



# Configuring POS Interfaces

This module describes the configuration of Packet-over-SONET/SDH (POS) interfaces.

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.

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

Release	Modification
Release 2.0	This feature was introduced on the Cisco CRS-1 Router.
Release 3.2	Support for the following hardware was introduced on the Cisco CRS-1 Router: <ul style="list-style-type: none"> <li>• 1-Port OC-192c/STM-64 POS/RPR XFP SPA</li> <li>• 4-Port OC-3c/STM-1 POS SPA</li> <li>• SIP-800</li> </ul>
Release 3.3.0	Support for the 8-Port OC-12c/STM-4 POS SPA was introduced on the Cisco CRS-1 Router.
Release 3.4.0	Support was added on the Cisco CRS-1 Router for the following hardware: <ul style="list-style-type: none"> <li>• Cisco 2-port OC-48c/STM16c POS SPA</li> <li>• Cisco 4-port OC-48c/STM16c POS SPA</li> </ul>
Release 3.4.1	Support was added on the Cisco CRS-1 Router for the Cisco 1-Port OC-192c/STM-64 POS/RPR VSR Optics SPA.

- [Prerequisites for Configuring POS Interfaces, on page 2](#)

- [Information About Configuring POS Interfaces, on page 2](#)
- [How to Configure a POS Interface, on page 5](#)
- [Configuration Examples for POS Interfaces, on page 12](#)

## 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 one of the following controller types:
  - A SONET controller, as described in the [Configuring Clear Channel T3/E3 Controllers and Channelized T3 and T1/E1 Controllers](#) module.
  - A DWDM controller, as described in the [Configuring Dense Wavelength Division Multiplexing Controllers](#) module.



---

**Note** POS DWDM controller configuration is supported on the Cisco 1-Port OC-768c/STM-256c DWDM PLIM only.

---

## Information About Configuring POS Interfaces

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

On the Cisco CRS-1 Router, a single POS interface carries data using PPP or Cisco HDLC encapsulation. Frame Relay is not supported on the Cisco CRS-1 Router.

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 this table are present. These default settings can be changed by configuration.

Table 1: POS Modular Services Card and PLIM Default Interface Settings

Parameter	Configuration File Entry	Default Settings
Keepalive	<b>keepalive</b> { <i>interval</i> [ <i>retry</i> ]   <b>disable</b> } <b>no keepalive</b>	Interval of 10 seconds Retry of: <ul style="list-style-type: none"> <li>• 5 (with PPP encapsulation)</li> <li>• 3 (with HDLC encapsulation)</li> </ul>
Encapsulation	<b>encapsulation</b> [hdlc   ppp]	<b>hdlc</b>
Maximum transmission unit (MTU)	<b>mtu</b> <i>bytes</i>	4474 bytes
Cyclic redundancy check (CRC)	<b>crc</b> [16   32]	<b>32</b>



**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 serial 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 the default encapsulation type for the serial interfaces.

Cisco HDLC uses keepalives to monitor the link state, as described in the [“Keepalive Timer” section on page 553](#).

## 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](#).

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 Authentication](#) 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).

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.

---

## 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.

#### Before you begin

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
<b>Step 1</b>	<b>show interfaces</b> <b>Example:</b> RP/0/RP0/CPU0:router# show interfaces	(Optional) Displays configured interfaces. <ul style="list-style-type: none"> <li>• Use this command to also confirm that the router recognizes the PLIM card.</li> </ul>
<b>Step 2</b>	<b>configure</b> <b>Example:</b> RP/0/RP0/CPU0:router# configure	Enters global configuration mode.
<b>Step 3</b>	<b>interface pos interface-path-id</b> <b>Example:</b> RP/0/RP0/CPU0:router(config)# interface POS 0/3/0/0	Specifies the POS interface name and notation <i>rack/slot/module/port</i> , and enters interface configuration mode.
<b>Step 4</b>	<b>ipv4 address ipv4_address/prefix</b> <b>Example:</b> RP/0/RP0/CPU0:router (config)# ipv4 address 10.46.8.6/24	Assigns an IP address and subnet mask to the interface. <p><b>Note</b></p> <ul style="list-style-type: none"> <li>• 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.</li> </ul>
<b>Step 5</b>	<b>no shutdown</b> <b>Example:</b> RP/0/RP0/CPU0:router (config-if)# no shutdown	Removes the shutdown configuration. <p><b>Note</b></p> <ul style="list-style-type: none"> <li>• 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).</li> </ul>
<b>Step 6</b>	<b>end</b> or <b>commit</b> <b>Example:</b> RP/0/RP0/CPU0:router (config-if)# end or	Saves configuration changes. <ul style="list-style-type: none"> <li>• When you issue the <b>end</b> command, the system prompts you to commit changes:   Uncommitted changes found, commit them before</li> </ul>

	Command or Action	Purpose
	RP/0/RP0/CPU0:router (config-if)# commit	<p>exiting (yes/no/cancel)? [cancel]:</p> <ul style="list-style-type: none"> <li>- Entering <b>yes</b> saves configuration changes to the running configuration file, exits the configuration session, and returns the router to EXEC mode.</li> <li>- Entering <b>no</b> exits the configuration session and returns the router to EXEC mode without committing the configuration changes.</li> <li>- Entering <b>cancel</b> leaves the router in the current configuration session without exiting or committing the configuration changes.</li> </ul> <ul style="list-style-type: none"> <li>• Use the <b>commit</b> command to save the configuration changes to the running configuration file and remain within the configuration session.</li> </ul>
<b>Step 7</b>	<p><b>exit</b></p> <p><b>Example:</b></p> <pre>RP/0/RP0/CPU0:router (config-if)# exit</pre>	Exits interface configuration mode and enters global configuration mode.
<b>Step 8</b>	<p><b>exit</b></p> <p><b>Example:</b></p> <pre>RP/0/RP0/CPU0:router (config)# exit</pre>	Exits global configuration mode and enters EXEC mode.
<b>Step 9</b>	<p>Repeat Step 1 through Step 8 to bring up the interface at the other end of the connection.</p> <p><b>Example:</b></p> <pre>RP/0/RP0/CPU0:router# show interfaces  RP/0/RP0/CPU0:router# configure RP/0/RP0/CPU0:router (config)# interface pos 0/3/0/0 RP/0/RP0/CPU0:router (config-if)# no shutdown RP/0/RP0/CPU0:router (config-if)# commit RP/0/RP0/CPU0:router (config-if)# exit RP/0/RP0/CPU0:router (config)# exit</pre>	<p><b>Note</b></p> <ul style="list-style-type: none"> <li>• The configuration on both ends of the POS connection must match.</li> </ul>
<b>Step 10</b>	<p><b>show ipv4 interface brief</b></p> <p><b>Example:</b></p> <pre>RP/0/RP0/CPU0:router # show ipv4 interface brief</pre>	<p>Verifies that the interface is active and properly configured.</p> <p>If you have brought up a POS interface properly, the “Status” field for that interface in the <b>show ipv4 interface brief</b> command output shows “Up.”</p>
<b>Step 11</b>	<p><b>show interfaces pos interface-path-id</b></p> <p><b>Example:</b></p>	(Optional) Displays the interface configuration.

	Command or Action	Purpose
	RP/0/RP0/CPU0:router# show interfaces pos 0/3/0/0	

### 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 594.

## Configuring Optional POS Interface Parameters

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

### Before you begin

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](#).

### 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 ]
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
<b>Step 1</b>	<b>configure</b> <b>Example:</b> RP/0/RP0/CPU0:router# configure	Enters global configuration mode.
<b>Step 2</b>	<b>interface pos</b> <i>interface-path-id</i> <b>Example:</b> RP/0/RP0/CPU0:router(config)# interface POS 0/3/0/0	Specifies the POS interface name and notation <i>rack/slot/module/port</i> , and enters interface configuration mode.



	Command or Action	Purpose
<b>Step 3</b>	<b>encapsulation [hdlc   ppp ]</b> <b>Example:</b> <pre>RP/0/RP0/CPU0:router(config-if)# encapsulation hdlc</pre>	(Optional) Configures the interface encapsulation parameters and details such as HDLC or PPP. <b>Note</b> <ul style="list-style-type: none"> <li>The default encapsulation is <b>hdlc</b>.</li> </ul>
<b>Step 4</b>	<b>pos crc {16   32}</b> <b>Example:</b> <pre>RP/0/RP0/CPU0:router(config-if)# pos crc 32</pre>	(Optional) Configures the CRC value for the interface. Enter the <b>16</b> keyword to specify 16-bit CRC mode, or enter the <b>32</b> keyword to specify 32-bit CRC mode. <b>Note</b> <ul style="list-style-type: none"> <li>The default CRC is <b>32</b>.</li> </ul>
<b>Step 5</b>	<b>mtu value</b> <b>Example:</b> <pre>RP/0/RP0/CPU0:router(config-if)# mtu 4474</pre>	(Optional) Configures the MTU value. <ul style="list-style-type: none"> <li>The default value is 4474.</li> <li>The POS MTU range is 64–9216.</li> </ul>
<b>Step 6</b>	<b>end or commit</b> <b>Example:</b> <pre>RP/0/RP0/CPU0:router (config-if)# end</pre> or <pre>RP/0/RP0/CPU0:router(config-if)# commit</pre>	Saves configuration changes. <ul style="list-style-type: none"> <li>When you issue the <b>end</b> command, the system prompts you to commit changes:  <pre>Uncommitted changes found, commit them before   exiting(yes/no/cancel)? [cancel]:</pre> <ul style="list-style-type: none"> <li>Entering <b>yes</b> saves configuration changes to the running configuration file, exits the configuration session, and returns the router to EXEC mode.</li> <li>Entering <b>no</b> exits the configuration session and returns the router to EXEC mode without committing the configuration changes.</li> <li>Entering <b>cancel</b> leaves the router in the current configuration session without exiting or committing the configuration changes.</li> </ul> </li> <li>Use the <b>commit</b> command to save the configuration changes to the running configuration file and remain within the configuration session.</li> </ul>
<b>Step 7</b>	<b>exit</b> <b>Example:</b> <pre>RP/0/RP0/CPU0:router (config-if)# exit</pre>	Exits interface configuration mode and enters global configuration mode.
<b>Step 8</b>	<b>exit</b> <b>Example:</b>	Exits global configuration mode and enters EXEC mode.

	Command or Action	Purpose
	RP/0/RP0/CPU0:router (config)# exit	
<b>Step 9</b>	<b>show interfaces pos</b> [ <i>interface-path-id</i> ] <b>Example:</b> RP/0/RP0/CPU0:router# show interface pos 0/3/0/0	(Optional) Displays general information for the specified POS interface.

### What to do next

- To configure PPP authentication on POS interfaces where PPP encapsulation is enabled, see the [Configuring PPP on the Cisco ASR 9000 Series Router](#) 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 601.

## 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.

Cisco HDLC is enabled by default on POS interfaces.

### Before you begin

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](#).

### 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 pos** *interface-path-id*

## DETAILED STEPS

	Command or Action	Purpose
<b>Step 1</b>	<b>configure</b> <b>Example:</b> RP/0/RP0/CPU0:router# configure	Enters global configuration mode.
<b>Step 2</b>	<b>interface pos <i>interface-path-id</i></b> <b>Example:</b> RP/0/RP0/CPU0:router(config)# interface POS 0/3/0/0	Specifies the POS interface name and notation <i>rack/slot/module/port</i> and enters interface configuration mode.
<b>Step 3</b>	<b>keepalive {<i>seconds</i> [<i>retry-count</i>]   <b>disable</b>}</b> or <b>no keepalive</b> <b>Example:</b> RP/0/RP0/CPU0:router(config-if)# keepalive 3 or RP/0/RP0/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. <ul style="list-style-type: none"> <li>Use the <b>keepalive disable</b> command, the <b>no keepalive</b>, or the <b>keepalive</b> command with an argument of 0 to disable the keepalive feature entirely.</li> </ul>
<b>Step 4</b>	<b>end</b> or <b>commit</b> <b>Example:</b> RP/0/RP0/CPU0:router(config-if)# end or RP/0/RP0/CPU0:router(config-if)# commit	Saves configuration changes. <ul style="list-style-type: none"> <li>When you issue the <b>end</b> command, the system prompts you to commit changes:  <pre>Uncommitted changes found, commit them before   exiting(yes/no/cancel)? [cancel]:</pre> <ul style="list-style-type: none"> <li>Entering <b>yes</b> saves configuration changes to the running configuration file, exits the configuration session, and returns the router to EXEC mode.</li> <li>Entering <b>no</b> exits the configuration session and returns the router to EXEC mode without committing the configuration changes.</li> <li>Entering <b>cancel</b> leaves the router in the current configuration session without exiting or committing the configuration changes.</li> </ul> </li> <li>Use the <b>commit</b> command to save the configuration changes to the running configuration file and remain within the configuration session.</li> </ul>
<b>Step 5</b>	<b>show interfaces pos <i>interface-path-id</i></b> <b>Example:</b> RP/0/RP0/CPU0:router# show interfaces POS 0/3/0/0	(Optional) Verifies the interface configuration.

# Configuration Examples for POS Interfaces

This section provides the following configuration examples:

## 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/RP0/CPU0:router# configure
RP/0/RP0/CPU0:router(config)# interface POS 0/3/0/0
RP/0/RP0/CPU0:router(config-if)# ipv4 address 172.18.189.38 255.255.255.224
RP/0/RP0/CPU0:router(config-if)# no shutdown
RP/0/RP0/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/RP0/CPU0:router# configure
RP/0/RP0/CPU0:router(config)# interface POS 0/3/0/0
RP/0/RP0/CPU0:router(config-if)# keepalive 10
RP/0/RP0/CPU0:router(config-if)# 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/RP0/CPU0:router# configure
RP/0/RP0/CPU0:router(config)# interface POS 0/3/0/0
RP/0/RP0/CPU0:router(config-if)# ipv4 address 172.18.189.38 255.255.255.224
RP/0/RP0/CPU0:router(config-if)# encapsulation ppp
RP/0/RP0/CPU0:router(config-if)# no shutdown
RP/0/RP0/CPU0:router(config-if)# end
Uncommitted changes found, commit them? [yes]: yes

RP/0/RP0/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 runs, 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
```

