Configuring POS Interfaces on Cisco IOS XR Software

This module describes the configuration of Packet-over-SONET/SDH (POS) interfaces on the Cisco XR 12000 Series Routers.

POS interfaces provide secure and reliable data transmission over SONET and Synchronous Digital Hierarchy (SDH) frames using Cisco High-Level Data Link Control (HDLC) protocol or Point-to-Point Protocol (PPP) encapsulation. In addition to Cisco HDLC and PPP encapsulation, the Cisco XR 12000 Series Routersupports Frame Relay encapsulation.

The commands for configuring Layer 1 POS interfaces are provided in the *Cisco IOS XR Interface and Hardware Component Command Reference*.

Feature History for Configuring POS Interfaces on Cisco IOS XR Software

Release	Modification
Release 3.2	This feature was introduced on the Cisco XR 12000 Series Router.
Release 3.3.0	Support was added on the Cisco XR 12000 Series Router for the 2-Port OC-48 POS/RPR SPA.

Release 3.4.0	Support for the following features was introduced on the Cisco XR 12000 Series Router: • Subinterfaces with permanent virtual circuits (PVCs) • Frame Relay encapsulation on POS main interfaces and PVCs on the following hardware: – Cisco 4-Port OC-3 POS/SDH SPA – Cisco 8-Port OC-3 POS/SDH SPA – Cisco 2-Port OC-12 POS/SDH SPA – Cisco 4-Port OC-12 POS/SDH SPA – Cisco 8-Port OC-12 POS/SDH SPA – Cisco 1-Port OC-48/STM-16 POS/SDH SPA – Cisco 2-Port OC-48/STM-16 POS/SDH SPA – Cisco 1-Port OC-192c/STM-64c POS/SDH SPA – Cisco 4-Port OC-3c/STM-1 POS/SDH Line Card – Cisco 8-Port OC-3c/STM-1c POS/SDH Line Card – Cisco 16-Port OC-3c/STM-1c POS/SDH Line Card – Cisco 4-Port Channelized OC-12/STM-4 POS ISE Line Card – Cisco 4-Port OC-12c/STM-4 POS/SDH ISE Line Card – Cisco 1-Port Channelized OC-48/STM-16 POS ISE Line Card – Cisco 1-Port OC-48c/STM-16c POS/SDH ISE Line Card
Release 3.5.0	Support was added on the Cisco XR 12000 Series Router for the following SPAs: • Cisco 1-Port Channelized OC-3/STM-1 SPA • Cisco 1-Port Channelized OC-48/STM-16 SPA • Cisco 1-Port Channelized OC-12/DS0 SPA • Cisco 2-Port OC12 POS • Cisco 4-Port OC12 POS • Cisco 8-Port OC12 POS • Cisco 4-Port OC3 POS • Cisco 8-Port OC3 POS On the Cisco XR 12000 Series Router, L2TPv3-based L2VPN support was added on Frame Relay encapsulated POS interfaces. On the Cisco XR 12000 Series Router, the **l2transport** keyword was added to the **interface** command.
Release 3.8.0	Support was added on the Cisco XR 12000 Series Router for quality of service (QoS) on Layer 2 subinterfaces.

Contents

Prerequisites for Configuring POS Interfaces

You must be in a user group associated with a task group that includes the proper task IDs. The command reference guides include the task IDs required for each command. If you suspect user group assignment is preventing you from using a command, contact your AAA administrator for assistance.

Before configuring POS interfaces, be sure that the following conditions are met:

- You know the IP address of the interface you will assign to the new POS interface configuration.
- You have configured a clear channel or channelized SONET controller, as described in the "Configuring Clear Channel SONET Controllers on Cisco IOS XR Software" or "Configuring Channelized SONET/SDH on Cisco IOS XR Software" modules.

Information About Configuring POS Interfaces

To configure POS interfaces, you must understand the following concepts:

On the Cisco XR 12000 Series Router, a single POS interface carries data using PPP, Cisco HDLC, or Frame Relay encapsulation.

The router identifies the POS interface address by the physical layer interface module (PLIM) card rack number, slot number, bay number, and port number that are associated with that interface. If a subinterface and permanent virtual circuits (PVCs) are configured under the POS interface, then the router includes the subinterface number in the POS interface path ID.

Default Settings for POS Interfaces

When a POS interface is brought up and no additional configuration commands are applied, the default interface settings shown in Table 16 are present. These default settings can be changed by configuration.

Table 16 POS Modular Services Card and PLIM Default Interface Settings

Parameter	Configuration File Entry	Default Settings
Keepalive **Note** The **keepalive** command applies to POS interfaces using HDLC or PPP encapsulation. It does not apply to POS interfaces using Frame Relay encapsulation.	**keepalive** {*interval* [*retry*] | **disable**} **no keepalive**	Interval of 10 seconds Retry of: • 5 (with PPP encapsulation) • 3 (with HDLC encapsulation)
Encapsulation	**encapsulation** [**hdlc** | **ppp** | **frame-relay** [**IETF**]]	**hdlc**
Maximum transmission unit (MTU)	**mtu** *bytes*	4474 bytes
Cyclic redundancy check (CRC)	**crc** [**16** | **32**]	**32**

Note Default settings do not appear in the output of the **show running-config** command.

Cisco HDLC Encapsulation

Cisco High-Level Data Link Controller (HDLC) is the Cisco proprietary protocol for sending data over synchronous serial links using HDLC. Cisco HDLC also provides a simple control protocol called Serial Line Address Resolution Protocol (SLARP) to maintain serial link keepalives. HDLC is the default encapsulation type for POS interfaces under Cisco IOS XR software. Cisco HDLC is the default for data encapsulation at Layer 2 (data link) of the Open System Interconnection (OSI) stack for efficient packet delineation and error control.

Note Cisco HDLC is enabled by default for POS interfaces.

Cisco HDLC uses keepalives to monitor the link state, as described in the "Keepalive Timer" section on page 413.

PPP Encapsulation

PPP is a standard protocol used to send data over synchronous serial links. PPP also provides a Link Control Protocol (LCP) for negotiating properties of the link. LCP uses echo requests and responses to monitor the continuing availability of the link.

Note When an interface is configured with PPP encapsulation, a link is declared down, and full LCP negotiation is re-initiated after three ECHOREQ packets are sent without receiving an ECHOREP response.

PPP provides the following Network Control Protocols (NCPs) for negotiating the properties of data protocols that run on the link:

- IP Control Protocol (IPCP)—negotiates IP properties
- Multiprotocol Label Switching control processor (MPLSCP)—negotiates MPLS properties
- Cisco Discovery Protocol control processor (CDPCP)—negotiates CDP properties
- IPv6CP—negotiates IP Version 6 (IPv6) properties
- Open Systems Interconnection control processor (OSICP)—negotiates OSI properties

PPP uses keepalives to monitor the link state, as described in the "Keepalive Timer" section on page 413.

PPP supports the following authentication protocols, which require a remote device to prove its identity before allowing data traffic to flow over a connection:

- Challenge Handshake Authentication Protocol (CHAP)—CHAP authentication sends a challenge message to the remote device. The remote device encrypts the challenge value with a shared secret and returns the encrypted value and its name to the local router in a response message. The local router attempts to match the remote device's name with an associated secret stored in the local username or remote security server database; it uses the stored secret to encrypt the original challenge and verify that the encrypted values match.
- Microsoft Challenge Handshake Authentication Protocol (MS-CHAP)—MS-CHAP is the Microsoft version of CHAP. Like the standard version of CHAP, MS-CHAP is used for PPP authentication; in this case, authentication occurs between a personal computer using Microsoft Windows NT or Microsoft Windows 95 and a Cisco router or access server acting as a network access server.
- Password Authentication Protocol (PAP)—PAP authentication requires the remote device to send a name and a password, which are checked against a matching entry in the local username database or in the remote security server database.

Note For more information on enabling and configuring PPP authentication protocols, see the "Configuring PPP on Cisco IOS XR Software" module later in this manual.

Use the **ppp authentication** command in interface configuration mode to enable CHAP, MS-CHAP, and PAP on a POS interface.

Note Enabling or disabling PPP authentication does not effect the local router's willingness to authenticate itself to the remote device.

Keepalive Timer

Cisco keepalives are useful for monitoring the link state. Periodic keepalives are sent to and received from the peer at a frequency determined by the value of the keepalive timer. If an acceptable keepalive response is not received from the peer, the link makes the transition to the down state. As soon as an acceptable keepalive response is obtained from the peer or if keepalives are disabled, the link makes the transition to the up state.

If three keepalives are sent to the peer and no response is received from peer, then the link makes the transition to the down state. ECHOREQ packets are sent out only when LCP negotiation is complete (for example, when LCP is open).

Note The **keepalive** command applies to POS interfaces using HDLC or PPP encapsulation. It does not apply to POS interfaces using Frame Relay encapsulation.

Use the **keepalive** command in interface configuration mode to set the frequency at which LCP sends ECHOREQ packets to its peer. To restore the system to the default keepalive interval of 10 seconds, use the **keepalive** command with **no** argument. To disable keepalives, use the **keepalive disable** command. For both PPP and Cisco HDLC, a keepalive of 0 disables keepalives and is reported in the **show running-config** command output as **keepalive disable**.

To remove the **keepalive** command from the configuration entirely, use the **no keepalive** command. You must remove the **keepalive** command from an interface configuration before you can configure Frame Relay encapsulation on that interface. Frame Relay interfaces do not support keepalives.

Note During MDR, the keepalive interval must be 10 seconds or more.

When LCP is running on the peer and receives an ECHOREQ packet, it responds with an echo reply (ECHOREP) packet, regardless of whether keepalives are enabled on the peer.

Keepalives are independent between the two peers. One peer end can have keepalives enabled while the other end has them disabled. Even if keepalives are disabled locally, LCP still responds with ECHOREP packets to the ECHOREQ packets it receives. Similarly, LCP also works if the period of keepalives at each end is different.

Note Use the **debug chdlc slarp packet** command and other Cisco HDLC **debug** commands to display information about the Serial Line Address Resolution Protocol (SLARP) packets that are sent to the peer after the keepalive timer has been configured.

Frame Relay Encapsulation

On the Cisco XR 12000 Series Router, Frame Relay encapsulated POS interface configuration is hierarchical and comprises the following elements:

1. The POS main interface is comprised of the physical interface and port. If you are not using the POS interface to support Cisco HDLC and PPP encapsulated connections, then you must configure subinterfaces with PVCs under the POS main interface. Frame Relay connections are supported on PVCs only.
2. POS subinterfaces are configured under the POS main interface. A POS subinterface does not actively carry traffic until you configure a PVC under the POS subinterface.
3. Point-to-point and Layer 2 attachment circut (AC) PVCs are configured under a POS subinterface. You cannot configure a PVC directly under a main interface. A single point-to-point or L2 AC PVC is allowed per subinterface. PVCs use a predefined circuit path and fail if the path is interrupted. PVCs remain active until the circuit is removed. Connections on the POS PVC support Frame Relay encapsulation only.
4. Layer 3 configuration typically takes place on the subinterface.

Note The administrative state of a parent interface drives the state of the subinterface and its PVC. When the administrative state of a parent interface or subinterface changes, so does the administrative state of any child PVC configured under that parent interface or subinterface.

On the Cisco XR 12000 Series Router, the following hardware supports Frame Relay encapsulation:

- Cisco 1-Port 192c/STM-64c POS/SDH SPA
- Cisco 2-Port OC48/STM16 POS/SDH SPA
- Cisco 4-Port OC-3c/STM-1 POS/SDH Line Card
- Cisco 8-Port OC-3c/STM-1c POS/SDH Line Card
- Cisco 16-Port OC-3c/STM-1c POS/SDH Line Card
- Cisco 4-Port OC-12c/STM4 POS/SDH ISE Line Card
- Cisco 1-Port OC-48c/STM16c POS/SDH ISE Line Card

To configure Frame Relay encapsulation on POS interfaces, use the **encapsulation frame-relay** command.

Frame Relay interfaces support two types of encapsulated frames:

- Cisco (this is the default)
- IETF

Use the **encap** command in PVC configuration mode to configure Cisco or IETF encapsulation on a PVC. If the encapsulation type is not configured explicitly for a PVC, then that PVC inherits the encapsulation type from the main POS interface.

Note Cisco encapsulation is required on POS main interfaces that are configured for MPLS. IETF encapsulation is not supported for MPLS.

Before you configure Frame Relay encapsulation on an interface, you must verify that all prior Layer 3 configuration is removed from that interface. For example, you must ensure that there is no IP address configured directly under the main interface; otherwise, any Frame Relay configuration done under the main interface will not be viable.

LMI on Frame Relay Interfaces

The Local Management Interface (LMI) protocol monitors the addition, deletion, and status of PVCs. LMI also verifies the integrity of the link that forms a Frame Relay UNI interface. By default, **cisco** LMI is enabled on all PVCs. However, you can modify the default LMI type to be ANSI or Q.933, as described in the "Modifying the Default Frame Relay Configuration on an Interface" module later in this manual.

If the LMI type is **cisco** (the default LMI type), the maximum number of PVCs that can be supported under a single interface is related to the MTU size of the main interface. Use the following formula to calculate the maximum number of PVCs supported on a card or SPA:

```
(MTU - 13)/8 = maximum number of PVCs
```

Note The default setting of the **mtu** command for a POS interface is 4474 bytes. Therefore, the default numbers of PVCs supported on a POS interface configured with **cisco** LMI is 557.

Note You must configure the LMI interface type on Frame Relay interfaces; otherwise, the POS interface does not come up. For connections between Provider Edge (PE) and Customer Edge (CE) routers, the PE end must be DCE and the CE end must be DTE for LMI to come up. For more information about configuring the LMI interface type on a Frame Relay interface, see the "Configuring Frame Relay Cisco IOS XR Software" module.

Layer 2 Tunnel Protocol Version 3-Based Layer 2 VPN for Frame Relay

Layer 2 Tunnel Protocol Version 3 (L2TPv3) is a protocol used for tunneling Layer 2 payloads over an IP core network. L2TPv3 defines the signaling and formatting of packets for L2VPN on an IP Network.

Cisco IOS XR software supports a point-to-point, end-to-end service, where two attachment circuits (ACs) are connected together.

L2TPv3 connection setup requires the following tasks:

- Configuring an AC on each Provider Edge (PE) router
- Configuring an L2TPv3 encapsulated pseudowire between two PE routers.

This module describes how to configure a Layer 2 AC on a Frame Relay encapsulated POS interface.

For detailed information about configuring L2TPv3 pseudowires in your network, see the *Layer 2 Tunnel Protocol Version 3 on Cisco IOS XR Software* module of the *Cisco IOS XR MPLS Configuration Guide.*

For detailed information about configuring L2VPNs, see the "Implementing MPLS Layer 2 VPNs on Cisco IOS XR Software" module of the *Cisco IOS XR MPLS Configuration Guide.*

How to Configure a POS Interface

This section contains the following procedures:

Bringing Up a POS Interface

This task describes the commands you can use to bring up a POS interface.

Prerequisites

You must have a POS line card or SPA installed in a router that is running Cisco IOS XR software.

Restrictions

The configuration on both ends of the POS connection must match for the interface to be active.

SUMMARY STEPS

1. **show interfaces**
2. **configure**
3. **interface pos** *interface-path-id*
4. **ipv4 address** *ipv4_address/prefix*
5. **no shutdown**
6. **end**
   or
   commit
7. **exit**
8. **exit**
9. Repeat Step 1 through Step 8 to bring up the interface at the other end of the connection.
10. **show ipv4 interface brief**
11. **show interfaces pos** *interface-path-id*

DETAILED STEPS

	Command or Action	Purpose
Step 1	**show interfaces** **Example:** `RP/0/0/CPU0:router# show interfaces`	(Optional) Displays configured interfaces. • Use this command to also confirm that the router recognizes the PLIM card.
Step 2	**configure** **Example:** `RP/0/0/CPU0:router# configure`	Enters global configuration mode.
Step 3	**interface pos** *interface-path-id* **Example:** `RP/0/0/CPU0:router(config)# interface POS 0/3/0/0`	Specifies the POS interface name and notation *rack/slot/module/port,* and enters interface configuration mode.
Step 4	**ipv4 address** *ipv4_address/prefix* **Example:** `RP/0/0/CPU0:router (config)#ipv4 address 10.46.8.6/24`	Assigns an IP address and subnet mask to the interface. **Note** Skip this step if you are configuring Frame Relay encapsulation on this interface. For Frame Relay, the IP address and subnet mask are configured under the subinterface.

	Command or Action	Purpose
Step 5	**no shutdown** **Example:** `RP/0/0/CPU0:router (config-if)# no shutdown`	Removes the shutdown configuration. **Note** Removal of the shutdown configuration eliminates the forced administrative down on the interface, enabling it to move to an up or down state (assuming the parent SONET layer is not configured administratively down).
Step 6	**end** or **commit** **Example:** `RP/0/0/CPU0:router (config-if)# end` or `RP/0/0/CPU0:router(config-if)# commit`	Saves configuration changes. • When you issue the **end** command, the system prompts you to commit changes: `Uncommitted changes found, commit them before exiting(yes/no/cancel)? [cancel]:` – Entering **yes** saves configuration changes to the running configuration file, exits the configuration session, and returns the router to EXEC mode. – Entering **no** exits the configuration session and returns the router to EXEC mode without committing the configuration changes. – Entering **cancel** leaves the router in the current configuration session without exiting or committing the configuration changes. • Use the **commit** command to save the configuration changes to the running configuration file and remain within the configuration session.
Step 7	**exit** **Example:** `RP/0/0/CPU0:router (config-if)# exit`	Exits interface configuration mode and enters global configuration mode.
Step 8	**exit** **Example:** `RP/0/0/CPU0:router (config)# exit`	Exits global configuration mode and enters EXEC mode.

	Command or Action	Purpose
Step 9	**show interfaces** **configure** **interface pos** *interface-path-id* **no shut** **exit** **exit** **commit** **Example:** `RP/0/0/CPU0:router# show interfaces` `RP/0/0/CPU0:router# configure` `RP/0/0/CPU0:router (config)# interface pos 0/3/0/0` `RP/0/0/CPU0:router (config-if)# no shutdown` `RP/0/0/CPU0:router (config-if)# commit` `RP/0/0/CPU0:router (config-if)# exit` `RP/0/0/CPU0:router (config)# exit`	Repeat Step 1 through Step 8 to bring up the interface at the other end of the connection. **Note** The configuration on both ends of the POS connection must match.
Step 10	**show ipv4 interface brief** **Example:** `RP/0/0/CPU0:router # show ipv4 interface brief`	Verifies that the interface is active and properly configured. If you have brought up a POS interface properly, the "Status" field for that interface in the **show ipv4 interface brief** command output shows "Up."
Step 11	**show interfaces pos** *interface-path-id* **Example:** `RP/0/0/CPU0:router# show interfaces pos 0/3/0/0`	(Optional) Displays the interface configuration.

What to Do Next

To modify the default configuration of the POS interface you just brought up, see the "Configuring Optional POS Interface Parameters" section on page 419.

Configuring Optional POS Interface Parameters

This task describes the commands you can use to modify the default configuration on a POS interface.

Prerequisites

Before you modify the default POS interface configuration, you must bring up the POS interface and remove the shutdown configuration, as described in the "Bringing Up a POS Interface" section on page 416.

Restrictions

The configuration on both ends of the POS connection must match for the interface to be active.

SUMMARY STEPS

1. **configure**
2. **interface pos** *interface-path-id*
3. **encapsulation** [**hdlc** | **ppp** | **frame-relay** [**IETF**]]
4. **pos crc** {**16** | **32**}
5. **mtu** *value*
6. **end**
   or
   commit
7. **exit**
8. **exit**
9. **show interfaces pos** [*interface-path-id*]

DETAILED STEPS

	Command or Action	Purpose
Step 1	**configure** **Example:** `RP/0/0/CPU0:router# configure`	Enters global configuration mode.
Step 2	**interface pos** *interface-path-id* **Example:** `RP/0/0/CPU0:router(config)# interface POS 0/3/0/0`	Specifies the POS interface name and notation *rack/slot/module/port,* and enters interface configuration mode.
Step 3	**encapsulation** [**hdlc** | **ppp** | **frame-relay** [**IETF**]] **Example:** `RP/0/0/CPU0:router(config-if)# encapsulation hdlc`	(Optional) Configures the interface encapsulation parameters and details such as HDLC or PPP. **Note** The default encapsulation is **hdlc**.
Step 4	**pos crc** {**16** | **32**} **Example:** `RP/0/0/CPU0:router(config-if)# pos crc 32`	(Optional) Configures the CRC value for the interface. Enter the **16** keyword to specify 16-bit CRC mode, or enter the **32** keyword to specify 32-bit CRC mode. **Note** The default CRC is **32**.
Step 5	**mtu** *value* **Example:** `RP/0/0/CPU0:router(config-if)# mtu 4474`	(Optional) Configures the MTU value. • The default value is 4474. • The POS MTU range is 64–9216.

	Command or Action	Purpose
Step 6	**end** or **commit** **Example:** `RP/0/0/CPU0:router (config-if)# end` or `RP/0/0/CPU0:router(config-if)# commit`	Saves configuration changes. • When you issue the **end** command, the system prompts you to commit changes: `Uncommitted changes found, commit them before exiting(yes/no/cancel)? [cancel]:` – Entering **yes** saves configuration changes to the running configuration file, exits the configuration session, and returns the router to EXEC mode. – Entering **no** exits the configuration session and returns the router to EXEC mode without committing the configuration changes. – Entering **cancel** leaves the router in the current configuration session without exiting or committing the configuration changes. • Use the **commit** command to save the configuration changes to the running configuration file and remain within the configuration session.
Step 7	**exit** **Example:** `RP/0/0/CPU0:router (config-if)# exit`	Exits interface configuration mode and enters global configuration mode.
Step 8	**exit** **Example:** `RP/0/0/CPU0:router (config)# exit`	Exits global configuration mode and enters EXEC mode.
Step 9	**show interfaces pos** [*interface-path-id*] **Example:** `RP/0/0/CPU0:router# show interface pos 0/3/0/0`	(Optional) Displays general information for the specified POS interface.

What to Do Next

- To create a point-to-point Frame Relay subinterface with a PVC on the POS interface you just brought up, see the "Creating a Point-to-Point POS Subinterface with a PVC" section on page 422.
- To configure PPP authentication on POS interfaces where PPP encapsulation is enabled, see the *Configuring PPP on Cisco IOS XR Software* module later in this manual.
- To modify the keepalive interval on POS interfaces that have Cisco HDLC or PPP encapsulation enabled, see the "Modifying the Keepalive Interval on POS Interfaces" section on page 426.
- To modify the default Frame Relay configuration on POS interfaces that have Frame Relay encapsulation enabled, see the "Modifying the Default Frame Relay Configuration on an Interface" of the *Configuring Frame Relay Cisco IOS XR Software* module in this manual.

Creating a Point-to-Point POS Subinterface with a PVC

The procedure in this section creates a point-to-point POS subinterface and configures a permanent virtual circuit (PVC) on that POS subinterface.

Note Subinterface and PVC creation is supported on interfaces with Frame Relay encapsulation only.

Prerequisites

Before you can create a subinterface on a POS interface, you must bring up the main POS interface with Frame Relay encapsulation, as described in the "Bringing Up a POS Interface" section on page 416.

Restrictions

Only one PVC can be configured for each point-to-point POS subinterface.

SUMMARY STEPS

1. **configure**
2. **interface pos** *interface-path-id.subinterface* **point-to-point**
3. **ipv4 address** *ipv4_address/prefix*
4. **pvc** *dlci*
5. **end**
   or
   commit
6. Repeat Step 1 through Step 5 to bring up the POS subinterface and any associated PVC at the other end of the connection.

DETAILED STEPS

	Command or Action	Purpose
Step 1	**configure** **Example:** `RP/0/0/CPU0:router# configure`	Enters global configuration mode.
Step 2	**interface pos** *interface-path-id.subinterface* **point-to-point** **Example:** `RP/0/0/CPU0:router (config)# interface pos 0/3/0/0.1 point-to-point`	Enters POS subinterface configuration mode. Replace *subinterface* with a subinterface ID, in the range from 1 through 4294967295.

	Command or Action	Purpose
Step 3	**ipv4 address** *ipv4_address/prefix* **Example:** `RP/0/0/CPU0:router (config-subif)#ipv4 address 10.46.8.6/24`	Assigns an IP address and subnet mask to the subinterface.
Step 4	**pvc** *dlci* **Example:** `RP/0/0/CPU0:router (config-subif)# pvc 20`	Creates a POS permanent virtual circuit (PVC) and enters Frame Relay PVC configuration submode. Replace *dlci* with a PVC identifier, in the range from 16 to 1007. **Note** Only one PVC is allowed per subinterface.
Step 5	**end** or **commit** **Example:** `RP/0/0/CPU0:router (config-fr-vc)# end` or `RP/0/0/CPU0:router(config-fr-vc)# commit`	Saves configuration changes. • When you issue the **end** command, the system prompts you to commit changes: `Uncommitted changes found, commit them before exiting(yes/no/cancel)? [cancel]:` – Entering **yes** saves configuration changes to the running configuration file, exits the configuration session, and returns the router to EXEC mode. – Entering **no** exits the configuration session and returns the router to EXEC mode without committing the configuration changes. – Entering **cancel** leaves the router in the current configuration session without exiting or committing the configuration changes. • Use the **commit** command to save the configuration changes to the running configuration file and remain within the configuration session.
Step 6	**configure** **interface pos** *interface-path-id.subinterface* **pvc** *dlci* **commit** **Example:** `RP/0/0/CPU0:router# configure` `RP/0/0/CPU0:router (config)# interface pos 0/3/0/1.1` `RP/0/0/CPU0:router (config-subif)#ipv4 address 10.46.8.5/24` `RP/0/0/CPU0:router (config-subif)# pvc 20` `RP/0/0/CPU0:router (config-fr-vc)# commit`	Repeat Step 1 through Step 5 to bring up the POS subinterface and any associated PVC at the other end of the connection. **Note** The DLCI (or PVC identifier) must match on both ends of the subinterface connection. **Note** When assigning an IP address and subnet mask to the subinterface at the other end of the connection, keep in mind that the addresses at both ends of the connection must be in the same subnet.

What to Do Next

- To configure optional PVC parameters, see the "Configuring Optional PVC Parameters" section on page 424.
- To modify the default Frame Relay configuration on POS interfaces that have Frame Relay encapsulation enabled, see the "Modifying the Default Frame Relay Configuration on an Interface" of the *"Configuring Frame Relay Cisco IOS XR Software"* module.
- To attach a Layer 3 QOS service policy to the PVC under the PVC submode, refer to the appropriate Cisco IOS XR software configuration guide.

Configuring Optional PVC Parameters

This task describes the commands you can use to modify the default configuration on a POS PVC.

Prerequisites

Before you can modify the default PVC configuration, you must create the PVC on a POS subinterface, as described in the "Creating a Point-to-Point POS Subinterface with a PVC" section on page 422.

Restrictions

- The DLCI (or PVC identifier) must match on both ends of the PVC for the connection to be active.
- To change the PVC DLCI, you must delete the PVC and then add it back with the new DLCI.

SUMMARY STEPS

1. **configure**
2. **interface pos** *interface-path-id.subinterface*
3. **pvc** *dlci*
4. **encap** [**cisco** | **ietf**]
5. **service-policy** {**input** | **output**} *policy-map*
6. **end**
   or
   commit
7. Repeat Step 1 through Step 6 to configure the PVC at the other end of the connection.
8. **show frame-relay pvc** *dlci-number*
9. **show policy-map interface pos** *interface-path-id.subinterface* {**input** | **output**}
   or
   show policy-map type qos interface pos *interface-path-id.subinterface* {**input** | **output**}

DETAILED STEPS

	Command or Action	Purpose
Step 1	**configure** **Example:** `RP/0/0/CPU0:router# configure`	Enters global configuration mode.
Step 2	**interface pos** *interface-path-id.subinterface* **Example:** `RP/0/0/CPU0:router (config)# interface pos 0/3/0/0.1`	Enters POS subinterface configuration mode.
Step 3	**pvc** *dlci* **Example:** `RP/0/0/CPU0:router (config-subif)# pvc 20`	Enters subinterface configuration mode for the PVC. Replace *dlci* with the DLCI number used to identify the PVC. Range is from 16 to 1007.
Step 4	**encap** [**cisco** | **ietf**] **Example:** `RP/0/0/CPU0:router (config-fr-vc)# encap ietf`	(Optional) Configures the encapsulation for a Frame Relay PVC. **Note** If the encapsulation type is not configured explicitly for a PVC, then that PVC inherits the encapsulation type from the main POS interface.
Step 5	**service-policy** {**input** | **output**} *policy-map* **Example:** `RP/0/0/CPU0:router (config-fr-vc)# service-policy output policy1`	Attaches a policy map to an input subinterface or output subinterface. Once attached, the policy map is used as the service policy for the subinterface. **Note** For information on creating and configuring policy maps, refer to the *Cisco IOS XR Modular Quality of Service Configuration Guide*,
Step 6	**end** or **commit** **Example:** `RP/0/0/CPU0:router (config-fr-vc)# end` or `RP/0/0/CPU0:router(config-fr-vc)# commit`	Saves configuration changes. • When you issue the **end** command, the system prompts you to commit changes: `Uncommitted changes found, commit them before exiting(yes/no/cancel)? [cancel]:` – Entering **yes** saves configuration changes to the running configuration file, exits the configuration session, and returns the router to EXEC mode. – Entering **no** exits the configuration session and returns the router to EXEC mode without committing the configuration changes. – Entering **cancel** leaves the router in the current configuration session without exiting or committing the configuration changes. • Use the **commit** command to save the configuration changes to the running configuration file and remain within the configuration session.

	Command or Action	Purpose
Step 7	**configure** **interface pos** *interface-path-id.subinterface* **pvc** *dlci* **encap** [**cisco** | **ietf**] **commit** **Example:** `RP/0/0/CPU0:router# configure` `RP/0/0/CPU0:router (config)# interface pos 0/3/0/1.1` `RP/0/0/CPU0:router (config-subif)# pvc 20` `RP/0/0/CPU0:router (config-fr-vc)# encap cisco` `RP/0/0/CPU0:router (config-fr-vc)# commit`	Repeat Step 1 through Step 6 to bring up the POS subinterface and any associated PVC at the other end of the connection. **Note** The configuration on both ends of the subinterface connection must match.
Step 8	**show frame-relay pvc** *dlci-number* **Example:** `RP/0/0/CPU0:router# show frame-relay pvc 20`	(Optional) Verifies the configuration of specified POS interface.
Step 9	**show policy-map interface pos** *interface-path-id.subinterface* {**input** | **output**} or **show policy-map type qos interface pos** *interface-path-id.subinterface* {**input** | **output**} **Example:** `RP/0/0/CPU0:router# show policy-map interface pos 0/3/0/0.1 output` or `RP/0/0/CPU0:router# show policy-map type qos interface pos 0/3/0/0.1 output`	(Optional) Displays the statistics and the configurations of the input and output policies that are attached to a subinterface.

What to Do Next

To modify the default Frame Relay configuration on POS interfaces that have Frame Relay encapsulation enabled, see the "Modifying the Default Frame Relay Configuration on an Interface" of the *"Configuring Frame Relay Cisco IOS XR Software"* module.

Modifying the Keepalive Interval on POS Interfaces

Perform this task to modify the keepalive interval on POS interfaces that have Cisco HDLC or PPP encapsulation enabled.

Note When you enable Cisco HDLC or PPP encapsulation on a POS interface, the default keepalive interval is 10 seconds. Use this procedure to modify that default keepalive interval.

Note Cisco HDLC is enabled by default on POS interfaces.

Prerequisites

Before you can modify the keepalive timer configuration, you must ensure that Cisco HDLC or PPP encapsulation is enabled on the interface. Use the **encapsulation** command to enable Cisco HDLC or PPP encapsulation on the interface, as described in the "Configuring Optional POS Interface Parameters" section on page 419.

Restrictions

During MDR, the keepalive interval must be 10 seconds or more.

SUMMARY STEPS

1. **configure**
2. **interface pos** *interface-path-id*
3. **keepalive** {*seconds* [*retry-count*] | **disable**}
   or
   no keepalive
4. **end**
   or
   commit
5. **show interfaces** *type interface-path-id*

DETAILED STEPS

	Command or Action	Purpose
Step 1	**configure** **Example:** `RP/0/0/CPU0:router# configure`	Enters global configuration mode.
Step 2	**interface pos** *interface-path-id* **Example:** `RP/0/0/CPU0:router(config)# interface POS 0/3/0/0`	Specifies the POS interface name and notation *rack/slot/module/port* and enters interface configuration mode.
Step 3	**keepalive** {*seconds* [*retry-count*] | **disable**} or **no keepalive** **Example:** `RP/0/0/CPU0:router(config-if)# keepalive 3` or `RP/0/0/CPU0:router(config-if)# no keepalive`	Specifies the number of seconds between keepalive messages, and optionally the number of keepalive messages that can be sent to a peer without a response before transitioning the link to the down state. • Use the **keepalive disable** command, the **no keepalive**, or the **keepalive** command with an argument of 0 to disable the keepalive feature entirely. • If keepalives are configured on an interface, use the **no keepalive** command to disable the keepalive feature before you configure Frame Relay encapsulation on that interface.

	Command or Action	Purpose
Step 4	**end** or **commit** **Example:** `RP/0/0/CPU0:router(config-if)# end` or `RP/0/0/CPU0:router(config-if)# commit`	Saves configuration changes. • When you issue the **end** command, the system prompts you to commit changes: `Uncommitted changes found, commit them before exiting(yes/no/cancel)? [cancel]:` – Entering **yes** saves configuration changes to the running configuration file, exits the configuration session, and returns the router to EXEC mode. – Entering **no** exits the configuration session and returns the router to EXEC mode without committing the configuration changes. – Entering **cancel** leaves the router in the current configuration session without exiting or committing the configuration changes. • Use the **commit** command to save the configuration changes to the running configuration file and remain within the configuration session.
Step 5	**show interfaces pos** *interface-path-id* **Example:** `RP/0/0/CPU0:router# show interfaces POS 0/3/0/0`	(Optional) Verifies the interface configuration.

How to Configure a Layer 2 Attachment Circuit

The Layer 2 AC configuration tasks are described in the following procedures:

- Creating a Layer 2 Frame Relay Subinterface with a PVC
- Configuring Optional Layer 2 PVC Parameters

Note After you configure an interface for Layer 2 switching, no routing commands such as **ipv4 address** are permissible.

Note Layer 2 ACs are not supported on interfaces configured with HDLC or PPP encapsulation.

Creating a Layer 2 Frame Relay Subinterface with a PVC

The procedure in this section creates a Layer 2 Frame Relay subinterface with a PVC.

Prerequisites

Before you can create a subinterface on a POS interface, you must bring up a POS interface, as described in the "Bringing Up a POS Interface" section on page 416.

Note You must skip Step 4 of the "Bringing Up a POS Interface" configuration steps when configuring an interface for Layer 2 switching. The **ipv4 address** command is not permissible on Frame Relay encapsulated interface.

Restrictions

- Only one PVC can be configured for each subinterface.
- The configuration on both ends of the PVC must match for the connection to operate properly.
- The **ipv4 address** command is not permissible on Frame Relay encapsulated interface. Any previous configuration of an IP address must be removed before you can configure an interface for Layer 2 transport mode.
- Layer 2 configuration is supported on Frame Relay PVCs only. Layer 2 Port mode, where Layer 2 configuration is applied directly under the main POS interface, is not supported.

SUMMARY STEPS

1. **configure**
2. **interface pos** *interface-path-id.subinterface* **l2transport**
3. **pvc** *dlci*
4. **end**
   or
   commit
5. Repeat Step 1 through Step 4 to bring up the subinterface and any associated PVC at the other end of the AC.

DETAILED STEPS

	Command or Action	Purpose
Step 1	**configure** **Example:** `RP/0/0/CPU0:router# configure`	Enters global configuration mode.
Step 2	**interface pos** *interface-path-id.subinterface* **l2transport** **Example:** `RP/0/0/CPU0:router(config)# interface pos 0/3/0/0.1 l2transport`	Creates a subinterface and enters POS subinterface configuration mode for that subinterface. **Note** The *subinterface* must be unique to any other subinterfaces configured under a single main interface.
Step 3	**pvc** dlci **Example:** `RP/0/0/CPU0:router(config-if)# pvc 100`	Creates a Frame Relay permanent virtual circuit (PVC) and enters Layer 2 transport PVC configuration mode. Replace *dlci* with the DLCI number used to identify the PVC. Range is from 16 to 1007. **Note** Only one PVC is allowed per subinterface.
Step 4	**end** or **commit** **Example:** `RP/0/0/CPU0:router(config-fr-vc)# end` or `RP/0/0/CPU0:router(config-fr-vc)# commit`	Saves configuration changes. • When you issue the **end** command, the system prompts you to commit changes: `Uncommitted changes found, commit them before exiting(yes/no/cancel)? [cancel]:` – Entering **yes** saves configuration changes to the running configuration file, exits the configuration session, and returns the router to EXEC mode. – Entering **no** exits the configuration session and returns the router to EXEC mode without committing the configuration changes. – Entering **cancel** leaves the router in the current configuration session without exiting or committing the configuration changes. • Use the **commit** command to save the configuration changes to the running configuration file and remain within the configuration session.
Step 5	Repeat Step 1 through Step 4 to bring up the subinterface and any associated PVC at the other end of the AC.	Brings up the AC. **Note** The configuration on both ends of the AC must match.

What to Do Next

- To configure optional subinterface parameters, see the "Configuring Optional Layer 2 Subinterface Parameters" section on page 433.

- To configure optional PVC parameters, see the "Configuring Optional Layer 2 PVC Parameters" section on page 431.
- To configure a point-to-point pseudowire XConnect on the AC you just created, see the "*Implementing Layer 2 Tunnel Protocol Version 3 on Cisco IOS XR Software"* module of the *Cisco IOS XR Virtual Private Network Configuration Guide for the Cisco XR 12000 Series Router.*
- To configure an L2VPN, see the "*Implementing MPLS Layer 2 VPNs on Cisco IOS XR Software"* module of the *Cisco IOS XR Virtual Private Network Configuration Guide for the Cisco XR 12000 Series Router.*

Configuring Optional Layer 2 PVC Parameters

This task describes the commands you can use to modify the default configuration on a Frame Relay Layer 2 PVC.

Prerequisites

You must create the PVC on a Layer 2 subinterface, as described in the "Creating a Layer 2 Frame Relay Subinterface with a PVC" section on page 429.

SUMMARY STEPS

1. **configure**
2. **interface pos** *interface-path-id.subinterface* **l2transport**
3. **pvc** *dlci*
4. **encap** [**cisco** | **ietf**]
5. **service-policy** {**input** | **output**} *policy-map*
6. **end**
   or
   commit
7. Repeat Step 1 through Step 5 to configure the PVC at the other end of the AC.
8. **show policy-map interface pos** *interface-path-id.subinterface* {**input** | **output**}
   or
   show policy-map type qos interface pos *interface-path-id.subinterface* {**input** | **output**}

DETAILED STEPS

	Command or Action	Purpose
Step 1	**configure** **Example:** `RP/0/0/CPU0:router# configure`	Enters global configuration mode.
Step 2	**interface pos** *interface-path-id.subinterface* **l2transport** **Example:** `RP/0/0/CPU0:router(config)# interface pos 0/6/0/1.10 l2transport`	Enters POS subinterface configuration mode for a Layer 2 Frame Relay subinterface.
Step 3	**pvc** *dlci* **Example:** `RP/0/0/CPU0:router(config-if)# pvc 100`	Enters Frame Relay PVC configuration mode for the specified PVC. Replace *dlci* with the DLCI number used to identify the PVC. Range is from 16 to 1007.
Step 4	**encap** {**cisco** | **ietf**} **Example:** `RP/0/0/CPU0:router(config-fr-vc)# encap ietf`	Configures the encapsulation for a Frame Relay PVC. The encapsulation type must match on both ends of the PVC.
Step 5	**service-policy** {**input** | **output**} *policy-map* **Example:** `RP/0/0/CPU0:router (config-fr-vc)# service-policy output policy1`	Attaches a policy map to an input subinterface or output subinterface. Once attached, the policy map is used as the service policy for the subinterface. **Note** For information on creating and configuring policy maps, refer to the *Cisco IOS XR Modular Quality of Service Configuration Guide*,
Step 6	**end** or **commit** **Example:** `RP/0/0/CPU0:router(config-pos-l2transport-pvc)# end` or `RP/0/0/CPU0:router(config-pos-l2transport-pvc)# commit`	Saves configuration changes. • When you issue the **end** command, the system prompts you to commit changes: `Uncommitted changes found, commit them before exiting(yes/no/cancel)? [cancel]:` – Entering **yes** saves configuration changes to the running configuration file, exits the configuration session, and returns the router to EXEC mode. – Entering **no** exits the configuration session and returns the router to EXEC mode without committing the configuration changes. – Entering **cancel** leaves the router in the current configuration session without exiting or committing the configuration changes. Use the **commit** command to save the configuration changes to the running configuration file and remain within the configuration session.

	Command or Action	Purpose
Step 7	Repeat Step 1 through Step 5 to configure the PVC at the other end of the AC.	Brings up the AC. **Note** The configuration on both ends of the connection must match.
Step 8	**show policy-map interface pos** *interface-path-id.subinterface* {**input** | **output**} or **show policy-map type qos interface pos** *interface-path-id.subinterface* {**input** | **output**} **Example:** `RP/0/0/CPU0:router# show policy-map interface pos 0/6/0/1.10 output` or `RP/0/0/CPU0:router# show policy-map type qos interface pos 0/6/0/1.10 output`	(Optional) Displays the statistics and the configurations of the input and output policies that are attached to a subinterface.

What to Do Next

- To configure a point-to-point pseudowire XConnect on the AC you just created, see the "*Implementing Layer 2 Tunnel Protocol Version 3 on Cisco IOS XR Software*" module of the *Cisco IOS XR Virtual Private Network Configuration Guide for the Cisco XR 12000 Series Router.*
- To configure an L2VPN, see the "*Implementing MPLS Layer 2 VPNs on Cisco IOS XR Software*" module of the *Cisco IOS XR Virtual Private Network Configuration Guide for the Cisco XR 12000 Series Router.*

Configuring Optional Layer 2 Subinterface Parameters

This task describes the commands you can use to modify the default configuration on a Frame Relay Layer 2 subinterface.

Prerequisites

Before you can modify the default PVC configuration, you must create the PVC on a Layer 2 subinterface, as described in the "Creating a Layer 2 Frame Relay Subinterface with a PVC" section on page 429.

Restrictions

In most cases, the MTU that is configured under the subinterface has priority over the MTU that is configured under the main interface. The exception to this rule is when the subinterface MTU is higher than main interface MTU. In such cases, the subinterface MTU displays the configured value in the CLI output, but the actual operational MTU is the value that is configured under the main interface value. To avoid confusion when troubleshooting and optimizing your Layer 2 connections, we recommend always configuring a higher MTU on main interface.

SUMMARY STEPS

1. **configure**
2. **interface pos** *interface-path-id.subinterface*
3. **mtu** *value*
4. **end**
   or
   commit
5. Repeat Step 1 through Step 4 to configure the subinterface at the other end of the AC.

DETAILED STEPS

	Command or Action	Purpose
Step 1	**configure** **Example:** `RP/0/0/CPU0:router# configure`	Enters global configuration mode.
Step 2	**interface pos** *interface-path-id.subinterface* **Example:** `RP/0/0/CPU0:router(config)# interface pos 0/3/0/1.1`	Enters POS subinterface configuration mode for a Layer 2 Frame Relay subinterface.
Step 3	**mtu** *value* **Example:** `RP/0/0/CPU0:router(config-if)# mtu 5000`	(Optional) Configures the MTU value. Range is from 64 through 65535.

	Command or Action	Purpose
Step 4	**end** or **commit** **Example:** `RP/0/0/CPU0:router(config-pos-l2transport-pvc)# end` or `RP/0/0/CPU0:router(config-pos-l2transport-pvc)# commit`	Saves configuration changes. • When you issue the **end** command, the system prompts you to commit changes: `Uncommitted changes found, commit them before exiting(yes/no/cancel)? [cancel]:` – Entering **yes** saves configuration changes to the running configuration file, exits the configuration session, and returns the router to EXEC mode. – Entering **no** exits the configuration session and returns the router to EXEC mode without committing the configuration changes. – Entering **cancel** leaves the router in the current configuration session without exiting or committing the configuration changes. Use the **commit** command to save the configuration changes to the running configuration file and remain within the configuration session.
Step 5	Repeat Step 1 through Step 4 to configure the PVC at the other end of the AC.	Brings up the AC. **Note** The configuration on both ends of the connection must match.

Configuration Examples for POS Interfaces

This section provides the following configuration examples:

- Bringing Up and Configuring a POS Interface with Cisco HDLC Encapsulation: Example, page 435
- Configuring a POS Interface with Frame Relay Encapsulation: Example, page 436
- Configuring a POS Interface with PPP Encapsulation: Example, page 437

Bringing Up and Configuring a POS Interface with Cisco HDLC Encapsulation: Example

The following example shows how to bring up a basic POS interface with Cisco HDLC encapsulation:

```
RP/0/0/CPU0:router# configure
RP/0/0/CPU0:router(config)# interface POS 0/3/0/0
RP/0/0/CPU0:router(config-if)# ipv4 address 172.18.189.38 255.255.255.224
RP/0/0/CPU0:router(config-if)# no shutdown
RP/0/0/CPU0:router(config-if)# end
Uncommitted changes found, commit them? [yes]: yes
```

The following example shows how to configure the interval between keepalive messages to be 10 seconds:

```
RP/0/0/CPU0:router# configure
RP/0/0/CPU0:router(config)# interface POS 0/3/0/0
RP/0/0/CPU0:router(config-if)# keepalive 10
RP/0/0/CPU0:router(config-if)# commit
```

Configuring a POS Interface with Frame Relay Encapsulation: Example

The following example shows how to create a POS interface with Frame Relay encapsulation and a point-to-point POS subinterface with a PVC on router 1:

```
RP/0/0/CPU0:router# configure
RP/0/0/CPU0:router(config)# interface POS 0/3/0/0
RP/0/0/CPU0:router(config-if)# encapsulation frame-relay
RP/0/0/CPU0:router(config-if)# no shutdown
RP/0/0/CPU0:router(config-if)# end
Uncommitted changes found, commit them? [yes]: yes

RP/0/0/CPU0:router# configure
RP/0/0/CPU0:router (config)# interface pos 0/3/0/0.1 point-to-point
RP/0/0/CPU0:router (config-subif)#ipv4 address 10.20.3.1/24
RP/0/0/CPU0:router (config-subif)# pvc 100
RP/0/0/CPU0:router(config-if)# end
Uncommitted changes found, commit them? [yes]: yes

RP/0/0/CPU0:router# show interface POS 0/3/0/0

Wed Oct  8 04:20:30.248 PST DST
POS0/3/0/0 is up, line protocol is up
  Interface state transitions: 1
  Hardware is Packet over SONET/SDH
  Internet address is 10.20.3.1/24
  MTU 4474 bytes, BW 155520 Kbit
     reliability 255/255, txload 0/255, rxload 0/255
  Encapsulation FRAME-RELAY, crc 32, controller loopback not set,
  LMI enq sent  116, LMI stat recvd 76, LMI upd recvd 0, DTE LMI up
  LMI enq recvd 0, LMI stat sent  0, LMI upd sent  0
  LMI DLCI 1023  LMI type is CISCO  frame relay DTE
  Last clearing of "show interface" counters 00:00:06
  5 minute input rate 0 bits/sec, 0 packets/sec
  5 minute output rate 0 bits/sec, 0 packets/sec
     1 packets input, 13 bytes, 0 total input drops
     0 drops for unrecognized upper-level protocol
     Received 0 runts, 0 giants, 0 throttles, 0 parity
     0 input errors, 0 CRC, 0 frame, 0 overrun, 0 ignored, 0 abort
     1 packets output, 13 bytes, 0 total output drops
     0 output errors, 0 underruns, 0 applique, 0 resets
     0 output buffer failures, 0 output buffers swapped out
```

The following example shows how to create a POS interface with Frame Relay encapsulation and a point-to-point POS subinterface with a PVC on router 2, which is connected to router 1:

```
RP/0/0/CPU0:router# configure
RP/0/0/CPU0:router(config)# interface POS 0/3/0/1
RP/0/0/CPU0:router(config-if)# encapsulation frame-relay
RP/0/0/CPU0:router(config-if)# frame-relay intf-type dce
RP/0/0/CPU0:router(config-if)# no shutdown
RP/0/0/CPU0:router(config-if)# end
Uncommitted changes found, commit them? [yes]: yes
```

```
RP/0/0/CPU0:router# configure
RP/0/0/CPU0:router (config)# interface pos 0/3/0/1.1 point-to-point
RP/0/0/CPU0:router (config-subif)#ipv4 address 10.20.3.2/24
RP/0/0/CPU0:router (config-subif)# pvc 100
RP/0/0/CPU0:router(config-if)# end
Uncommitted changes found, commit them? [yes]: yes

RP/0/0/CPU0:router# show interface POS 0/3/0/1

Wed Oct  8 04:20:38.037 PST DST
POS0/3/0/1 is up, line protocol is up
  Interface state transitions: 1
  Hardware is Packet over SONET/SDH
  Internet address is 10.20.3.2/24
  MTU 4474 bytes, BW 155520 Kbit
     reliability 255/255, txload 0/255, rxload 0/255
  Encapsulation FRAME-RELAY, crc 32, controller loopback not set,
  LMI enq sent  0, LMI stat recvd 0, LMI upd recvd 0
  LMI enq recvd 77, LMI stat sent  77, LMI upd sent  0 , DCE LMI up
  LMI DLCI 1023  LMI type is CISCO  frame relay DCE
  Last clearing of "show interface" counters 00:00:14
  5 minute input rate 0 bits/sec, 0 packets/sec
  5 minute output rate 0 bits/sec, 0 packets/sec
     2 packets input, 26 bytes, 0 total input drops
     0 drops for unrecognized upper-level protocol
     Received 0 runts, 0 giants, 0 throttles, 0 parity
     0 input errors, 0 CRC, 0 frame, 0 overrun, 0 ignored, 0 abort
     2 packets output, 26 bytes, 0 total output drops
     0 output errors, 0 underruns, 0 applique, 0 resets
     0 output buffer failures, 0 output buffers swapped out
```

The following example shows how create a Layer 2 POS subinterface with a PVC on the main POS interface:

```
RP/0/0/CPU0:router# configure
RP/0/0/CPU0:router (config)# interface pos 0/3/0/0.1 l2transport
RP/0/0/CPU0:router (config-subif)# pvc 100
RP/0/0/CPU0:router(config-subif)# commit
```

Configuring a POS Interface with PPP Encapsulation: Example

The following example shows how to create and configure a POS interface with PPP encapsulation:

```
RP/0/0/CPU0:router# configure
RP/0/0/CPU0:router(config)# interface POS 0/3/0/0
RP/0/0/CPU0:router(config-if)# ipv4 address 172.18.189.38 255.255.255.224
RP/0/0/CPU0:router(config-if)# encapsulation ppp
RP/0/0/CPU0:router(config-if)# no shutdown
RP/0/0/CPU0:router(config-if)# end
Uncommitted changes found, commit them? [yes]: yes

RP/0/0/CPU0:router# show interfaces POS 0/3/0/0

POS0/3/0/0 is down, line protocol is down
  Hardware is Packet over SONET
  Internet address is 172.18.189.38/27
  MTU 4474 bytes, BW 2488320 Kbit
     reliability 0/255, txload Unknown, rxload Unknown
  Encapsulation PPP, crc 32, controller loopback not set, keepalive set (
10 sec)
  LCP Closed
  Closed: IPCP
```

```
Last clearing of "show interface" counters never
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 total input drops
   0 drops for unrecognized upper-level protocol
   Received 0 broadcast packets, 0 multicast packets
            0 runts, 0 giants, 0 throttles, 0 parity
   0 input errors, 0 CRC, 0 frame, 0 overrun, 0 ignored, 0 abort
   0 packets output, 0 bytes, 0 total output drops
   Output 0 broadcast packets, 0 multicast packets
   0 output errors, 0 underruns, 0 applique, 0 resets
   0 output buffer failures, 0 output buffers swapped out
   0 carrier transitions
```

Additional References

The following sections provide references related to POS interface configuration.

Related Documents

Related Topic	Document Title
Cisco IOS XR master command reference	*Cisco IOS XR Master Commands List*
Cisco IOS XR interface configuration commands	*Cisco IOS XR Interface and Hardware Component Command Reference*
Initial system bootup and configuration information for a router using the Cisco IOS XR software.	*Cisco IOS XR Getting Started Guide*
Cisco IOS XR AAA services configuration information	*Cisco IOS XR System Security Configuration Guide* and *Cisco IOS XR System Security Command Reference*
Information about user groups and task IDs	*Cisco IOS XR Interface and Hardware Component Command Reference*

Standards

Standards	Title
FRF.1.2	*PVC User-to-Network Interface (UNI) Implementation Agreement - July 2000*
ANSI T1.617 Annex D	—
ITU Q.933 Annex A	—

MIBs

MIBs	MIBs Link
—	To locate and download MIBs using Cisco IOS XR software, use the Cisco MIB Locator found at the following URL and choose a platform under the Cisco Access Products menu: http://cisco.com/public/sw-center/netmgmt/cmtk/mibs.shtml

RFCs

RFCs	Title
RFC 1294	*Multiprotocol Interconnect Over Frame Relay*
RFC 1315	*Management Information Base for Frame Relay DTEs*
RFC 1490	*Multiprotocol Interconnect Over Frame Relay*
RFC 1586	*Guidelines for Running OSPF Over Frame Relay Networks*
RFC 1604	*Definitions of Managed Objects for Frame Relay Service*
RFC 2115	*Management Information Base for Frame Relay DTEs Using SMIv2*
RFC 2390	*Inverse Address Resolution Protocol*
RFC 2427	*Multiprotocol Interconnect Over Frame Relay*
RFC 2954	*Definitions of Managed Objects for Frame Relay Service*

Technical Assistance

Description	Link
The Cisco Technical Support website contains thousands of pages of searchable technical content, including links to products, technologies, solutions, technical tips, and tools. Registered Cisco.com users can log in from this page to access even more content.	http://www.cisco.com/techsupport