a different container. Each container can also have multiple interfaces.

VirtualPortGroup interfaces are configured by using the interface virtualportgroup command. Once these interfaces are created, IP address and other resources are allocated.

The VirtualPortGroup interface connects the application hosting network to the IOS routing domain. The Layer 3 interface of the application receives routed traffic from IOS. The VirtualPortGroup interface connects through the SVC Bridge to the container/application interface.

The following graphic helps to understand the relationship between the VirtualPortGroup and other interfaces, as it is different than the IR8x9 routers.

Figure 42: Virtual Port Group Mapping



vNIC

For the container life cycle management, the Layer 3 routing model that supports one container per internal logical interface is used. This means that a virtual Ethernet pair is created for each application; and one interface of this pair, called vNIC is part of the application container. The other interface, called vpgX is part of the host system.

NIC is the standard Ethernet interface inside the container that connects to the platform dataplane for the sending and receiving of packets. IOx is responsible for the gateway (VirtualPortGroup interface), IP address, and unique MAC address assignment for each vNIC in the container.

The vNIC inside the container/application are considered as standard Ethernet interfaces.

How to Configure Application Hosting

This section contains the following:

Enabling IOx

Perform this task to enable access to the IOx Local Manager. The IOx Local Manager provides a web-based user interface that you can use to manage, administer, monitor, and troubleshoot apps on the host system, and to perform a variety of related activities

Note In the steps that follow, IP HTTP commands do not enable IOX, but allow the user to access the WebUI to connect the IOX Local Manager.

Procedure

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device>enable`	Enables privileged EXEC mode. Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device#configure terminal`	Enters global configuration mode.
Step 3	**iox** **Example:** `Device(config)#iox`	Enables IOx.
Step 4	**ip http server** **Example:** `Device(config)#ip http server`	Enables the HTTP server on your IP or IPv6 system.
Step 5	**ip http secure-server** **Example:** `Device(config)#ip http secure-server`	Enables a secure HTTP (HTTPS) server.
Step 6	**username** *name* **privilege** *level* **password** {**0** | **7** | *user-password* }*encrypted-password* **Example:** `Device(config)#username cisco privilege 15 password 0 cisco`	Establishes a username-based authentication system and privilege level for the user. The username privilege level must be configured as 15.

	Command or Action	Purpose
Step 7	**end** **Example:** `Device(config-if)#end`	Exits interface configuration mode and returns to privileged EXEC mode

Configuring a VirtualPortGroup to a Layer 3 Data Port

Multiple Layer 3 data ports can be routed to one or more VirtualPortGroups or containers. VirutalPortGroups and Layer 3 data ports must be on different subnets.

Enable the **ip routing** command to allow external routing on the Layer 3 data-port.

Procedure

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device>enable`	Enables privileged EXEC mode. Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device#configure terminal`	Enters global configuration mode.
Step 3	**ip routing** **Example:** `Device(config)#ip routing`	Enables IP routing. The **ip routing** command must be enabled to allow external routing on Layer 3 data ports.
Step 4	**interface** *type number* **Example:** `Device(config)#interface gigabitethernet 0/0/0`	Configures an interface and enters interface configuration mode.
Step 5	**no switchport** **Example:** `Device(config)#no switchport`	Places the interface in Layer 3 mode, and makes it operate more like a router interface rather than a switch port.
Step 6	**ip address** *ip-address mask* **Example:** `Device(config)#ip address 10.1.1.1 255.255.255.0`	Configures an IP address for the interface.
Step 7	**exit** **Example:** `Device(config-if)#exit`	Exits interface configuration mode and returns to global configuration mode.

	Command or Action	Purpose
Step 8	**interface** *type number* **Example:** `Device(config)#interface virtualportgroup 0`	Configures an interface and enters interface configuration mode.
Step 9	**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 10	**end** **Example:** `Device(config-if)#end`	Exits interface configuration mode and returns to privileged EXEC mode
Step 11	**configure terminal** **Example:** `Device#configure terminal`	Enters global configuration mode.
Step 12	**app-hosting appid** *app_number* **Example:** `Device(config)#app-hosting appid app1`	Configures the application and enters the application configuration mode.
Step 13	**app-vnic gateway0 virtualportgroup 0 guest-interface 0** **Example:** `Device(config-app-hosting)#app-vnic gateway0 virtualportgroup 0 guest-interface 0`	Configures the application interface and the gateway of the application.
Step 14	**guest-ipaddress** *ip_address* **netmask** *netmask* **Example:** `Device(config-app-hosting-gateway0)#guest-ipaddress 192.168.0.2 netmask 255.255.255.0`	Configures the application Ethernet interface ip address.
Step 15	**app-default-gateway** *ip_address* **guest-interface 0** **Example:** `Device(config-app-hosting-gateway0)#app-default-gateway 192.168.0.1 guest-interface 0`	Configures the default gateway for the application.
Step 16	**end** **Example:** `Device#end`	Exits interface configuration mode and returns to privileged EXEC mode

Installing and Uninstalling Apps

Procedure

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device>enable`	Enables privileged EXEC mode. Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device#configure terminal`	Enters global configuration mode.
Step 3	**app-hosting install appid** *application-name* **package** *package-path* **Example:** `Device#app-hosting install appid lxc_app package flash:my_iox_app.tar`	Installs an app from the specified location. The app can be installed from any local storage location such as, flash, bootflash, and usbflash0.
Step 4	**app-hosting start appid** *application-name* **Example:** `Device#app-hosting start appid app1`	Starts the application. Application start-up scripts are activated.
Step 5	**app-hosting stop appid** *application-name* **Example:** `Device#app-hosting stop appid app1`	Stops the application.
Step 6	**app-hosting deactivate appid** *application-name* **Example:** `Device#app-hosting deactivate appid app1`	Deactivates all resources allocated for the application.
Step 7	**app-hosting uninstall appid** *application-name* **Example:** `Device#app-hosting uninstall appid app1`	Uninstalls the application. Uninstalls all packaging and images stored. All changes and updates to the application are also removed.

Overriding the App Resource Configuration

Resource changes will take effect only after the app-hosting activate command is configured.

Procedure

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device>enable`	Enables privileged EXEC mode. Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device#configure terminal`	Enters global configuration mode.
Step 3	**app-hosting appid** *name* **Example:** `Device(config-app-hosting)#app-resource profile custom`	Enables application hosting and enters application hosting configuration mode.
Step 4	**app-resource profile** *name* **Example:** `Device#app-hosting start appid app1`	Configures the custom application resource profile, and enters custom application resource profile configuration mode. Only the custom profile name is supported.
Step 5	**cpu** *unit* **Example:** `Device(config-app-resource-profile-custom)# cpu 800`	Changes the default CPU allocation for the application. Resource values are application-specific, and any adjustment to these values must ensure that the application can run reliably with the changes.
Step 6	**memory** *memory* **Example:** `Device(config-app-resource-profile-custom)# memory 512`	Changes the default memory allocation.
Step 7	**vcpu** *number* **Example:** `Device(config-app-resource-profile-custom)# vcpu 2`	Changes the virtual CPU (vCPU) allocation for the application.
Step 8	**end** **Example:** `Device(config-app-resource-profile-custom)# end`	Exits custom application resource profile configuration mode and returns to privileged EXEC mode.

Verifying the Application Hosting Configuration

Procedure

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

Step 3	**show iox-service**
	Example:
	`Device(config)#show iox-service`
	`IOx Infrastructure Summary:`
	`---`
	`IOx service (CAF) 1.8.0.2 : Running`
	`IOx service (HA) : Not Supported`
	`IOx service (IOxman) : Running`
	`Libvirtd 1.3.4 : Running`
	`Device#`
		Displays the status of all IOx services.

Step 4	**show app-hosting detail**
	Example:
	`Device#show app-hosting detail`
	`App id : app1`
	`Owner : iox`
	`State : RUNNING`
	`Application`
	`Type : lxc`
	`Name : nt08-stress`
	`Version : 0.1`
	`Description : Stress Testing Application`
	`Path : usbflash0:my_iox_app.tar`
	`Activated profile name : custom`
	`Resource reservation`
	`Memory : 64 MB`
	`Disk : 2 MB`
	`CPU : 500 units`
	`Attached devices`
	`Type Name Alias`
	`---`
	`serial/shell iox_console_shell serial0`
	`serial/aux iox_console_aux serial1`
		Displays detailed information about the application.

	Command or Action	Purpose

	`  serial/syslog  iox_syslog serial2`
	`  serial/trace  iox_trace serial3`
	
	`Network interfaces`
	
	`----------`
	`eth0:`
	`  MAC address  :`
	`52:54:dd:fa:25:ee`	

Step 5	**show app-hosting list**
	Example:
	`Device#show app-hosting list`
	`App id  State`
	`----------`
	`app1  RUNNING`
		Displays the list of applications and their status.

Configuration Examples for Application Hosting

See the following examples:

Example: Enabling IOx

```
Device> enable
Device# configure terminal
Device(config)# iox
Device(config)# ip http server
Device(config)# ip http secure-server
Device(config)# username cisco privilege 15 password 0 cisco
Device(config)# end
```

Example: Configuring a VirtualPortGroup to a Layer 3 Data Port

```
Device> enable
Device# configure terminal
Device(config)# ip routing
Device(config)# interface gigabitethernet 0/0/0
Device(config-if)# 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
```

Native docker support

Native Docker Support enables users to deploy the docker applications on the IR1800. The application lifecycle process is similar to the procedure in the Installing and Uninstalling Apps section. For docker applications, entry point configuration is required as part of the application configuration. Please refer to the following example for the entry point configuration.

```
Router#conf t
Enter configuration commands, one per line. End with CNTL/Z.
Router(config)#app-hosting appid app3
Router(config-app-hosting)#app-vnic gateway0 virtualportgroup 0 guest-interface 0
Router(config-app-hosting-gateway0)#guest-ipaddress 192.168.0.7 netmask 255.255.255.0
Router(config-app-hosting-gateway0)#app-default-gateway 192.168.0.1 guest-interface 0
Router(config-app-hosting)#app-resource docker
Router(config-app-hosting-docker)#run-opts 1 "--entrypoint '/bin/sleep 10000'"
Router(config-app-hosting-docker)#end
Router#
```

The output for docker applications is shown in the following example:

```
Router#show app-hosting detail
App id : app1
Owner : iox
State : RUNNING
Application
Type : docker
Name : aarch64/busybox
Version : latest
Description :
Path : bootflash:busybox.tar
Activated profile name : custom
Resource reservation
Memory : 431 MB
Disk : 10 MB
CPU : 577 units
VCPU : 1
Attached devices
Type Name Alias
---------------------------------------------
serial/shell iox_console_shell serial0
serial/aux iox_console_aux serial1
```

```
serial/syslog iox_syslog serial2
serial/trace iox_trace serial3
Network interfaces
---------------------------------------
eth0:
MAC address : 52:54:dd:e9:ab:7a
IPv4 address : 192.168.0.7
Network name : VPG0
Docker
------
Run-time information
Command :
Entry-point : /bin/sleep 10000
Run options in use : --entrypoint '/bin/sleep 10000'
Application health information
Status : 0
Last probe error :
Last probe output :
Router#
```

Digital IO for IOx container applications

IOx container applications are able to access the digital IO. There is a CLI for alarm contact command.

```
Router(config)# alarm contact ?
  <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 IR1800. All ports are initially set to Input mode with voltage pulled up to 3.3v.

The following are examples of IOCTL calls:

Read Digital IO Port:

```
cat /dev/dio-1
```

Write to Digital IO Port:

```
echo 0 > /dev/dio-1
echo 1 > /dev/dio-1
```

Change mode:

```
echo out > /dev/dio-1
echo in > /dev/dio-1
```

List of IOCTLs supported:

```
DIO_GET_STATE = 0x1001
DIO_SET_STATE = 0x1002
DIO_GET_MODE = 0x1003
DIO_SET_MODE_OUTPUT = 0x1004
DIO_SET_MODE_INPUT = 0x1005
```

```
DIO_GET_THRESHOLD 0x1006
DIO_SET_THRESHOLD = 0x1007
DIO_GET_VOLTAGE = 0x1009
```

Read State using IOCTL:

```
import fcntl, array
file = open("/dev/dio-1","rw")
state = array.array('L',[0])
fcntl.ioctl(file, DIO_GET_STATE, state)
print(state[0])
```

Change mode using IOCTL:

```
import fcntl
file = open("/dev/dio-1","rw")
fcntl.ioctl(file, DIO_SET_MODE_OUTPUT, 0)
```

Signed Application Support

Cisco Signed applications are now supported on the IR1800. In order to install a signed application, signed verification has to be enabled on the device. Signed verification can be enabled by following the following instructions.

```
Router#configure terminal
Enter configuration commands, one per line. End with CNTL/Z.
Router(config)#
Router(config)#app-hosting signed-verification
Router(config)#
Router(config)#exit
```

After enabling the signed verification, follow the instructions in the Installing and Uninstalling Apps section under IOx Application Hosting in order to install the application.

CHAPTER 27

Serial Relay Service

This chapter contains the following:

- IOx Serial Relay Service, on page 281
- Data Paths, on page 281
- Configuration Commands, on page 283

IOx Serial Relay Service

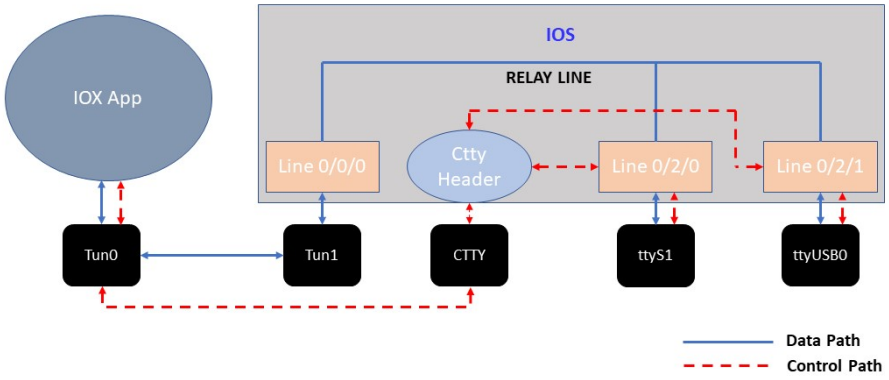
IOx Serial Relay service on the IR1800 enables IOx apps to communicate with the Async Serial port (/dev/ttyS1 or /dev/ttyUSB0 under IOS-XE). The configuration of IOx Serial Relay service is similar to that of the IR800.

Data Paths

On the IR1800, IOS-XE has complete control over the data path and control path of the Async Serial port. This aspect is essential to other encapsulations supported on the Aysnc port such as PPP, raw-socket, SCADA, etc. The IOx app is never allowed to exercise full control over the device. All data and configurations are passed through IOS-XE before going to the device.

Instead of exposing the actual Serial port to IOx apps, the Serial relay service creates a software emulated serial tty device enumerated as /dev/ttyTun0 (shown below). The pair of devices /dev/ttyTun0 and /dev/ttyTun1 represent a data tunnel whose primary function is to act as a pass-through gateway during any data transfer. /dev/ttyTun1 is open by IOS-XE and all the ingress/egress data from IOS to the app uses this device during data transfer. Line 0/0/0 is used to communicated with /dev/ttyTun1. Serial relay service should be configured beforehand to allow the connection between two lines.

Figure 43: Data Paths



Data Path:

1. When the IOx app sends a character to /dev/ttyTun0, the tunnel driver automatically pushes the data to /dev/ttyTun1.
2. IOS reads the data which it then passes to the Serial relay service.
3. The Serial relay service retrieves information about the other end of the relay service (Line 0/2/0 or Line 0/2/1 in this case) and forwards the data to the Line's buffer.
4. The line driver actively pushes the data into the actual serial device (/dev/ttyS1 or /dev/ttyUSB0) based on buffer availability.
5. The reverse path functions the same with the roles of /dev/ttyS1 or /dev/ttyUSB0 and /dev/tun0 reversed.

Control Path:

1. When the IOx app performs TCGETS ioctl call on /dev/ttyTun0, the tunnel driver uses /dev/cttyTun to send request to the CTTY handler service running in IOS.
2. CTTY handler service and the kernel driver use a client-server architecture to communicate configuration objects.
3. Upon receiving the request about TCGETS from /dev/cttyTun, the CTTY handler examines the request and requests Line driver to populate the required data into control data structures.
4. Upon receiving the control data structures, CTTY handler sends out a response to /dev/cttyTun which eventually goes back to /dev/ttyTun0.
5. /dev/ttyTun0 passes the control data to IOx app as requested.

6. Similar path can be extrapolated for TCSETS where the CTTY handler requests the Line driver to update the settings of the underneath /dev/ttyS1 or /dev/ttyUSB0 driver.
7. Line driver of Line 0/2/0 or Line 0/2/1 and driver config on /dev/ttyTun0 are always in sync with each other. Any configuration changes such as baud rate modification is transparently propagated to the Line driver without any additional configuration overhead. This emulates the propagation feature of Serial relay on the IR800 series where the virtual serial port can configure the parameters of the real serial port.

Configuration Commands

```
IR1800#configure terminal
IR1800(config)#interface async 0/2/0
IR1800(config-if)#encapsulation relay-line
IR1800(config-if)#exit
IR1800(config)#relay line 0/2/0 0/0/0
IR1800(config)#exit
IR1800#
```

CHAPTER 28

Cisco SD-WAN Support

This chapter contains the following:

- Cisco SD-WAN Overview, on page 285
- Related Documentation, on page 286

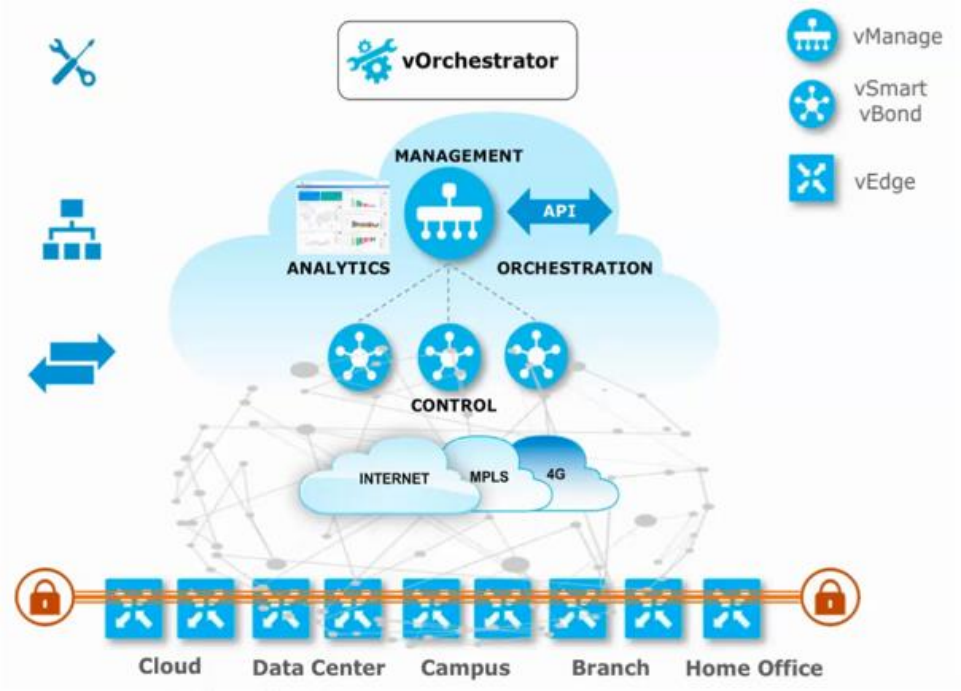
Cisco SD-WAN Overview

Cisco SD-WAN is a cloud-first architecture that separates data and control planes, managed through the Cisco vManage console. You can quickly establish an SD-WAN overlay fabric to connect data centers, branches, campuses, and co-location facilities to improve network speed, security, and efficiency.

Cisco SDWAN adopts a cloud based solution, it consists of vOrchestrator, vManage, vSmart and vEdge.

- vOrchestrator is responsible for launching all controllers VMs in the cloud.
- vManage is the management plane for the overall SDWAN solution. It uses netconf/YANG to talk to vEdge devices.
- vSmart is the control plane for the overall SDWAN solution. It talks to the vEdge device, acts as the route reflector, key reflector, and policy engine.
- vEdge is the data plane of the overall SDWAN solution. The IR1800 platform talks to vSmart, vManage, as part of the SDWAN network.

The follow diagram shows the high level architecture of SDWAN:



While Cisco SD-WAN is a cloud-first architecture, some of the components can be deployed on-premisis. Refer to the Cisco SD-WAN landing page for further information on the capabilities of SD-WAN.

Starting with IOS XE release 17.3.2, the IOS XE image can be configured as controller mode to run SD-WAN. A single universalk9 image is used to deploy Cisco IOS XE SD-WAN and Cisco IOS XE functionality. This universalk9 image supports two modes - Autonomous mode (for Cisco IOS XE features) and Controller mode (for Cisco SD-WAN features).

Access the Cisco IOS XE and Cisco IOS XE SD-WAN functionality through Autonomous and Controller execution modes, respectively. The Autonomous mode is the default mode for the router and includes the Cisco IOS XE functionality. To access Cisco IOS XE SD-WAN functionality, switch to the Controller mode. You can use the existing Plug and Play Workflow to determine the mode of the device. See the Cisco SD-WAN Getting Started Guide for further information.

Note The PnP process works on either Gi0/0/0 or Cellular.

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 29

ROM Monitor Overview

- ROM Monitor Overview, on page 287
- Access ROM Monitor Mode, on page 288
- Displaying the Configuration Register Setting, on page 290
- Environment Variable Settings, on page 290
- Exiting ROM Monitor Mode, on page 292

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 IR1800 platform.

Access ROM Monitor Mode

The following sections describe how to enter the ROMMON mode, and contains the following sections:

Checking the Current ROMMON Version

To display the version of ROMmon running on a router, use the **show rom-monitor** command. To show all variables that are set in ROMmon, use the **show romvar** command.

```
Router#show rom-monitor r0
============================================================

System Bootstrap, Version 3.9(REL), RELEASE SOFTWARE
Copyright (c) 1994-2021 by cisco Systems, Inc.


Router# show romvar
ROMMON variables:
PS1 = rommon ! >
THRPUT =
LICENSE_BOOT_LEVEL =
RET_2_RTS =
DEVICE_MANAGED_MODE = autonomous
BOOT = bootflash:/ir1800-universalk9.17.06.01prd18.SPA.bin,12;
BOOT_STAGED = bootflash:ir1800-universalk9.17.06.01prd18.SPA.bin
BSI = 0
RET_2_RCALTS =
RANDOM_NUM = 231998661
```

```
Router#
```

If your configuration register was set to hex value 0x0 or 0x1820, reload operation will bring you to the ROMmon mode command prompt (rommon 1>). Invoking the set command at the prompt (rommon 1> set) will display the same information as "show romvar" above in IOS/XE exec mode.

```
rommon 1 > set
PS1=rommon ! >
THRPUT=
LICENSE_BOOT_LEVEL=
RET_2_RTS=
DEVICE_MANAGED_MODE=autonomous
BOOT=bootflash:/ir1800-universalk9.17.06.01prd18.SPA.bin,12;
BOOT_STAGED=bootflash:ir1800-universalk9.17.06.01prd18.SPA.bin
BSI=0
RANDOM_NUM=231998661
RET_2_RCALTS=1626821700
```

Commonly Used ROM Monitor Commands

The following table summarizes the commands commonly used in ROM Monitor. For specific instructions on using these commands, refer to the relevant procedure in this document.

Table 20: Commonly Used ROM Monitor Commands

ROMMON Command	Description
boot image	Manually boots a Cisco IOS XE software image.
boot image –o config-file-path	Manually boots the Cisco IOS XE software with a temporary alternative administration configuration file.
confreg	Changes the config-register setting.
dev	Displays the available local storage devices.
dir	Displays the files on a storage device.
reset	Resets the node.
set	Displays the currently set ROM Monitor environmental settings.
sync	Saves the new ROM Monitor environmental settings.
unset	Removes an environmental variable setting.

Examples

The following example shows what appears when you enter the **?** command on a router:

```
rommon 1 > ?
alias               set and display aliases command
boot                boot up an external process
confreg             configuration register utility
dev                 list the device table
dir                 list files in file system
```

```
help                monitor builtin command help
history             monitor command history
meminfo             main memory information
repeat              repeat a monitor command
reset               system reset
set                 display the monitor variables
showmon             display currently selected ROM monitor
sync                write monitor environment to NVRAM
token               display board's unique token identifier
unalias             unset an alias
unset               unset a monitor variable
```

Changing the ROM Monitor Prompt

You can change the prompt in ROM Monitor mode by using the **PS1=** command as shown in the following example:

```
rommon 8 > PS1="IR1800 rommon ! > "
IR1800 rommon 9 >
```

Changing the prompt is useful if you are working with multiple routers in ROM Monitor at the same time. This example specifies that the prompt should be “IR1800 rommon ”, followed by the line number, and then followed by “ > “ by the line number.

Displaying the Configuration Register Setting

To display the current configuration register setting, enter the **confreg** command without parameters as follows:

```
rommon > confreg
Configuration Summary
(Virtual Configuration Register: )
enabled are:
[ 0 ] break/abort has effect
[ 1 ] console baud: 9600
 boot:...... the ROM Monitor
do you wish to change the configuration? y/n  [n]:
```

The configuration register setting is labeled *Virtual Configuration Register*. Enter the **no** command to avoid changing the configuration register setting.

Environment Variable Settings

The ROM Monitor environment variables define the attributes of the ROM Monitor. Environmental variables are entered like commands and are always followed by the equal sign (=). Environment variable settings are entered in capital letters, followed by a definition. For example:

```
IP_ADDRESS=10.0.0.2
```

Under normal operating conditions, you do not need to modify these variables. They are cleared or set only when you need to make changes to the way ROM Monitor operates.

This section includes the following topics:

Frequently Used Environmental Variables

The following table shows the main ROM Monitor environmental variables. For instructions on how to use these variables, see the relevant instructions in this document. The IR1800 boot loader does not support netboot, so any setting like the following environment variables are not used:

- IP_ADDRESS
- IP_SUBNET_MASK
- DEFAULT_GATEWAY
- TFTP_SERVER
- TFTP_FILE

Table 21: Frequently Used ROM Monitor Environmental Variables

Environmental variable	Description
BOOT=path/file	Identifies the boot software for a node. This variable is usually set automatically when the router boots.

Displaying Environment Variable Settings

To display the current environment variable settings, enter the **set** command :

```
rommon 1 > showmon
System Bootstrap, Version 3.9(REL), RELEASE SOFTWARE
Copyright (c) 1994-2021 by cisco Systems, Inc.

IR1833-K9 platform with 4194304 Kbytes of main memory

MCU Version - Bootloader: 22, App: 4D
MCU is in application mode.
```

Entering Environment Variable Settings

Environment variable settings are entered in capital letters, followed by a definition. The following example shows the environmental variables that can be configured in ROMmon mode.:

```
rommon 1 > confreg 0x0
rommon 1> BOOT_WDOG = DISABLE
rommon 1> BOOT = IR1800-K9_image_name
```

Saving Environment Variable Settings

To save the current environment variable settings, enter the **sync** command:

```
rommon > sync
```

Note Environmental values that are not saved with the **sync** command are discarded whenever the system is reset or booted.

Exiting ROM Monitor Mode

To exit ROM Monitor mode, you must change the configuration register and reset the router.

Procedure

	Command or Action	Purpose
Step 1	**confreg** **Example:** `rommon 1> confreg`	Initiates the configuration register configuration prompts.
Step 2	Respond to each prompt as instructed.	See the example that follows this procedure for more information.
Step 3	**reset** **Example:** `rommon 2> reset`	Resets and initializes the router.

Configuration Example

```
rommon 3 > confreg
Configuration Summary
(Virtual Configuration Register: 0x0)
enabled are:
 [ 0 ] break/abort has effect
 [ 1 ] console baud: 9600
 boot: ...... the ROM Monitor
do you wish to change the configuration? y/n  [n]:  y
 enable  "diagnostic mode"? y/n  [n]:
 enable  "use net in IP bcast address"? y/n  [n]:
 enable  "load rom after netboot fails"? y/n  [n]:
 enable  "use all zero broadcast"? y/n  [n]:
 disable "break/abort has effect"? y/n  [n]:
 enable  "ignore system config info"? y/n  [n]:
 change console baud rate? y/n  [n]:
change the boot characteristics? y/n  [n]:
Configuration Summary
(Virtual Configuration Register: 0x0)
enabled are:
 [ 0 ] break/abort has effect
 [ 1 ] console baud: 9600
 boot: ...... the ROM Monitor
do you wish to change the configuration? y/n  [n]:
```

Upgrading the ROMmon for a Router

ROMmon upgrade on the IR1800-K9 router is automatically done when the image is booted. The latest version of the ROMmon is bundled with the IOSXE image. An algorithm detects if the current running version is older than the bundled version, if so, it is automatically upgraded. If the current running version is equal to the bundled version no upgrade is executed. For every successful upgrade, the router is automatically rebooted in order for the new version to get loaded and executed.

Procedure

Step 1 (Optional) Run the **show rom-monitor** *slot* command on the router to see the current release numbers of ROMmon on the hardware. See the Checking the Current ROMMON Version, on page 288 for information about interpreting the output of the command that you run.

Step 2 If autoboot has not been enabled by using the **config-register 0x2102** command, run the **boot** *filesystem:/file-location* command at the ROMmon prompt to boot the Cisco IOS XE image, where *filesystem:/file-location* is the path to the consolidated package file. The ROMmon upgrade is not permanent for any piece of hardware until the Cisco IOS XE image is booted.

Step 3 Run the **enable** command at the user prompt to enter the privileged EXEC mode after the boot is complete.

Step 4 Run the **show rom-monitor** *slot* command to verify whether the ROMmon has been upgraded.

CHAPTER 30

WAN Monitoring

This chapter contains the following topics:

- Information About WANMon, on page 295
- Built-in Recovery Actions, on page 295
- Prerequisites , on page 296
- Guidelines and Limitations, on page 296
- Configuring WANMon, on page 297
- Verifying WANMon Configuration, on page 298
- Configuration Examples, on page 299

Information About WANMon

WANMon is a flexible solution to address the WAN link recovery requirements for the following products and interfaces:

- Physical networks: 4G LTE and Ethernet (WAN port)
- Virtual links: Non-crypto map based IPSec tunnels (either legacy or FlexVPN); that is, any IPSec tunnel you configure as an interface.

You enable WANMon to monitor your WAN links and initiate link recovery actions on receipt of link failure triggers.

Built-in Recovery Actions

The following are the three levels of built-in recovery processes specific to the link type:

Link Type	Recovery Actions		
	Level 0 (Immediate)	**Level 1 (Active)**	**Level 2 (Last-Resort)**
4G LTE	Clear interface, and then shut/no-shut	Module reload	System reload
Ethernet	Clear interface, and then shut/no-shut	No action taken	System reload

Link Type	Recovery Actions		
	Level 0 (Immediate)	Level 1 (Active)	Level 2 (Last-Resort)
Tunnel	Shut/no-shut	No action taken	System reload

Each level has two time-based thresholds based on which built-in recovery actions are taken. The following are the default settings for each level:

- *threshold* is the wait time in minutes after receipt of a link failure trigger to initiate the recovery action as set in the specified level.
- *mintime* is the frequency to perform the recovery action if the link remains down.

The built-in values are:

Level	threshold	mintime	Description
Level 0	10 min	10 min	Triggers Level 0 actions 10 minutes after the link went down. Repeat no more than every 10 minutes.
Level 1	60 min	60 min	Triggers Level 1 actions 10 minutes after the link went down. Repeat no more than every 60 minutes.
Level 2	480 min	60 min	Triggers Level 2 actions 480 minutes after the link went down. Repeat no more than every 60 minutes.

Note If threshold values are specified as 0, no recovery actions are taken for that level. You can use this to avoid system reload (the built-in Level 2 recovery action) on receipt of a link failure trigger where other WAN links may be operational.

Prerequisites

Ensure that the WANMon module is available. The WANMon module is included in the IOS-XE image as the *tm_wanmon.tcl* policy file.

Guidelines and Limitations

- WANMon automatically performs IP address checking (no user configuration) as required for cellular interfaces.
- For all other interfaces, WANMon never performs IP address checking.
- WANMon indirectly triggers user-specified actions by generating an application event that link resetter applets monitor.

- If your network is live, ensure that you understand the potential impact of any command.

Configuring WANMon

You can enable WANMon on the router and assign WAMMon support to specific interfaces. Optionally, you can override the built-in recovery actions, define custom recovery links, and define an event manager environment policy to set the track object value and disable IP address checking. WANMon is disabled by default.

Procedure

	Command or Action	Purpose
Step 1	**event manager policy** *tm_wanmon.tcl* **authorization bypass**	Enables the WANMon link recovery module. Use **authorization bypass** to avoid authorization for CLIs invoked by this policy.
Step 2	**event manager environment wanmon_if_list** *<instance>* {*interface name* {**ipsla** *<instance>*}}	Configures WANMon for the interfaces in your WAN, and indicates that this is an interface configuration command. **Note** Any environment variable with the prefix wanmon_if_list constitutes an interface configuration. Multiple interfaces are allowed by specifying an instance. Be sure to specify the full interface name (for example, cellular0/4/0 or cellular0/5/0). You can set the IP SLA icmp-echo trigger, if desired. Multiple IP SLA triggers are allowed by specifing an instance. **Note** WANMon only looks at the status of the SLA ID. Even though *icmp-echo* is most common, if needed any other type of SLA probe (for example, *udp-echo*) can be used instead.
Step 3	**event manager environment wanmon_if_listx** {*interface name* {**recovery** *Level0* {*Level1* } *Level2*}}	(Optional) Overrides the built-in thresholds.
Step 4	**publish-event sub-system 798 type 2000 arg1** *<interface name>* **arg2** *<level >*	(Optional) Configures custom recovery actions using link resetter applets. *<interface >* is the full interface name (for example, cellular0/4/0 or cellular0/5/0). *<level >* is 0, 1, or 2 to match the desired link recovery action.

	Command or Action	Purpose
Step 5	**{stub** <*track-stub-id* > **}**	(Optional) Allows an event manager environment policy to set the track object value. WANMon can set a track-stub-object value to reflect the link state so that an external applet can track the stub object.
Step 6	**event manager environment wanmon_if_listx** {<*interface name* > {**checkip** <*instance* >}}	(Optional) Disables IP address checking.

What to do next

EXAMPLES

```
event manager policy tm_wanmon.tcl authorization bypass
```

The following examples are Event Manager commands to configure cellular and Ethernet interfaces:

```
event manager environment wanmon_if_list1 {cellular0/4/0 {ipsla 1}}
event manager environment wanmon_if_list2 {GigabitEthernet0/0/0 {ipsla 2}}
```

This example sets custom recovery thresholds:

```
event manager environment wanmon_if_list {cellular0/4/0 {recovery 20 {90 75} 600}
```

Where:

- The Level 0 threshold is set to 20 minutes after the link failure trigger. Level 0 recovery actions are performed for the cellular interface. Repeats indefinitely, no more than every 10 minutes (default).
- Level 1 threshold is set to 90 minutes. Level 1 recovery actions are performed for the cellular interface. Repeats no more frequently than every 75 minutes.
- The Level 2 threshold is set to 600 minutes (10 hours).

The following sets the track-stub-object value to 21:

```
conf t
track 21 stub-object
event manager environment wanmon_if_list {cellular0/4/0 {ipsla 1} {stub 21}
```

Verifying WANMon Configuration

Use the following steps to verify your WANMon configuraion.

Procedure

	Command or Action	Purpose
Step 1	**show event manager policy registered**	Displays the WAN monitoring policy.
Step 2	**show event manager environment**	Displays the interface environment variables set during interface configuration.

What to do next

EXAMPLE

```
show event manager policy registered
1    script    system  multiple   Off   Thu Jan 16 18:44:29 2014  tm_wanmon.tcl

show event manager environment
1 wanmon_if_list   {cell0/4/0 {ipsla 1}}
```

Configuration Examples

The following examples are provided:

WANMon Cellular Interface Configuration Example

```
track 1 ip sla 1
ip sla 1
 icmp-echo 172.27.166.250
 timeout 6000
 frequency 300
ip sla schedule 1 life forever start-time now
event manager environment wanmon_if_list {cellular0/4/0 {ipsla 1}}
event manager policy tm_wanmon.tcl authorization bypass
```

Multiple WAN Link Monitoring Example

```
track 1 ip sla 1
track 21 stub-object
ip sla 1
 icmp-echo 172.27.166.250
 timeout 6000
 frequency 300
ip sla schedule 1 life forever start-time now
track 2 ip sla 2
track 22 stub-object
ip sla 2
 icmp-echo 10.27.16.25
 timeout 6000
 frequency 300
ip sla schedule 2 life forever start-time now
event manager environment wanmon_if_list1 {cellular0/4/0 {ipsla 1} {stub 21}}
event manager policy tm_wanmon.tcl authorization bypass
```

CHAPTER 31

Yang Data Models

This chapter contains the following:

- Support for YANG Data Models (Call-home), on page 301
- Yang Data Model Support for Raw Socket Transport, on page 301
- Yang Data Model Support for Scada, on page 302

Support for YANG Data Models (Call-home)

The YANG models supported for the call-home feature are similar to the earlier releases of Cisco-IOS-XE, and the same is supported on the release of IOS-XE on IR1800. The following references are available for earlier YANG models:

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

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

Software Activation Configuration Guide, Cisco IOS XE Release 3S

Yang Data Model Support for Raw Socket Transport

Release 17.2.1 adds support for additional Yang Data Models. These additional models include Raw Socket Transport.

Yang Data Models can be found here:

https://github.com/YangModels/yang/tree/master/vendor/cisco/xe/1721

There are two feature modules available for raw socket that belong to the main Cisco-IOS-XE-native model. They are:

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

The Cisco-IOS-XE-rawsocket-oper.yang module contains a collection of YANG definitions for Raw Socket Transport Configuration commands.

The module has the following corresponding CLI commands:

```
# encapsulation raw-tcp
# encapsulation raw-udp
# raw-socket packet-length <length>
# raw-socket packet-timer <timer>
# raw-socket special-char <value>
# raw-socket tcp server <port> <ip>
# raw-socket tcp idle-timeout <value>
# raw-socket tcp client <dest-ip> <dest-port>
# raw-socket tcp idle-timeout <timeout>
# raw-socket tcp tcp-session <value>
# raw-socket tcp dscp <value>
# raw-socket udp connection <dest-ip> <dest-port> <local_port>
```

The Cisco-IOS-XE-rawsocket-oper.yang module contains a collection of YANG definitions for Raw Socket Transport operational data.

The module has the following corresponding CLI commands:

```
# show raw udp statistics
# show raw tcp statistics
# show raw tcp session
# show raw udp session
# show raw tcp session local
# show raw udp session local
```

The following is a list of the Dependent Modules:

- Cisco-IOS-XE-native
- Cisco-IOS-XE-features
- ietf-inet-types
- Cisco-IOS-XE-interfaces
- Cisco-IOS-XE-ip
- Cisco-IOS-XE-vlan
- ietf-yang-types @ (any revision)
- cisco-semver

Yang Data Model Support for Scada

The Cisco IOS XE 17.1.1 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 — Contains a collection of YANG definitions for Scada Gateway Configuration commands.

- Cisco-IOS-XE-scada-gw-oper.yang — Contains a collection of YANG definitions for Scada Gateway operational data.

There are 8 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)# sector <sector_name>
(config-t101)# attach-to-session <session-name>
(config-t101)# asdu-addr <addr>

(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

CHAPTER 32

Connected Grid NMS Agent (CGNA) Support

This chapter contains the following sections:

- Connected Grid NMS Agent (CGNA) Support, on page 305
- Overview, on page 305
- WebSocket Support, on page 306

Connected Grid NMS Agent (CGNA) Support

CGNA is a subsystem running on IOS which provides connectivity and manageability with network management system like FND and IOT Operations Dashboard. Complete documentation sets for IoT Field Network Director and IoT Industrial Network Director can be found in the following links:

https://www.cisco.com/c/en/us/support/cloud-systems-management/iot-field-network-director/series.html

https://www.cisco.com/c/en/us/support/cloud-systems-management/industrial-network-director/series.html#%7Etab-documents

Overview

The majority of features are implemented on the IR1800 including basic profile, execution profile, heartbeat profile, and firmware download. Some features like bspatch will not be imported on the IR1800 due to the limitation on the IOS image.

CGNA has four different profiles for different use cases. They are:

- Profile - This is the basic profile. It can be configured to support tunnel provision, registration, and periodic update.
- Execution profile - This can be configured to execute CLI commands
- Heartbeat profile - This can be configured to send heartbeat messages
- Transport profile - This can be configured to provide different transport protocol for the basic profile to use. Currently only supports WebSocket.

CGNA also provides CLIs for resumable image retrieval from network management system and simple SNMP server for redirecting events to WebSocket connection.

WebSocket Support

Traditional IOS provided the WebSocket service in the IOSd kernel. Polaris IOS-XE moves the WebSocket service out of the IOSd kernel for improved performance. The websocket version will be upgraded to 3.2.3.

CHAPTER 33

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 307
- Monitoring Hardware Using Alarms, on page 316
- Network Management System Alerts a Network Administrator when an Alarm is Reported Through SNMP, on page 317

Monitoring Control Plane Resources

The following sections explain the details of memory and CPU monitoring from the perspective of the Cisco IOS process and the overall control plane:

Avoiding Problems Through Regular Monitoring

Processes should provide monitoring and notification of their status/health to ensure correct operation. When a process fails, a syslog error message is displayed and either the process is restarted or the router is rebooted. A syslog error message is displayed when a monitor detects that a process is stuck or has crashed. If the process can be restarted, it is restarted; else, the router is restarted.

Monitoring system resources enables you to detect potential problems before they occur, thus avoiding outages. It also establishes a baseline for a normal system load. You can use this information as a basis for comparison, when you upgrade hardware or software to see if the upgrade has affected resource usage.

Cisco IOS Process Resources

You can view CPU utilization statistics on active processes and see the amount of memory being used in these processes using the **show memory** command and the **show process cpu** command. These commands provide a representation of memory and CPU utilization from the perspective of only the Cisco IOS process; they do not include information for resources on the entire platform. When the **show memory** command is used in a system with 4 GB RAM running a single Cisco IOS process, the following memory usage is displayed:

```
Router# show memory
Tracekey : 1#b93c0f1c0d5d16ddc3ab8e54342a8dd5

                Head     Total(b)     Used(b)     Free(b)   Lowest(b)  Largest(b)
Processor  7F5F358048   1997625144   290200588   1707424556   487419128   1509949348
```

```
reserve P  7F5F3580A0      102404          92      102312      102312      102312
lsmpi_io  7F5D9901A8     6295128     6294304         824         824         412
Dynamic heap limit(MB) 1440      Use(MB) 0

          Processor memory

Address      Bytes      Prev       Next    Ref   PrevF     NextF          what
   Alloc PC
7F5F358048 0000102408 00000000 7F5F3710A8 001  -------- -------- *Init*
:555F22A000+5E5691C
7F5F3710A8 0000000056 7F5F358048 7F5F371138 001  -------- -------- *Init*
:555F22A000+5E56938
7F5F371138 0000008224 7F5F3710A8 7F5F3731B0 001  -------- -------- *Init*
:555F22A000+5E56958
7F5F3731B0 0000000296 7F5F371138 7F5F373330 001  -------- -------- *Init*
:555F22A000+86B9178
7F5F373330 0000000568 7F5F3731B0 7F5F3735C0 001  -------- -------- *Init*
:555F22A000+86BEE5C
7F5F3735C0 0000032776 7F5F373330 7F5F37B620 001  -------- -------- Managed Chunk Q
:555F22A000+86AAC38
7F5F37B620 0000000056 7F5F3735C0 7F5F37B6B0 001  -------- -------- *Init*
:555F22A000+5EA29DC
7F5F37B6B0 0000032776 7F5F37B620 7F5F383710 001  -------- -------- Queue Pair - Q
:555F22A000+86D3A28
7F5F383710 0000012808 7F5F37B6B0 7F5F386970 001  -------- -------- *Init*
:555F22A000+11EFF930
7F5F386970 0000032776 7F5F383710 7F5F38E9D0 001  -------- -------- List Elements
:555F22A000+86798AC
7F5F38E9D0 0000032776 7F5F386970 7F5F396A30 001  -------- -------- List Headers
:555F22A000+86798EC
7F5F396A30 0000032776 7F5F38E9D0 7F5F39EA90 001  -------- -------- IOSXE Process S
:555F22A000+984FEE0
7F5F39EA90 0000032776 7F5F396A30 7F5F3A6AF0 001  -------- -------- IOSXE Queue Pro
:555F22A000+984FF24
7F5F3A6AF0 0000065544 7F5F39EA90 7F5F3B6B50 001  -------- -------- IOSXE Queue Bal
:555F22A000+984FF68
7F5F3B6B50 0000000328 7F5F3A6AF0 7F5F3B6CF0 001  -------- -------- *Init*
:555F22A000+11EF88AC
7F5F3B6CF0 0000000328 7F5F3B6B50 7F5F3B6E90 001  -------- -------- *Init*
:555F22A000+11EF88AC
7F5F3B6E90 0000000192 7F5F3B6CF0 7F5F3B6FA8 001  -------- -------- SDB String
:555F22A000+8629F60
7F5F3B6FA8 0000036872 7F5F3B6E90 7F5F3C0008 001  -------- -------- *Init*
:555F22A000+98482F4
7F5F3C0008 0000010008 7F5F3B6FA8 7F5F3C2778 001  -------- -------- Platform VM Pag
:555F22A000+986FC68
7F5F3C2778 0000002008 7F5F3C0008 7F5F3C2FA8 001  -------- -------- *Init*
iosd_crb_crankshaft_unix:7F83050000+6D850
7F5F3C2FA8 0000200712 7F5F3C2778 7F5F3F4008 001  -------- -------- Interrupt Stack
:555F22A000+98482F4
7F5F3F4008 0000000328 7F5F3C2FA8 7F5F3F41A8 001  -------- -------- *Init*
:555F22A000+11EF88AC
7F5F3F41A8 0000002008 7F5F3F4008 7F5F3F49D8 001  -------- -------- Watcher Message
:555F22A000+86DE21C
7F5F3F49D8 0000000360 7F5F3F41A8 7F5F3F4B98 001  -------- -------- Process Events
:555F22A000+86D94A4
7F5F3F4B98 0000000328 7F5F3F49D8 7F5F3F4D38 001  -------- -------- *Init*
:555F22A000+11EF88AC
7F5F3F4D38 0000000184 7F5F3F4B98 7F5F3F4E48 001  -------- -------- *Init*
:555F22A000+86C945C
7F5F3F4E48 0000000264 7F5F3F4D38 7F5F3F4FA8 001  -------- -------- *Init*
:555F22A000+86C945C
7F5F3F4FA8 0000036872 7F5F3F4E48 7F5F3FE008 001  -------- -------- *Init*
:555F22A000+98482F4
```

```
7F5F3FE008 0000000328 7F5F3F4FA8 7F5F3FE1A8 001  -------- -------- *Init*
:555F22A000+11EF88AC
7F5F3FE1A8 0000001504 7F5F3FE008 7F5F3FE7E0 001  -------- -------- Reg Function Se
:555F22A000+868AE50
7F5F3FE7E0 0000001504 7F5F3FE1A8 7F5F3FEE18 001  -------- -------- Reg Function Se
:555F22A000+868AEE0
7F5F3FEE18 0000000064 7F5F3FE7E0 7F5F3FEEB0 001  -------- -------- Parser Linkage
:555F22A000+5D98838
7F5F3FEEB0 0000000160 7F5F3FEE18 7F5F3FEFA8 001  -------- -------- *Init*
:555F22A000+530A17C
7F5F3FEFA8 0000036872 7F5F3FEEB0 7F5F408008 001  -------- -------- *Init*
:555F22A000+98482F4
7F5F408008 0000000328 7F5F3FEFA8 7F5F4081A8 001  -------- -------- *Init*
:555F22A000+11EF88AC
7F5F4081A8 0000000760 7F5F408008 7F5F4084F8 001  -------- -------- *Init*
:555F22A000+868BC18
7F5F4084F8 0000000576 7F5F4081A8 7F5F408790 001  -------- -------- *Init*
:555F22A000+868BC18
7F5F408790 0000000400 7F5F4084F8 7F5F408978 001  -------- -------- *Init*
:555F22A000+868BC18
7F5F408978 0000000488 7F5F408790 7F5F408BB8 001  -------- -------- *Init*
:555F22A000+868BC18
7F5F408BB8 0000000920 7F5F408978 7F5F408FA8 001  -------- -------- *Init*
:555F22A000+868BC18
7F5F408FA8 0000200712 7F5F408BB8 7F5F43A008 001  -------- -------- Interrupt Stack
:555F22A000+98482F4
7F5F43A008 0000000968 7F5F408FA8 7F5F43A428 001  -------- -------- *Init*
iosd_crb_crankshaft_unix:7F83050000+378C0
7F5F43A428 0000000280 7F5F43A008 7F5F43A598 001  -------- -------- *Init*
:555F22A000+A3CE294
7F5F43A598 0000000896 7F5F43A428 7F5F43A970 001  -------- -------- Watched Message
:555F22A000+86DE1E8
7F5F43A970 0000001320 7F5F43A598 7F5F43AEF0 001  -------- -------- Process
:555F22A000+86E50BC
7F5F43AEF0 0000000096 7F5F43A970 7F5F43AFA8 001  -------- -------- *Init*
:555F22A000+868BC18
7F5F43AFA8 0000036872 7F5F43AEF0 7F5F444008 001  -------- -------- *Init*
:555F22A000+98482F4
7F5F444008 0000003008 7F5F43AFA8 7F5F444C20 001  -------- -------- Watched Semapho
:555F22A000+86DE180
7F5F444C20 0000000360 7F5F444008 7F5F444DE0 001  -------- -------- Process Events
:555F22A000+86D94A4
7F5F444DE0 0000000368 7F5F444C20 7F5F444FA8 001  -------- -------- *Init*
:555F22A000+11EF88AC
7F5F444FA8 0000200712 7F5F444DE0 7F5F476008 001  -------- -------- Interrupt Stack
:555F22A000+98482F4
7F5F476008 0000002336 7F5F444FA8 7F5F476980 001  -------- -------- Process Array
:555F22A000+86E4F94
7F5F476980 0000000184 7F5F476008 7F5F476A90 001  -------- -------- *Init*
:555F22A000+86C945C
7F5F476A90 0000000184 7F5F476980 7F5F476BA0 001  -------- -------- *Init*
:555F22A000+86C945C
7F5F476BA0 0000000184 7F5F476A90 7F5F476CB0 001  -------- -------- *Init*
:555F22A000+86C945C
7F5F476CB0 0000000184 7F5F476BA0 7F5F476DC0 001  -------- -------- *Init*
:555F22A000+86C945C
7F5F476DC0 0000000184 7F5F476CB0 7F5F476ED0 001  -------- -------- *Init*
:555F22A000+86C945C
7F5F476ED0 0000000128 7F5F476DC0 7F5F476FA8 001  -------- -------- *Init*
:555F22A000+868BC18
7F5F476FA8 0000036872 7F5F476ED0 7F5F480008 001  -------- -------- *Init*
:555F22A000+98482F4
7F5F480008 0000001320 7F5F476FA8 7F5F480588 001  -------- -------- Process
:555F22A000+86E50BC
```

```
7F5F480588 0000000184 7F5F480008 7F5F480698 001  -------- -------- *Init*
:555F22A000+86C945C
7F5F480698 0000000184 7F5F480588 7F5F4807A8 001  -------- -------- *Init*
:555F22A000+86C945C
7F5F4807A8 0000000184 7F5F480698 7F5F4808B8 001  -------- -------- *Init*
:555F22A000+86C945C
7F5F4808B8 0000000184 7F5F4807A8 7F5F4809C8 001  -------- -------- *Init*
:555F22A000+86C945C
7F5F4809C8 0000000184 7F5F4808B8 7F5F480AD8 001  -------- -------- *Init*
:555F22A000+86C945C
7F5F480AD8 0000000184 7F5F4809C8 7F5F480BE8 001  -------- -------- *Init*
:555F22A000+86C945C
7F5F480BE8 0000000184 7F5F480AD8 7F5F480CF8 001  -------- -------- *Init*
:555F22A000+86C945C
7F5F480CF8 0000000096 7F5F480BE8 7F5F480DB0 001  -------- -------- *Init*
:555F22A000+86C940C
7F5F480DB0 0000000096 7F5F480CF8 7F5F480E68 001  -------- -------- Init
:555F22A000+862A110
7F5F480E68 0000000232 7F5F480DB0 7F5F480FA8 001  -------- -------- *Init*
:555F22A000+60E2660
7F5F480FA8 0000200712 7F5F480E68 7F5F4B2008 001  -------- -------- Interrupt Stack
:555F22A000+98482F4
7F5F4B2008 0000003008 7F5F480FA8 7F5F4B2C20 001  -------- -------- Reg Function Li
:555F22A000+868AE80
7F5F4B2C20 0000000064 7F5F4B2008 7F5F4B2CB8 001  -------- -------- Parser Linkage
:555F22A000+5D98838
7F5F4B2CB8 0000000064 7F5F4B2C20 7F5F4B2D50 001  -------- -------- Parser Linkage
:555F22A000+5D98A78
7F5F4B2D50 0000000080 7F5F4B2CB8 7F5F4B2DF8 001  -------- -------- Init
:555F22A000+5E87AC0
7F5F4B2DF8 0000000200 7F5F4B2D50 7F5F4B2F18 001  -------- -------- Init
:555F22A000+5E87AC0
7F5F4B2F18 0000000056 7F5F4B2DF8 7F5F4B2FA8 001  -------- -------- Init
:555F22A000+54DD60C
7F5F4B2FA8 0000036872 7F5F4B2F18 7F5F4BC008 001  -------- -------- *Init*
:555F22A000+98482F4
7F5F4BC008 0000001504 7F5F4B2FA8 7F5F4BC640 001  -------- -------- Reg Function Ca
:555F22A000+868AF10
7F5F4BC640 0000000224 7F5F4BC008 7F5F4BC778 001  -------- -------- *Init*
:555F22A000+868BC18
7F5F4BC778 0000000224 7F5F4BC640 7F5F4BC8B0 001  -------- -------- *Init*
:555F22A000+868BC18
7F5F4BC8B0 0000000328 7F5F4BC778 7F5F4BCA50 001  -------- -------- *Init*
:555F22A000+868BC18
7F5F4BCA50 0000000328 7F5F4BC8B0 7F5F4BCBF0 001  -------- -------- *Init*
:555F22A000+868BC18
7F5F4BCBF0 0000000328 7F5F4BCA50 7F5F4BCD90 001  -------- -------- *Init*
:555F22A000+868BC18
7F5F4BCD90 0000000216 7F5F4BCBF0 7F5F4BCEC0 001  -------- -------- Init
:555F22A000+5E87AC0
7F5F4BCEC0 0000000144 7F5F4BCD90 7F5F4BCFA8 001  -------- -------- Init
:555F22A000+530F7A4
7F5F4BCFA8 0000200712 7F5F4BCEC0 7F5F4EE008 001  -------- -------- Interrupt Stack
:555F22A000+98482F4
7F5F4EE008 0000006888 7F5F4BCFA8 7F5F4EFB48 001  -------- -------- TTY data
:555F22A000+85C9E44
7F5F4EFB48 0000004104 7F5F4EE008 7F5F4F0BA8 001  -------- -------- TTY Input Buf
:555F22A000+85CBD60
7F5F4F0BA8 0000004104 7F5F4EFB48 7F5F4F1C08 001  -------- -------- TTY Output Buf
:555F22A000+85CBDB0
7F5F4F1C08 0000024584 7F5F4F0BA8 7F5F4F7C68 001  -------- -------- proc_hist_lmt_v
:555F22A000+BA3B718
7F5F4F7C68 0000008200 7F5F4F1C08 7F5F4F9CC8 001  -------- -------- proc_hist_lmt_v
:555F22A000+BA3B74C
```

```
7F5F4F9CC8 0000008200 7F5F4F7C68 7F5F4FBD28 001  -------- -------- proc_hist_lmt_v
:555F22A000+BA3B784
7F5F4FBD28 0000005008 7F5F4F9CC8 7F5F4FD110 001  -------- -------- messages
:555F22A000+86DE040
7F5F4FD110 0000005008 7F5F4FBD28 7F5F4FE4F8 001  -------- -------- Watched message
:555F22A000+86DE078
7F5F4FE4F8 0000020008 7F5F4FD110 7F5F503378 001  -------- -------- Watched Queue
:555F22A000+86DE0AC
7F5F503378 0000065544 7F5F4FE4F8 7F5F5133D8 001  -------- -------- Watched Queue I
:555F22A000+86DE0E4
7F5F5133D8 0000020008 7F5F503378 7F5F518258 001  -------- -------- Watched Boolean
:555F22A000+86DE118
7F5F518258 0000020008 7F5F5133D8 7F5F51D0D8 001  -------- -------- Watched Bitfiel
:555F22A000+86DE14C
7F5F51D0D8 0000010008 7F5F518258 7F5F51F848 001  -------- -------- Watcher Info
:555F22A000+86DE1B4
7F5F51F848 0000010008 7F5F51D0D8 7F5F521FB8 001  -------- -------- Read/Write Lock
:555F22A000+86DE250
7F5F521FB8 0000001232 7F5F51F848 7F5F5224E0 001  -------- -------- *Init*
:555F22A000+868BC18
7F5F5224E0 0000000064 7F5F521FB8 7F5F522578 001  -------- -------- Init
:555F22A000+8D8F3A0
7F5F522578 0000002008 7F5F5224E0 7F5F522DA8 001  -------- -------- Injected msg CB
:555F22A000+ACBBF08
7F5F522DA8 0000000064 7F5F522578 7F5F522E40 001  -------- -------- SDB String
:555F22A000+8629F60
7F5F522E40 0000000056 7F5F522DA8 7F5F522ED0 001  -------- -------- *Init*
:555F22A000+5EA29DC
7F5F522ED0 0000000128 7F5F522E40 7F5F522FA8 001  -------- -------- XOS_MEM_XDT
:555F22A000+894FE1C
7F5F522FA8 0000028104 7F5F522ED0 7F5F529DC8 001  -------- -------- Process Stack
:555F22A000+98482F4
7F5F529DC8 0000000096 7F5F522FA8 7F5F529E80 001  -------- -------- Init
:555F22A000+862A110
7F5F529E80 0000000208 7F5F529DC8 7F5F529FA8 001  -------- -------- *Init*
:555F22A000+86C8D58
7F5F529FA8 0000016104 7F5F529E80 7F5F52DEE8 001  -------- -------- Process Stack
:555F22A000+98482F4
7F5F52DEE8 0000032776 7F5F529FA8 7F5F535F48 001  -------- -------- List Elements
:555F22A000+8679DCC
7F5F535F48 0000032776 7F5F52DEE8 7F5F53DFA8 001  -------- -------- List Elements
:555F22A000+8679DCC
7F5F53DFA8 0000032776 7F5F535F48 7F5F546008 001  -------- -------- List Elements
:555F22A000+8679DCC
7F5F546008 0000032776 7F5F53DFA8 7F5F54E068 001  -------- -------- List Elements
:555F22A000+8679DCC
7F5F54E068 0000032776 7F5F546008 7F5F5560C8 001  -------- -------- List Elements
:555F22A000+8679DCC
7F5F5560C8 0000032776 7F5F54E068 7F5F55E128 001  -------- -------- List Elements
:555F22A000+8679DCC
7F5F55E128 0000032776 7F5F5560C8 7F5F566188 001  -------- -------- List Elements
:555F22A000+8679DCC
7F5F566188 0000032776 7F5F55E128 7F5F56E1E8 001  -------- -------- List Elements
:555F22A000+8679DCC
7F5F56E1E8 0000005008 7F5F566188 7F5F56F5D0 001  -------- -------- Reg Function 12
:555F22A000+868AE1C
7F5F56F5D0 0000020008 7F5F56E1E8 7F5F574450 001  -------- -------- Subsys Malloc I
:555F22A000+86875C8
7F5F574450 0000001176 7F5F56F5D0 7F5F574940 001  -------- -------- SPA variable ms
:555F22A000+B9B3700
7F5F574940 0000000920 7F5F574450 7F5F574D30 001  -------- -------- SAMsgThread
:555F22A000+54E49DC
7F5F574D30 0000000064 7F5F574940 7F5F574DC8 001  -------- -------- Parser Linkage
:555F22A000+5D98A78
```

```
7F5F574DC8 0000000064 7F5F574D30 7F5F574E60 001  -------- -------- Parser Linkage
:555F22A000+5D98838
7F5F574E60 0000000064 7F5F574DC8 7F5F574EF8 001  -------- -------- Parser Linkage
:555F22A000+5D98A78
7F5F574EF8 0000000088 7F5F574E60 7F5F574FA8 001  -------- -------- Init
:555F22A000+54DD60C
7F5F574FA8 0000000968 7F5F574EF8 7F5F5753C8 001  -------- -------- Crypto CA
:555F22A000+8DC26C0
7F5F5753C8 0000000216 7F5F574FA8 7F5F5754F8 001  -------- -------- Crypto CA
:555F22A000+8DC2588
7F5F5754F8 0000002648 7F5F5753C8 7F5F575FA8 000  7F609C3C28 7F69C82D38 (coalesced)
:555F22A000+52BC2B8
7F5F575FA8 0000028104 7F5F5754F8 7F5F57CDC8 001  -------- -------- Process Stack
:555F22A000+98482F4
7F5F57CDC8 0000013112 7F5F575FA8 7F5F580158 001  -------- -------- SAMsgThread
:555F22A000+5360944
7F5F580158 0000004728 7F5F57CDC8 7F5F581428 001  -------- -------- *Packet Data*
:555F22A000+AF448CC
7F5F581428 0000000968 7F5F580158 7F5F581848 001  -------- -------- Exec
:555F22A000+5D9C004
7F5F581848 0000000600 7F5F581428 7F5F581AF8 001  -------- -------- Ether OAM subbl
:555F22A000+962A100
7F5F581AF8 0000000056 7F5F581848 7F5F581B88 000  7F69F8B620 7F6A906D40 (fragment)
:555F22A000+962A100
7F5F581B88 0000005008 7F5F581AF8 7F5F582F70 001  -------- -------- Reg Function iL
:555F22A000+868AEB0
7F5F582F70 0000065544 7F5F581B88 7F5F592FD0 001  -------- -------- Registry Call S
:555F22A000+8690EFC
7F5F592FD0 0000002584 7F5F582F70 7F5F593A40 001  -------- -------- *Init*
:555F22A000+868BC18
7F5F593A40 0000002080 7F5F592FD0 7F5F5942B8 001  -------- -------- *Init*
:555F22A000+ACB41D8
7F5F5942B8 0000002600 7F5F593A40 7F5F594D38 001  -------- -------- *Init*
:555F22A000+ACB41D8
7F5F594D38 0000020008 7F5F5942B8 7F5F599BB8 001  -------- -------- Peer uid cb chu
:555F22A000+ACBBE98
```

The **show process cpu** command displays Cisco IOS CPU utilization average:

```
Router# show process cpu
CPU utilization for five seconds: 0%/0%; one minute: 0%; five minutes: 0%
PID Runtime(ms)     Invoked      uSecs   5Sec   1Min   5Min TTY Process
   1           8          31        258  0.00%  0.00%  0.00%   0 Chunk Manager
   2           8       10141          0  0.00%  0.00%  0.00%   0 Load Meter
   3           0           1          0  0.00%  0.00%  0.00%   0 PKI Trustpool
   4           0           1          0  0.00%  0.00%  0.00%   0 Retransmission o
   5           0           1          0  0.00%  0.00%  0.00%   0 IPC ISSU Dispatc
   6          28          13       2153  0.00%  0.00%  0.00%   0 RF Slave Main Th
   7           0           1          0  0.00%  0.00%  0.00%   0 EDDRI_MAIN
   8           0           1          0  0.00%  0.00%  0.00%   0 RO Notify Timers
   9       40648        7844       5182  0.00%  0.06%  0.05%   0 Check heaps
  10          16         845         18  0.00%  0.00%  0.00%   0 Pool Manager
  11           0           1          0  0.00%  0.00%  0.00%   0 DiscardQ Backgro
  12           0           2          0  0.00%  0.00%  0.00%   0 Timers
  13           0         176          0  0.00%  0.00%  0.00%   0 WATCH_AFS
  14           0           1          0  0.00%  0.00%  0.00%   0 MEMLEAK PROCESS
  15           0           1          0  0.00%  0.00%  0.00%   0 ARP Input
  16           4       52892          0  0.00%  0.00%  0.00%   0 ARP Background
  17           0           2          0  0.00%  0.00%  0.00%   0 ATM Idle Timer
  18           0           1          0  0.00%  0.00%  0.00%   0 ATM ASYNC PROC
  19           0           1          0  0.00%  0.00%  0.00%   0 CEF MIB API
```

```
  20          0          1          0  0.00%  0.00%  0.00%   0 AAA_SERVER_DEADT
  21          0          1          0  0.00%  0.00%  0.00%   0 Policy Manager
  22          0          2          0  0.00%  0.00%  0.00%   0 DDR Timers
  23         16         10       1600  0.00%  0.00%  0.00%   0 Entity MIB API
  24        120         27       4444  0.00%  0.00%  0.00%   0 PrstVbl
  25          0          2          0  0.00%  0.00%  0.00%   0 Serial Backgroun
  26          0          1          0  0.00%  0.00%  0.00%   0 RMI RM Notify Wa
  27          0          2          0  0.00%  0.00%  0.00%   0 ATM AutoVC Perio
  28          0          2          0  0.00%  0.00%  0.00%   0 ATM VC Auto Crea
  29          4      25354          0  0.00%  0.00%  0.00%   0 IOSXE heartbeat
  30          0         86          0  0.00%  0.00%  0.00%   0 Btrace time base
  31          0         10          0  0.00%  0.00%  0.00%   0 DB Lock Manager
  32          4      50697          0  0.00%  0.00%  0.00%   0 GraphIt
  33          0          1          0  0.00%  0.00%  0.00%   0 DB Notification
  34          0          1          0  0.00%  0.00%  0.00%   0 IPC Apps Task
  35          0          1          0  0.00%  0.00%  0.00%   0 ifIndex Receive
  36          0      10142          0  0.00%  0.00%  0.00%   0 IPC Event Notifi
  37          0      49518          0  0.00%  0.00%  0.00%   0 IPC Mcast Pendin
  38          0          1          0  0.00%  0.00%  0.00%   0 Platform appsess
  39          0        846          0  0.00%  0.00%  0.00%   0 IPC Dynamic Cach
  40          0      10142          0  0.00%  0.00%  0.00%   0 IPC Service NonC
  41          0          1          0  0.00%  0.00%  0.00%   0 IPC Zone Manager
  42          8      49518          0  0.00%  0.00%  0.00%   0 IPC Periodic Tim
  43          0      49518          0  0.00%  0.00%  0.00%   0 IPC Deferred Por
  44          0          1          0  0.00%  0.00%  0.00%   0 IPC Process leve
  45          0          1          0  0.00%  0.00%  0.00%   0 IPC Seat Manager
```

show process cpu platform sorted

```
CPU utilization for five seconds:  9%, one minute: 10%, five minutes: 10%
Core 0: CPU utilization for five seconds:  3%, one minute:  4%, five minutes:  4%
Core 1: CPU utilization for five seconds:  4%, one minute:  4%, five minutes:  4%
Core 2: CPU utilization for five seconds:  2%, one minute:  2%, five minutes:  2%
Core 3: CPU utilization for five seconds: 38%, one minute: 38%, five minutes: 38%
   Pid    PPid    5Sec    1Min    5Min  Status        Size  Name
-------------------------------------------------------------------------------
18700   18679     44%     44%     44%  S           235192  qfp-ucode-avent
  5226    5216      2%      3%      3%  S           697124  linux_iosd-imag
24238   24231      1%      1%      1%  S             8288  ngiolite
18412   18398      1%      1%      1%  S           135696  fman_fp_image
30574       2      0%      0%      0%  S                0  kworker/0:3
24231   16366      0%      0%      0%  S             2460  pman
24025   23998      0%      0%      0%  S             3392  nginx
24024   23998      0%      0%      0%  S             4300  nginx
23998   23990      0%      0%      0%  S             7944  nginx
23990    4251      0%      0%      0%  S             2460  pman
23605   23599      0%      0%      0%  S             7988  ngiolite
23599   16366      0%      0%      0%  S             2464  pman
23330   23309      0%      0%      0%  S            39600  iomd
23309   16366      0%      0%      0%  S             2460  pman
21981   15002      0%      0%      0%  S              416  sleep
21935   21906      0%      0%      0%  S            38680  iomd
21906   16366      0%      0%      0%  S             2460  pman
21830   13884      0%      0%      0%  S              416  sleep
21694       2      0%      0%      0%  S                0  kworker/0:0
21042       2      0%      0%      0%  S                0  kworker/u8:4
21041       2      0%      0%      0%  S                0  kworker/u8:3
21040       2      0%      0%      0%  S                0  kworker/u8:0
20737       2      0%      0%      0%  S                0  kworker/1:3
20731       2      0%      0%      0%  S                0  SarIosdMond
20574   20548      0%      0%      0%  S            12004  btman
20548   16921      0%      0%      0%  S             2432  pman
20180   20146      0%      0%      0%  S            17428  cman_fp
20146   16921      0%      0%      0%  S             2432  pman
20135   20105      0%      0%      0%  S            12228  btman
```

```
20105   16366      0%      0%      0%  S             2432  pman
20093       2      0%      0%      0%  S                0  kworker/0:1
19819   19796      0%      0%      0%  S           107992  cpp_cp_svr
19796   16921      0%      0%      0%  S             2436  pman
19549   19528      0%      0%      0%  S            18948  cmcc
19541   19512      0%      0%      0%  S            35124  cpp_driver
19528   16366      0%      0%      0%  S             2432  pman
19512   16921      0%      0%      0%  S             2432  pman
19280   19243      0%      0%      0%  S            38708  cpp_ha_top_leve
19243   16921      0%      0%      0%  S             2436  pman
18966   18959      0%      0%      0%  S            49916  cpp_sp_svr
18959   16921      0%      0%      0%  S             2436  pman
18877   18862      0%      0%      0%  S             5780  pttcd
18862    4251      0%      0%      0%  S             2432  pman
18856       2      0%      0%      0%  S                0  kworker/1:1
18711   18691      0%      0%      0%  S            10352  hman
18691   16366      0%      0%      0%  S             2432  pman
18679   16921      0%      0%      0%  S             2436  pman
18517   18495      0%      0%      0%  S            60720  pubd
18495    4251      0%      0%      0%  S             2432  pman
18398   16921      0%      0%      0%  S             2432  pman
18211       2      0%      0%      0%  S                0  kworker/0:2
18140   18120      0%      0%      0%  S            10352  hman
18120   16921      0%      0%      0%  S             2436  pman
17448   16921      0%      0%      0%  S              428  inotifywait
17253       2      0%      0%      0%  S                0  kworker/1:0
17204       1      0%      0%      0%  S             2064  rotee
16921       1      0%      0%      0%  S             5512  pvp.sh
16744   16366      0%      0%      0%  S              428  inotifywait
16582       1      0%      0%      0%  S             2060  rotee
16366       1      0%      0%      0%  S             5512  pvp.sh
15627       2      0%      0%      0%  S                0  bioset
15626       2      0%      0%      0%  S                0  dmcrypt_write
15625       2      0%      0%      0%  S                0  kcryptd
15624       2      0%      0%      0%  S                0  kcryptd_io
15623       2      0%      0%      0%  S                0  bioset
15621       2      0%      0%      0%  S                0  kdmflush
15618       2      0%      0%      0%  S                0  loop1
15563       2      0%      0%      0%  S                0  ext4-rsv-conver
15562       2      0%      0%      0%  S                0  jbd2/mmcblk0p1-
15023       2      0%      0%      0%  S                0  kworker/u8:1
15002   14992      0%      0%      0%  S             1440  sort_files_by_i
14992    4251      0%      0%      0%  S             2416  pman
13884   13874      0%      0%      0%  S             2816  flash_check.sh
```

Overall Control Plane Resources

Control plane memory and CPU utilization on each control processor allows you to keep a tab on the overall control plane resources. You can use the **show platform software status control-processor brief** command (summary view) or the **show platform software status control-processor command** (detailed view) to view control plane memory and CPU utilization information.

All control processors should show status, Healthy. Other possible status values are Warning and Critical. Warning indicates that the router is operational, but that the operating level should be reviewed. Critical implies that the router is nearing failure.

If you see a Warning or Critical status, take the following actions:

- Reduce the static and dynamic loads on the system by reducing the number of elements in the configuration or by limiting the capacity for dynamic services.

- Reduce the number of routes and adjacencies, limit the number of ACLs and other rules, reduce the number of VLANs, and so on.

The following sections describe the fields in the **show platform software status control-processor** command output.

Load Average

Load average represents the process queue or process contention for CPU resources. For example, on a single-core processor, an instantaneous load of 7 would mean that seven processes are ready to run, one of which is currently running. On a dual-core processor, a load of 7 would mean that seven processes are ready to run, two of which are currently running.

Memory Utilization

Memory utilization is represented by the following fields:

- Total—Total system memory
- Used—Consumed memory
- Free—Available memory
- Committed—Virtual memory committed to processes

CPU Utilization

CPU utilization is an indication of the percentage of time the CPU is busy, and is represented by the following fields:

- CPU—Allocated processor
- User—Non-Linux kernel processes
- System—Linux kernel process
- Nice—Low-priority processes
- Idle—Percentage of time the CPU was inactive
- IRQ—Interrupts
- SIRQ—System Interrupts
- IOwait—Percentage of time CPU was waiting for I/O

Example: show platform software status control-processor Command

The following are some examples of using the **show platform software status control-processor** command:

```
Router# show platform software status control-processor
RP0: online, statistics updated 10 seconds ago
Load Average: healthy
  1-Min: 1.28, status: healthy, under 5.00
  5-Min: 0.74, status: healthy, under 5.00
  15-Min: 0.78, status: healthy, under 5.00
```

```
Memory (kb): healthy
  Total: 8154204
  Used: 2282364 (28%), status: healthy
  Free: 5871840 (72%)
  Committed: 2025108 (25%), under 90%
Per-core Statistics
CPU0: CPU Utilization (percentage of time spent)
  User:  2.46, System:  5.53, Nice:  0.00, Idle: 90.87
  IRQ:  0.82, SIRQ:  0.20, IOwait:  0.10
CPU1: CPU Utilization (percentage of time spent)
  User:  2.24, System:  5.91, Nice:  0.00, Idle: 90.91
  IRQ:  0.71, SIRQ:  0.20, IOwait:  0.00
CPU2: CPU Utilization (percentage of time spent)
  User:  0.50, System:  1.82, Nice:  0.00, Idle: 97.16
  IRQ:  0.50, SIRQ:  0.00, IOwait:  0.00
CPU3: CPU Utilization (percentage of time spent)
  User: 13.03, System: 12.88, Nice:  0.00, Idle: 62.51
  IRQ: 11.55, SIRQ:  0.00, IOwait:  0.00


Router# show platform software status control-processor brief
Load Average
Slot  Status  1-Min  5-Min 15-Min
  RP0 Healthy   0.99   0.72   0.77

Memory (kB)
Slot  Status    Total     Used (Pct)     Free (Pct) Committed (Pct)
  RP0 Healthy  8154204  2281012 (28%)  5873192 (72%)   2032232 (25%)

CPU Utilization
Slot  CPU   User System   Nice   Idle    IRQ   SIRQ IOwait
  RP0    0   1.02   1.84   0.00  96.30   0.61   0.10   0.10
         1   0.72   1.85   0.00  96.60   0.61   0.20   0.00
         2   0.50   1.62   0.00  97.25   0.60   0.00   0.00
         3  11.78  14.28   0.00  62.44  11.34   0.14   0.00


Boot Flash Disk Monitoring

*Aug 24 07:48:31.088 GMT: %FLASH_CHECK-3-DISK_QUOTA: R0/0: flash_check: Flash disk quota
exceeded
[free space is 83820 kB] - Please clean up files on flash1.
```

Monitoring Hardware Using Alarms

This section contains the following:

Router Design and Monitoring Hardware

The router sends alarm notifications when problems are detected, allowing you to monitor the network remotely. You do not need to use **show** commands to poll devices on a routine basis; however, you can perform onsite monitoring if you choose.

BootFlash Disk Monitoring

The bootflash disk must have enough free space to store two core dumps. This condition is monitored, and if the bootflash disk is too small to store two core dumps, a syslog alarm is generated, as shown in the following example:

```
Oct  6 14:10:56.292: %FLASH_CHECK-3-DISK_QUOTA: R0/0: flash_check: Flash disk quota exceeded

[free space is 1429020 kB] - Please clean up files on bootflash.
```

Approaches for Monitoring Hardware Alarms

This section contains the following:

Viewing the Console or Syslog for Alarm Messages

The network administrator can monitor alarm messages by reviewing alarm messages sent to the system console or to a system message log (syslog).

Enabling the logging alarm Command

The **logging alarm** command must be enabled for the system to send alarm messages to a logging device, such as the console or a syslog. This command is not enabled by default.

You can specify the severity level of the alarms to be logged. All the alarms at and above the specified threshold generate alarm messages. For example, the following command sends only critical alarm messages to logging devices:

```
Router(config)# logging alarm critical
```

If alarm severity is not specified, alarm messages for all severity levels are sent to logging devices.

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

The SNMP is an application-layer protocol that provides a standardized framework and a common language used for monitoring and managing devices in a network.

SNMP provides notification of faults, alarms, and conditions that might affect services. It allows a network administrator to access router information through a network management system (NMS) instead of reviewing logs, polling devices, or reviewing log reports.

To use SNMP to get alarm notification, use the following MIBs:

- ENTITY-MIB, RFC4133 (required for the CISCO-ENTITY-ALARM-MIB, ENTITY-STATE-MIB and CISCO-ENTITY-SENSOR-MIB to work)
- CISCO-ENTITY-ALARM-MIB
- ENTITY-STATE-MIB
- CISCO-ENTITY-SENSOR-MIB (for transceiver environmental alarm information, which is not provided through the CISCO-ENTITY-ALARM-MIB)

CHAPTER 34

Troubleshooting

This section contains the following:

- Troubleshooting, on page 319
- Understanding Diagnostic Mode, on page 319
- Before Contacting Cisco or Your Reseller, on page 320
- show interfaces Troubleshooting Command, on page 320
- Software Upgrade Methods, on page 321
- Change the Configuration Register, on page 321
- Recovering a Lost Password, on page 324

Troubleshooting

This section describes the troubleshooting scenarios.

Before troubleshooting a software problem, you must connect a PC to the router via the console port. With a connected PC, you can view status messages from the router and enter commands to troubleshoot a problem.

You can also remotely access the interface by using Telnet. The Telnet option assumes that the interface is up and running.

Understanding Diagnostic Mode

The router boots up or accesses diagnostic mode in the following scenarios:

- The IOS process or processes fail, in some scenarios. In other scenarios, the system resets when the IOS process or processes fail.
- A user-configured access policy was configured using the **transport-map** command that directs the user into the diagnostic mode.
- A send break signal (**Ctrl-C** or **Ctrl-Shift-6**) was entered while accessing the router, and the router was configured to enter diagnostic mode when a break signal was sent.

In the diagnostic mode, a subset of the commands that are available in user EXEC mode are made available to the users. Among other things, these commands can be used to:

- Inspect various states on the router, including the IOS state.

- Replace or roll back the configuration.
- Provide methods of restarting the IOS or other processes.
- Reboot hardware, such as the entire router, a module, or possibly other hardware components.
- Transfer files into or off of the router using remote access methods such as FTP, TFTP, and SCP.

The diagnostic mode provides a more comprehensive user interface for troubleshooting than previous routers, which relied on limited access methods during failures, such as ROMMON, to diagnose and troubleshoot Cisco IOS problems. The diagnostic mode commands can work when the Cisco IOS process is not working properly. These commands are also available in privileged EXEC mode on the router when the router is working normally.

Before Contacting Cisco or Your Reseller

If you cannot locate the source of a problem, contact your local reseller for advice. Before you call, you should have the following information ready:

- Chassis type and serial number
- Maintenance agreement or warranty information
- Type of software and version number
- Date you received the hardware
- Brief description of the problem
- Brief description of the steps you have taken to isolate the problem

show interfaces Troubleshooting Command

Use the **show interfaces** command to display the status of all physical ports and logical interfaces on the router. describes messages in the command output.

The IR1800 supports the following interfaces:

GigabitEthernet 0/0/0

GigabitEthernet 0/1/0 to 0/1/3

Cellular 0/4/0, 0/4/1, 0/5/0, and 0/5/1

Async 0/2/0 and 0/2/1

usbflash0:

msata

Alarm input alarm contact 0

Software Upgrade Methods

Several methods are available for upgrading software on the Cisco IR1800 Routers, including:

- Copy the new software image to flash memory over LAN or WAN when the existing Cisco IOS software image is in use.
- Copy the new software image to flash memory over the LAN while the boot image (ROM monitor) is operating.
- Copy the new software image over the console port while in ROM monitor mode.
- From ROM monitor mode, boot the router from a software image that is loaded on a TFTP server. To use this method, the TFTP server must be on the same LAN as the router.

Change the Configuration Register

To change a configuration register, follow these steps:

Procedure

Step 1 Connect a PC to the CONSOLE port on the router.

Step 2 At the privileged EXEC prompt (*router_name #*), enter the **show version** command to display the existing configuration register value (shown in bold at the bottom of this output example):

Example:

```
Router# show version
Cisco IOS XE Software, Version 17.06.01prd23
Cisco IOS Software [Bengaluru], ISR Software (ARMV8EL_LINUX_IOSD-UNIVERSALK9_IOT-M), Version
 17.6.1prd23, RELEASE SOFTWARE (fc1)
Technical Support: http://www.cisco.com/techsupport
Copyright (c) 1986-2021 by Cisco Systems, Inc.
Compiled Tue 20-Jul-21 02:28 by mcpre


Cisco IOS-XE software, Copyright (c) 2005-2021 by cisco Systems, Inc.
All rights reserved. Certain components of Cisco IOS-XE software are
licensed under the GNU General Public License ("GPL") Version 2.0. The
software code licensed under GPL Version 2.0 is free software that comes
with ABSOLUTELY NO WARRANTY. You can redistribute and/or modify such
GPL code under the terms of GPL Version 2.0. For more details, see the
documentation or "License Notice" file accompanying the IOS-XE software,
or the applicable URL provided on the flyer accompanying the IOS-XE
software.


ROM: 3.9(REL)

IR1833 uptime is 13 hours, 6 minutes
Uptime for this control processor is 13 hours, 9 minutes
System returned to ROM by Firmware Upgrade
System image file is "bootflash:ir1800-universalk9.17.06.01prd23.SPA.bin"
Last reload reason: Reload Command
```

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

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

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


Technology Package License Information:

------------------------------------------------------------------
Technology Type Technology-package Technology-package
Current Next Reboot
------------------------------------------------------------------
Smart License Perpetual network-advantage network-advantage
Smart License Subscription None None

The current throughput level is 50000 kbps


Smart Licensing Status: Registration Not Applicable/Not Applicable

cisco IR1833-K9 (1RU) processor with 470123K/6147K bytes of memory.
Processor board ID FCW2447P0EB
Router operating mode: Autonomous
MCU bootloader version: 0x22
MCU application version: 0x4d
1 Virtual Ethernet interface
6 Gigabit Ethernet interfaces
2 Serial interfaces
1 terminal line
4 Cellular interfaces
32768K bytes of non-volatile configuration memory.
3988088K bytes of physical memory.
7475198K bytes of Bootflash at bootflash:.

Configuration register is 0x2102

Router#
```

Step 3 Record the setting of the configuration register.

Step 4 To enable the break setting (indicated by the value of bit 8 in the configuration register), enter the **config-register** <*value*>command from privileged EXEC mode.

- Break enabled—Bit 8 is set to 0.
- Break disabled (default setting)—Bit 8 is set to 1.

Configuring the Configuration Register for Autoboot

Note Altering the configuration register is only for advanced troubleshooting and should only be done with guidance from Cisco support.

The configuration register can be used to change router behavior. This includes controlling how the router boots. Set the configuration register to 0x0 to boot into ROM, by using one of the following commands:

- In Cisco IOS configuration mode, use the **config-reg** 0x0 command.
- From the ROMMON prompt, use the **confreg** 0x0 command.

Note Setting the configuration register to 0x2102 will set the router to autoboot the Cisco IOS XE software.

Reset the Router

To reset the router, follow these steps:

Procedure

Step 1 If break is disabled, turn the router off (O), wait 5 seconds, and turn it on (|) again. Within 60 seconds, press the **Break** key. The terminal displays the ROM monitor prompt.

Note Some terminal keyboards have a key labeled *Break* . If your keyboard does not have a Break key, see the documentation that came with the terminal for instructions on how to send a break.

Step 2 Press break. The terminal displays the following prompt:

Example:

```
rommon 2>
```

Step 3 Enter **confreg 0x2142** to reset the configuration register:

Example:

```
rommon 2> confreg 0x142
```

Step 4 Sync the configuration changes with **sync** command.

Example:

```
rommon 2>sync
```

Step 5 Initialize the router by entering the **reload** command:

Example:

```
rommon 2>reload
```

The router cycles its power, and the configuration register is set to 0x2142. The router uses the boot ROM system image, indicated by the system configuration dialog:

Example:

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

Step 6 Enter **no** in response to the prompts until the following message is displayed:

Example:

```
Press RETURN to get started!
```

Step 7 Press **Return**. The following prompt appears:

Example:

```
Router>
```

Step 8 Enter the enable command to enter enable mode. Configuration changes can be made only in enable mode:

Example:

```
Router> enable
```

The prompt changes to the privileged EXEC prompt:

Example:

```
Router#
```

Step 9 Enter the **show startup-config** command to display an enable password in the configuration file:

Example:

```
Router# show startup-config
```

What to do next

If you are recovering an enable password, do not perform the steps in the Reset the Password and Save Your Changes section. Instead, complete the password recovery process by performing the steps in the Reset the Configuration Register Value section.

If you are recovering an enable secret password, it is not displayed in the **show startup-config** command output. Complete the password recovery process by performing the steps in the Reset the Password and Save Your Changes section.

Recovering a Lost Password

To recover a lost enable or lost enable-secret password, refer to the following sections:

1. Change the Configuration Register
2. Reset the Router

3. Reset the Password and Save your Changes (for lost enable secret passwords only)
4. Reset the Configuration Register Value.
5. If you have performed a **write erase**, or used the reset button, you will need to add the license.

```
IR1800#config term
IR1800#license smart reservation
```

Note Any vlan interfaces will have to be recreated, and it is possible you may need to re-generate certs.

Note Recovering a lost password is only possible when you are connected to the router through the console port. These procedures cannot be performed through a Telnet session.

Tip See the "Hot Tips" section on Cisco.com for additional information on replacing enable secret passwords.

Reset the Password and Save Your Changes

To reset your password and save the changes, follow these steps:

Procedure

Step 1 Enter the **configure terminal** command to enter global configuration mode:

Example:

```
Router# configure terminal
```

Step 2 Enter the **enable secret** command to reset the enable secret password in the router:

Example:

```
Router(config)# enable secret
password
```

Step 3 Enter **exit** to exit global configuration mode:

Example:

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

Step 4 Save your configuration changes:

Example:

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

Reset the Configuration Register Value

To reset the configuration register value after you have recovered or reconfigured a password, follow these steps:

Procedure

Step 1 Enter the **configure terminal** command to enter global configuration mode:

Example:

```
Router# configure terminal
```

Step 2 Enter the **configure register** command and the original configuration register value that you recorded.

Example:

```
Router(config)# config-reg
value
```

Step 3 Enter **exit** to exit configuration mode:

Example:

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

Note To return to the configuration being used before you recovered the lost enable password, do not save the configuration changes before rebooting the router.

Step 4 Reboot the router, and enter the recovered password.

Configuring a Console Port Transport Map

This task describes how to configure a transport map for a console port interface on the router.

Procedure

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

	Command or Action	Purpose
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**transport-map type console** *transport-map-name* **Example:** `Router(config)# transport-map type console consolehandler`	Creates and names a transport map for handling console connections, and enters transport map configuration mode.
Step 4	**connection wait** [**allow** [**interruptible**] | **none** [**disconnect**]] **Example:** `Router(config-tmap)# connection wait none`	Specifies how a console connection will be handled using this transport map. • **allow interruptible**—The console connection waits for a Cisco IOS VTY line to become available, and also allows users to enter diagnostic mode by interrupting a console connection that is waiting for a Cisco IOS VTY line to become available. This is the default setting. **Note** Users can interrupt a waiting connection by entering **Ctrl-C** or **Ctrl-Shift-6**. • **none**—The console connection immediately enters diagnostic mode.

Step 5	(Optional) **banner** [**diagnostic** | **wait**] *banner-message*
	Example:
	`Router(config-tmap)# banner diagnostic X`
	`Enter TEXT message. End with the character 'X'.`
	`--Welcome to Diagnostic Mode--`
	`X`
	`Router(config-tmap)#`
		(Optional) Creates a banner message that will be seen by users entering diagnostic mode or waiting for the Cisco IOS VTY line because of the console transport map configuration.
		• **diagnostic**—Creates a banner message seen by users directed to diagnostic mode because of the console transport map configuration.
		Note Users can interrupt a waiting connection by entering **Ctrl-C** or **Ctrl-Shift-6**.
		• **wait**—Creates a banner message seen by users waiting for Cisco IOS VTY to become available.
		• *banner-message*—Banner message, which begins and ends with the same delimiting character.

	Command or Action	Purpose
Step 6	**exit** **Example:** `Router(config-tmap)# exit`	Exits transport map configuration mode to re-enter global configuration mode.
Step 7	**transport type console** *console-line-number* **input** *transport-map-name* **Example:** `Router(config)# transport type console 0 input consolehandler`	Applies the settings defined in the transport map to the console interface. The *transport-map-name* for this command must match the *transport-map-name* defined in the **transport-map type console** command.

Examples

The following example shows how to create a transport map to set console port access policies and attach to console port 0:

```
Router(config)# transport-map type console consolehandler
Router(config-tmap)# connection wait allow interruptible
Router(config-tmap)# banner diagnostic X
Enter TEXT message. End with the character 'X'.
--Welcome to diagnostic mode--
X
Router(config-tmap)# banner wait X
Enter TEXT message. End with the character 'X'.
Waiting for IOS vty line
X
Router(config-tmap)# exit
Router(config)# transport type console 0 input consolehandler
```

Viewing Console Port, SSH, and Telnet Handling Configurations

Use the following commands to view console port, SSH, and Telnet handling configurations:

- **show transport-map**
- **show platform software configuration access policy**

Use the **show transport-map** command to view transport map configurations.

show transport-map [**all** | **name** *transport-map-name* | **type** [**console**]]

This command can be used either in user EXEC mode or privileged EXEC mode.

Example

The following example shows transport maps that are configured on the router: console port (`consolehandler`):

```
Router# show transport-map all
Transport Map:
Name: consolehandler Type: Console Transport
```

```
Connection:
Wait option: Wait Allow Interruptable Wait banner:

Waiting for the IOS CLI bshell banner:
Welcome to Diagnostic Mode



Router# show transport-map type console
Transport Map:
Name: consolehandler


REVIEW DRAFT - CISCO CONFIDENTIAL

Type: Console Transport

Connection:
Wait option: Wait Allow Interruptable Wait banner:

Waiting for the IOS CLI Bshell banner:
Welcome to Diagnostic Mode



Router# show transport-map type persistent ssh
Transport Map:
Name: consolehandler Type: Console Transport

Connection:
Wait option: Wait Allow Interruptable Wait banner:

Waiting for the IOS CLI Bshell banner:
Welcome to Diagnostic Mode
```

Use the **show platform software configuration access policy** command to view the current configurations for handling the incoming console port, SSH, and Telnet connections. The output of this command provides the current wait policy for each type of connection (Telnet, SSH, and console), as well as information on the currently configured banners.

Unlike the **show transport-map** command, the **show platform software configuration access policy** command is available in diagnostic mode so that it can be entered in scenarios where you need transport map configuration information, but cannot access the Cisco IOS CLI.

Example

The following example shows the **show platform software configuration access policy** command.

```
Router# show platform software configuration access policy
The current access-policies

Method : telnet
Rule : wait with interrupt Shell banner:
Welcome to Diagnostic Mode

Wait banner :
Waiting for IOS Process

Method : ssh Rule : wait Shell banner: Wait banner :
```

```
Method : console
Rule : wait with interrupt Shell banner:
Wait banner :
```

Using the factory reset Commands

The **factory reset** commands are used to remove all the customer specific data on a router/switch that has been added. The data can be configuration, log files, boot variables, core files, and so on.

The **factory-reset all** command erases the bootflash, nvram, rommon variables, licenses, and logs.

Caution Use of the factory reset command should not be done lightly. All customer configurations will be ddeleted and the platform will boot up as if new from the factory.

Note factory-reset all does not work if IOS-XE is running in controller mode. Please refer to SDWAN configuration information.

```
Router#factory-reset all
The factory reset operation is irreversible for all operations. Are you sure? [confirm]
*Enter*

*May 12 09:55:45.831: %SYS-5-RELOAD: Reload requested by Exec. Reload Reason: Factory Reset.

***Return to ROMMON Prompt
```

Boot Sequence after Factory Reset

Booting the image:

- The bootloader attempts to boot “golden.bin” from the bootflash: partition
- If no “golden.bin” is present, then boot the first image.

Loading the configuration:

- IOS looks for “golden.cfg” file on nvram: partition and applies it upon booting.
- If no “golden.cfg” is present on nvram: then IOS looks for “golden.cfg” file on bootflash: partition and applies it upon booting.
- If no “golden.cfg” is present on bootflash: then configurations are erased and Software Configuration dialog is used.