# Pseudowire Emulation Edge-to-Edge MIBs for Ethernet, Frame Relay, and ATM Services

**First Published: August 25, 2004**
**Last Updated: June 29, 2007**

The Pseudowire Emulation Edge-to-Edge MIBs for Ethernet, Frame Relay, and ATM Services feature provides Simple Network Management Protocol (SNMP) support within an Any Transport over Multiprotocol Label Switching (AToM) infrastructure emulating Ethernet, Frame Relay, and ATM services over packet switched networks (PSNs). The Pseudowire Emulation Edge-to-Edge (PWE3) MIBs are the following:

- CISCO-IETF-PW-MIB (PW-MIB)
- CISCO-IETF-PW-MPLS-MIB (PW-MPLS-MIB)
- CISCO-IETF-PW-ENET-MIB (PW-ENET-MIB)
- CISCO-IETF-PW-FR-MIB (PW-FR-MIB)
- CISCO-IETF-PW-ATM-MIB (PW-ATM-MIB)

Cisco IOS Release 12.2(28)SB introduces support for the CISCO-IETF-PW-ATM-MIB (PW-ATM-MIB), which provides network management information specific to an ATM over pseudowire connection in a Multiprotocol Label Switching (MPLS) AToM or an IP network.

### Finding Feature Information in This Module

Your Cisco IOS software release may not support all of the features documented in this module. To reach links to specific feature documentation in this module and to see a list of the releases in which each feature is supported, use the "Feature Information for Pseudowire Emulation Edge-to-Edge MIBs for Ethernet, Frame Relay, and ATM Services" section on page 28.

### Finding Support Information for Platforms and Cisco IOS and Catalyst OS Software Images

Use Cisco Feature Navigator to find information about platform support and Cisco IOS and Catalyst OS software image support. To access Cisco Feature Navigator, go to http://www.cisco.com/go/cfn. An account on Cisco.com is not required.

**Americas Headquarters:**
**Cisco Systems, Inc., 170 West Tasman Drive, San Jose, CA 95134-1706 USA**

# Contents



# Prerequisites for Pseudowire Emulation Edge-to-Edge MIBs for Ethernet, Frame Relay, and ATM Services

- SNMP must be enabled on the label switch routers (LSRs).
- MPLS must be enabled on the LSRs.
- Pseudowires must be configured with Ethernet, Frame Relay, or ATM access circuits. (For more detailed information, see the *Any Transport over MPLS* feature module; the *Cisco IOS Wide-Area Networking Configuration Guide*, Release 12.4; and the *Cisco IOS Asynchronous Transfer Mode Configuration Guide*, Release 12.4.)

# Restrictions for Pseudowire Emulation Edge-to-Edge MIBs for Ethernet, Frame Relay, and ATM Services

The PWE3 MIBs are limited to read-only (RO) permission for MIB objects except for the cpwVcUp and cpwVcDown notification enable object, cpwVcUpDownNotifEnable, which has been extended to be writable by the SNMP agent.

If you use the **encapsulation frame-relay** command to configure the CISCO-IETF-PW-FR-MIB, then the **xconnect** command is not supported.

- The following tables in the PW-MIB are not supported:
  - cpwVcPerfCurrentTable
  - cpwVcPerfIntervalTable
- The following objects in the PW-MPLS-MIB are not supported:
  - cpwVcMplsOutboundIndexNext

  - cpwVcMplsInboundIndexNext
- The following tables in the PW-ENET-MIB are not supported:
  - cpwVcEnetMplsPriMappingTable
  - cpwVcEnetStatsTable
- The following table in the PW-FR-MIB is not supported:
  - cpwVcFrPMTable
- The PW-ATM-MIB does not support cell counters on the Cisco 7500 series router. Consequently, an SNMP query for cell counters returns a 0 value.
- The PW-ATM-MIB does not support a high-capacity cell counter per virtual path (VP) or cells per port.
- The PW-ATM-MIB virtual path identifier (VPI)/virtual channel identifier (VCI) value for port mode cell relay is 0.
- The PW-ATM-MIB VP cell relay VCI value is 0.
- The PW-ATM-MIB VP does not support ATM adaptation layer 5 (AAL5); therefore, all packet counters are invalid.

**Note** This feature is not supported over Ethernet, Frame Relay, and ATM in all releases. See the "Feature Information for Pseudowire Emulation Edge-to-Edge MIBs for Ethernet, Frame Relay, and ATM Services" section on page 28 for more detailed information.

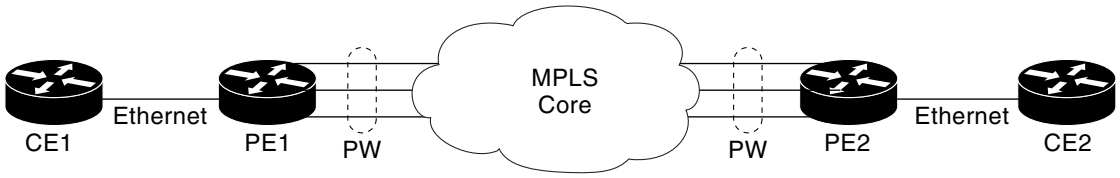
# Information About Pseudowire Emulation Edge-to-Edge MIBs for Ethernet, Frame Relay, and ATM Services

To configure the Pseudowire Emulation Edge-to-Edge MIBs for Ethernet, Frame Relay, and ATM Services feature, you should understand the following concepts:

# The Function of a Pseudowire in the PWE3 MIBs

A pseudowire is a point-to-point connection between pairs of provider edge (PE) routers (as shown in Figure 1). Its primary function is to emulate services like Ethernet over an underlying core MPLS network through encapsulation into a common MPLS format. By encapsulating services into a common MPLS format, a pseudowire allows carriers to converge their services to an MPLS network.

***Figure 1 Sample Pseudowire Topology***



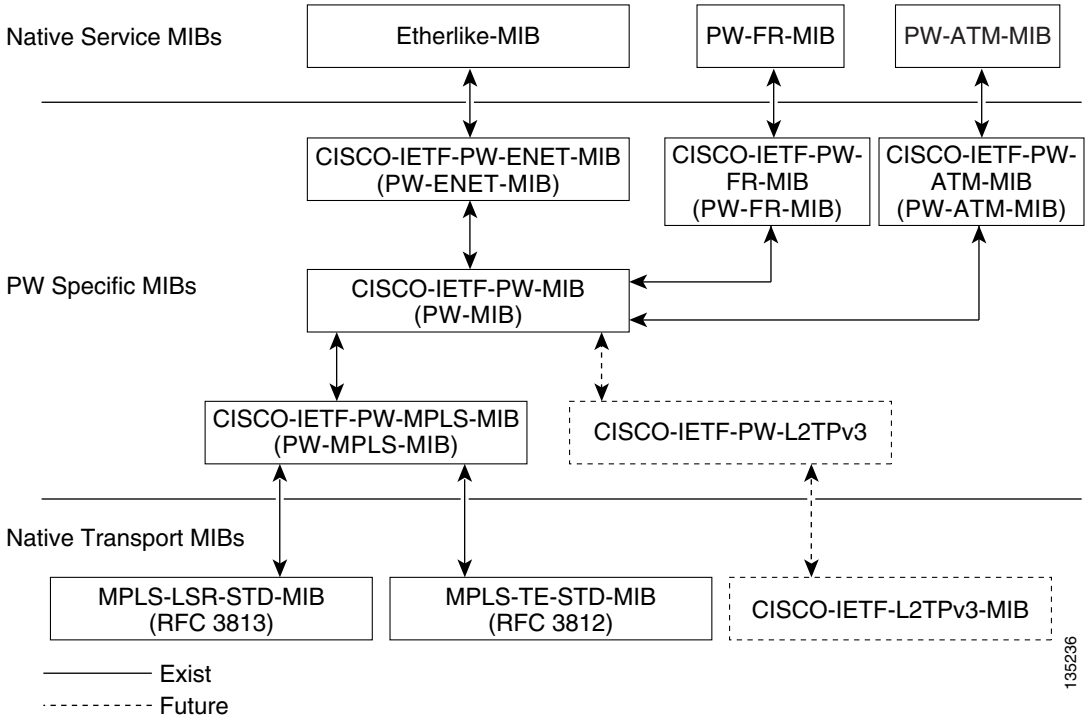

PW = Pseudowire
CE = Customer Edge Router
PE = Provider Edge Router
MPLS = Multiprotocol Label Switching

121002

# PWE3 MIBs Architecture

The PWE3 MIBs architecture shown in Figure 2 categorizes three groups of MIBs that, when used together, provide the complete emulated service; the native transport, which carries the service across the core network; and the relationship between the two.

***Figure 2 PWE3 MIBs Architecture***




135236

The architecture is modular in that once deployed, new emulated service MIB modules or additional transport MIB modules "plug in" to or extend the existing infrastructure rather than require a new and unique one. This allows you to build management applications without the concern of a new service requiring the deployment of a completely different management strategy. Because the architecture is a generalized association mechanism between existing service and transport MIB modules, native MIB modules work in the absence of the associated PWE3-specific MIBs. The advantage is that if a PWE3-specific MIB has not yet been deployed in Cisco IOS software, which associates a service or transport with pseudowires, these MIB modules can still be queried. However, the only drawback is that the associations with the pseudowires are absent.

# Components and Functions of the PWE3 MIBs

The PWE3 MIBs have the following components and functions:

- PW-MIB (the pseudowire MIB)

  This MIB binds the PW-MPLS-MIB and the PW-ENET-MIB together, and provides status of the pseudowire emulation and statistics and configuration information. The PW-MIB also defines the notifications for pseudowire fault and event monitoring.

- PW-MPLS-MIB (the pseudowire MPLS-MIB)

  This MIB contains managed objects that can be used by a network manager to monitor pseudowire emulation MPLS services, such as MPLS-traffic engineering (TE)-PSN and MPLS-non-TE-PSN.

  This MIB shows the following:

  - Cross-connect (XC) indexes for virtual circuits (VCs) that are Label Distribution Protocol (LDP)-signaled and have a preferred path that is not set to an MPLS TE tunnel.
  - Tunnel indexes for VCs with a preferred path set to a TE tunnel and an output interface that is a TE tunnel.

- PW-ENET-MIB (the pseudowire Ethernet services MIB)

  This MIB contains managed objects that can be used by a network manager to monitor pseudowire emulation Ethernet services.

- PW-FR-MIB (the pseudowire Frame Relay services MIB)

  This MIB contains managed objects that can be used by a network manager to monitor pseudowire emulation Frame Relay services.

  This MIB uses a Frame Relay over pseudowire (FRoPW) connection that consists of two segments: the Frame Relay segment and the pseudowire segment. The PW-FR-MIB provides hooks to those segments. The PW MIB contains information about the pseudowire segment, and the PW-FR-MIB contains information about the Frame Relay segment.

  The PW-FR-MIB is defined at the Pseudowire Service Emulation Layer and resides on top of the generic PW-MIB as shown in Figure 2. Therefore, the PW-FR-MIB is highly dependent on the existence and the service provided by the PW-MIB. In addition, an existing PW-FR connection entry must associate with an existing VC entry in the PW-MIB.

  The PW-FR-MIB and the generic PW-MIB are logically tied by the PW VC Index, which is an internal index defined to support the PW-MIB. Each PW VC index uniquely maps into an existing VC entry in the PW-MIB and the PW-FR-MIB.

- PW-ATM-MIB (the pseudowire ATM services MIB)

  This MIB contains managed objects that can be used by a network manager to monitor pseudowire emulation ATM services.

This MIB uses an ATM over pseudowire (ATMoPW) connection that consists of two segments: the ATM segment and the pseudowire segment. The PW-ATM-MIB provides hooks to those segments. The PW MIB contains information about the pseudowire segment, and the PW-ATM-MIB contains information about the ATM segment called the attachment circuit.

The PW-ATM-MIB is defined at the Pseudowire Service Emulation Layer and resides on top of the generic PW-MIB as shown in Figure 2. Therefore, the PW-ATM-MIB is highly dependent on the existence and the service provided by the PW-MIB. In addition, an existing PW-ATM connection entry must associate with an existing VC entry in the PW-MIB.

The PW-ATM-MIB and the generic PW-MIB are logically tied by the PW VC Index, which is an internal index defined to support the PW-MIB. Each PW VC index uniquely maps into an existing VC entry in the PW-MIB and the PW-ATM-MIB.

# Tables in the PW-MIB

The PW-MIB consists of the following tables:

- cpwVcTable (Table 1)—Contains high-level generic parameters related to VC creation. This table is implemented as read only and is indexed by the cpwVcIndex, which uniquely identifies a singular connection. A row in this table represents an emulated virtual connection. This table is used for all VC types.
- cpwVcPerfTotalTable (Table 2)—Provides per-VC performance information from the VC start time. This table is indexed by the cpwVcIndex.
- cpwVcIdMappingTable (Table 3)—Provides reverse mapping of the existing VCs based on VC type and VC ID ordering. This table is typically useful for element manager software (EMS) ordered query of existing VCs. This table is indexed by cpwVcIdMappingVcType, cpwVcIdMappingVcID, cpwVcIdMappingPeerAddrType, and cpwVcIdMappingPeerAddr. This table is implemented as read only.
- cpwVcPeerMappingTable (Table 4)—Provides reverse mapping of the existing VCs based on VC type and VC ID ordering. This table is typically useful for EMS ordered query of existing VCs. This table is indexed by cpwVcPeerMappingPeerAddrType, cpwVcPeerMappingPeerAddr, cpwVcPeerMappingVcType, and cpwVcPeerMappingVcID. This table is implemented as read only.

## cpwVcTable

Table 1 lists the cpwVcTable objects and their descriptions.

*Table 1 cpwVcTable Objects and Descriptions*

| Objects | Description |
|---|---|
| cpwVcType | Indicates the service to be carried over this VC. This is circuit type information. |
| cpwVcOwner | Set by the operator to indicate the protocol responsible for establishing this VC. Values are the following:<br>• manual(1)—Used when no maintenance protocol (PW signaling) is needed to set up the VC, such as configuration of entries in the VC tables including VC labels, and so forth.<br>• maintenanceProtocol(2)—Used for standard signaling of the VC for the specific PSN; for example, LDP for MPLS PSN as specified in *draft-martini-l2circuit-trans-mpls* or the Layer 2 Tunneling Protocol (L2TP).<br>• other(3)—Used for all other types of signaling. |
| cpwVcPsnType | Set by the operator to indicate the PSN type on which this VC is carried. Based on this object, the relevant PSN table entries are created in the PSN-specific MIB modules. For example, if mpls(1) is defined, the agent creates an entry in the cpwVcMplsTable, which further defines the MPLS PSN configuration. |
| cpwVcSetUpPriority | Defines the relative setup priority of the VC in a lowest-to-highest manner, where 0 is the highest priority. This value is significant if there are competing resources between VCs and the implementation supports this feature. Because this is not implemented in AToM, the value of 0 is used. |
| cpwVcHoldingPriority | Defines the relative holding priority of the VC in a lowest-to-highest manner, where 0 is the highest priority. This value is significant if there are competing resources between VCs and the implementation supports this feature. Because this is not implemented in AToM, the value of 0 is used. |
| cpwVcInboundMode | Enables greater security for implementations that use per-platform VC label space. Modes are the following:<br>• strict(1)<br>• loose(2)<br>In strict mode, packets coming from the PSN are accepted only from tunnels that are associated to the same VC via the inbound tunnel table in the case of MPLS, or as identified by the source IP address in the case of L2TP or IP PSN. The entries in the inbound tunnel table are either explicitly configured or implicitly known by the maintenance protocol used for VC setup.<br>If such association is not known, not configured, or not desired, loose mode should be configured, and the node should accept the packet based on the VC label only, regardless of the outer tunnel used to carry the VC. |

*Table 1 cpwVcTable Objects and Descriptions (continued)*

| Objects | Description |
|---|---|
| cpwVcPeerAddrType | Denotes the address type of the peer node maintenance protocol (signaling) address if the PW maintenance protocol is used for the VC creation. It should be set to unknown if the PW maintenance protocol is not used; for example, cpwVcOwner is set to manual. |
| cpwVcPeerAddr | Contains the value of the peer node address of the PW maintenance protocol entity. This object should contain a value of 0 if not relevant (manual configuration of the VC). |
| cpwVcID | Use in the outgoing VC ID field within the VC forward equivalence class (FEC) element with LDP signaling or the PW ID attribute-value (AV) pair for the L2TP. |
| cpwVcLocalGroupID | Use in the Group ID field sent to the peer PW within the maintenance protocol for VC setup; 0 if not used. |
| cpwVcControlWord | Defines if the control word is sent with each packet by the local node. |
| cpwVcLocalIfMtu | If not = 0, the optional IfMtu object in the maintenance protocol is sent with this value, representing the locally supported maximum transmission unit (MTU) size over the interface (or the virtual interface) associated with the VC. |
| cpwVcLocalIfString | Each VC is associated to an interface (or a virtual interface) in the ifTable of the node as part of the service configuration. This object defines if the maintenance protocol sends the interface's name as it appears in the ifTable in the name object as part of the maintenance protocol. If this object is set to false, the optional element is not sent. |
| cpwVcRemoteGroupID | Obtained from the Group ID field as received via the maintenance protocol used for VC setup; 0 if not used. The value of 0xFFFF is used if the object is not defined by the VC maintenance protocol. |
| cpwVcRemoteControlWord | If the maintenance protocol is used for VC establishment, this parameter indicates the received status of the control word usage; that is, if packets are received with the control word or not. The value of notYetKnown is used while the maintenance protocol has not yet received the indication from the remote node. In a manual configuration of the VC, this parameter indicates to the local node the expected encapsulation for the received packets. |
| cpwVcRemoteIfMtu | The remote interface MTU as received from the remote node via the maintenance protocol. This object should be 0 if this parameter is not available or not used. |
| cpwVcRemoteIfString | Indicates the interface description string as received by the maintenance protocol; it must be a NULL string if not applicable or not known yet. |

*Table 1 cpwVcTable Objects and Descriptions (continued)*

| Objects | Description |
|---|---|
| cpwVcOutboundVcLabel | The VC label used in the outbound direction toward the PSN. This object may be set up manually if the owner is manual; otherwise, it is automatic. Examples; for MPLS PSN, the label represents the 20 bits of the VC tag; for L2TP, it represents the 32 bits of the session ID. If the label is not yet known (signaling in process), the object should return a value of 0xFFFF. |
| cpwVcInboundVcLabel | The VC label used in the inbound direction for packets received from the PSN. This object may be set up manually if the owner is manual; otherwise, it is automatic. Examples; for MPLS PSN, the label represents the 20 bits of VC tag; for L2TP, the label represents the 32 bits of the session ID. If the label is not yet known (signaling in process), the object should return a value of 0xFFFF. |
| cpwVcName | The canonical name assigned to the VC. |
| cpwVcDescr | A textual string containing information about the VC. If there is no description, this object contains a 0 length string. |
| cpwVcCreateTime | System time when this VC was created. |
| cpwVcUpTime | Number of consecutive ticks that this VC has been up in both directions together. (Up is observed in cpwVcOperStatus.) |
| cpwVcAdminStatus | The desired operational status of this VC. |
| cpwVcOperStatus | Indicates the actual combined operational status of this VC. This object is up if both cpwVcInboundOperStatus and cpwVcOutboundOperStatus are in the up state. For all other values, if the VCs in both directions are of the same value, this object reflects that value; otherwise, it is set to the more severe status of the two. The order of severity from most severe to less severe is as follows: unknown, notPresent, down, lowerLayerDown, dormant, testing, and up. The operator can consult the direction of OperStatus for fault isolation. |
| cpwVcInboundOperStatus | Indicates the actual operational status of this VC in the inbound direction. Values are the following:<br>• up—The VC is established and ready to pass packets.<br>• down—PW signaling has not yet finished or indications available at the service level show that the VC is not passing packets.<br>• testing—AdminStatus at the VC level is set to test.<br>• dormant—The VC is not available because the required resources are occupied by higher priority VCs.<br>• notPresent—Some component needed for the setup of the VC is missing.<br>• lowerLayerDown—The underlying PSN is not in OperStatus up. |

*Table 1 cpwVcTable Objects and Descriptions (continued)*

| Objects | Description |
|---|---|
| cpwVcOutboundOperStatus | Indicates the actual operational status of this VC in the outbound direction. Values are the following:<br>• up—The VC is established and ready to pass packets.<br>• down—PW signaling has not yet finished or indications available at the service level show that the VC is not passing packets.<br>• testing—AdminStatus at the VC level is set to test.<br>• dormant—The VC is not available because the required resources are occupied by higher priority VCs.<br>• notPresent—Some component needed for the setup of the VC is missing.<br>• lowerLayerDown—The underlying PSN is not in OperStatus up. |
| cpwVcTimeElapsed | The number of seconds, including partial seconds, that have elapsed since the beginning of the current measurement period. If, for some reason, such as an adjustment in the system's time-of-day clock, and the current interval exceeds the maximum value, the agent returns the maximum value. Because cpwVcPerfIntervalTable is not implemented, this is 0. |
| cpwVcValidIntervals | The number of previous 15-minute intervals for which data was collected. An agent with PW capability must be capable of supporting at least *x* intervals. The minimum value of *x* is 4; the default of *x* is 32, and the maximum value of *x* is 96. The value is *x* unless the measurement was (re)started within the last *x**15 minutes, in which case the value will be the number of complete 15-minute intervals; for example, in the case where the agent is a proxy, some intervals may be unavailable. In this case, this interval is the maximum interval number for which data is available. This interval is set to 0. |
| cpwVcRowStatus | A read-only implementation that is always active(1). It is used for creating, modifying, and deleting. |
| cpwVcStorageType | The storage type for this object is a read-only implementation that is always volatile(2). |

## cpwVcPerfTotalTable

Table 2 lists the cpwVcPerfTotalTable objects and their descriptions.

*Table 2 cpwVcPerfTotalTable Objects and Descriptions*

| Objects | Description |
|---|---|
| cpwVcPerfTotalInHCPackets | High-capacity counter for the number of packets received by the VC from the PSN. |
| cpwVcPerfTotalInHCBytes | High-capacity counter for the number of bytes received by the VC from the PSN. |
| cpwVcPerfTotalOutHCPackets | High-capacity counter for the number of packets forwarded by the VC to the PSN. |
| cpwVcPerfTotalOutHCBytes | High-capacity counter for the number of bytes forwarded by the VC (to the PSN). |
| cpwVcPerfTotalDiscontinuityTime | The value of sysUpTime on the most recent occasion when one or more of this object's counters suffered a discontinuity. The relevant counters are the specific instances of any Counter32 or Counter64. If no such discontinuities have occurred since the last reinitialization of the local management subsystem, this object contains a 0 value. |

## cpwVcIdMappingTable

Table 3 lists the cpwVcIdMappingTable objects and their descriptions.

*Table 3 cpwVcIdMappingTable Objects and Descriptions*

| Objects | Description |
|---|---|
| cpwVcIdMappingVcType | The VC type (indicates the service) of this VC. |
| cpwVcIdMappingVcID | The VC ID of this VC; 0 if the VC is configured manually. |
| cpwVcIdMappingPeerAddrType | IP address type of the peer node. |
| cpwVcIdMappingPeerAddr | IP address of the peer node. |
| cpwVcIdMappingVcIndex | The value that represents the VC in the cpwVcTable. |

## cpwVcPeerMappingTable

Table 4 lists the cpwVcPeerMappingTable objects and their descriptions.

*Table 4 cpwVcPeerMappingTable Objects and Descriptions*

| Objects | Description |
|---|---|
| cpwVcPeerMappingPeerAddrType | IP address type of the peer node. |
| cpwVcPeerMappingPeerAddr | IP address of the peer node. |
| cpwVcPeerMappingVcType | The VC type (indicates the service) of this VC. |
| cpwVcPeerMappingVcID | The VC ID of this VC; 0 if the VC is configured manually. |
| cpwVcPeerMappingVcIndex | The value that represents the VC in the cpwVcTable. |

# Tables in the PW-MPLS-MIB

The PW-MPLS-MIB consists of the following tables:

- cpwVcMplsTable (Table 5)—Specifies information for the VC to be carried over an MPLS PSN. This table is indexed on cpwVcIndex.
- cpwVcMplsOutboundTable (Table 6)—Associates VCs using an MPLS PSN with the outbound MPLS tunnels toward the PSN or the physical interface in the case of the VC only. A row in this table represents a link between PW VCs that require MPLS tunnels and an MPLS tunnel toward the PSN. This table is indexed by the cpwVcIndex and an additional index that is not supported; consequently, its value is 1. The operator creates at least one entry in this table for each PW VC that requires an MPLS PSN. The VC-only case and the cpwVcMplsOutboundIndex is not supported.
- cpwVcMplsInboundTable (Table 7)—Associates VCs using an MPLS PSN with the inbound MPLS tunnels for packets coming from the PSN, if such association is desired mainly for security reasons. A row in this table represents a link between PW VCs that require MPLS tunnels and an MPLS tunnel for packets arriving from the PSN. This table is indexed by the set of indexes used to identify the VC, cpwVcIndex, and an additional index that is not supported; consequently, its value is 1. An entry is created in this table either automatically by the local agent or manually by the operator when strict mode is required. This table points to the appropriate MPLS MIB. For MPLS-TE, the four variables relevant to the indexing of an MPLS TE tunnel are set. The VC-only case and the cpwVcMplsInboundIndex are not supported.
- cpwVcMplsNonTeMappingTable (Table 8)—Maps an inbound or outbound tunnel to a VC in non-TE applications. A row in this table represents the association between a PW VC and its non-TE MPLS outer tunnel. An application can use this table to retrieve the PW carried over a specific non-TE MPLS outer tunnel quickly. This table is indexed by the xconnect index for the MPLS non-TE tunnel and the direction of the VC in the specific entry. The same table is used in both inbound and outbound directions, but in a different row for each direction. If the inbound association is not known, no rows should exist for it. Rows are created by the local agent when all the association data is available for display.
- cpwVcMplsTeMappingTable (Table 9)—Maps an inbound or outbound tunnel to a VC in MPLS-TE applications. A row in this table represents the association between a PW VC and its MPLS-TE outer tunnel. An application can use this table to retrieve the PW carried over a specific TE MPLS outer tunnel quickly. This table is indexed by the four indexes of a TE tunnel, the direction of the VC specific entry, and the VcIndex. The same table is used in both inbound and outbound directions, but a different row for each direction. If the inbound association is not known, no rows should exist for it. Rows are created by the local agent when all the association data is available for display. This table shows mappings between pseudowires and the xconnect index for non-TE outer tunnel or index.

## cpwVcMplsTable

Table 5 lists the cpwVcMplsTable objects and their descriptions.

*Table 5 cpwVcMplsTable Objects and Descriptions*

| Objects | Description |
|---|---|
| cpwVcMplsMplsType | Set by the operator to indicate the outer tunnel types, if they exist. Values are the following:<br>• mplsTe(0)—Used when the outer tunnel is set up by MPLS-TE.<br>• mplsNonTe(1)—Used when the outer tunnel is set up by LDP or manually. |
| cpwVcMplsExpBitsMode | Set by the operator to indicate the way the VC shim label EXP bits are to be determined. The value is the following:<br>• outerTunnel(1)—Used when there is an outer tunnel and cpwVcMplsMplsType is mplsTe or mplsNonTe. |
| cpwVcMplsExpBits | Set by the operator to indicate the MPLS EXP bits to be used on the VC shim label if cpwVcMplsExpBitsMode is specified; value = 0. |
| cpwVcMplsTtl | Set by the operator to indicate the VC time-to-live (TTL) bits to be used on the VC shim label; value = 0. |
| cpwVcMplsLocalLdpID | The local LDP identifier of the LDP entity creating this VC in the local node. Because the VC labels are always set from the per-platform label space, the last two octets in the LDP ID must be 0s. |
| cpwVcMplsLocalLdpEntityID | The local LDP entity index of the LDP entity to be used for this VC on the local node; this should be set to all 0s when this object is not used. |
| cpwVcMplsPeerLdpID | The peer LDP identifier as identified by the LDP session; this should be zero if not relevant or not known yet. |
| cpwVcMplsStorageType | The storage type for this object is a read-only implementation that is always volatile(2). |

## cpwVcMplsOutboundTable

Table 6 lists the cpwVcMplsOutboundTable objects and their descriptions.

*Table 6 cpwVcMplsOutboundTable Objects and Descriptions*

| Objects | Description |
|---|---|
| cpwVcMplsOutboundIndex | An arbitrary index for enabling multiple rows per VC in this table. The next available free index can be retrieved using cpwVcMplsOutboundIndexNext. The value = 1, because this object is not supported. |
| cpwVcMplsOutboundLsrXcIndex | Set by the operator. If the outer label is defined in the MPL-LSR-MIB, that is, set by LDP or manually, this object points to the xconnect index of the outer tunnel. Otherwise, this object is set to 0. |

*Table 6 cpwVcMplsOutboundTable Objects and Descriptions (continued)*

| Objects | Description |
|---|---|
| cpwVcMplsOutboundTunnelIndex | Part of the set of indexes for an outbound tunnel, specifically an MPLS-TE outer tunnel; otherwise, this object is set to 0. |
| cpwVcMplsOutboundTunnelInstance | Part of the set of indexes for an outbound tunnel, specifically an MPLS-TE outer tunnel; otherwise, this object is set to 0. |
| cpwVcMplsOutboundTunnelLclLSR | Part of the set of indexes for an outbound tunnel, specifically an MPLS-TE outer tunnel; otherwise, this object is set to NULL. |
| cpwVcMplsOutboundTunnelPeerLSR | Part of the set of indexes for an outbound tunnel, specifically an MPLS-TE outer tunnel; otherwise, this object is set to NULL. |
| cpwVcMplsOutboundIfIndex | For a VC only with no outer tunnel, this object holds the ifIndex of the outbound port. The value = 0. |
| cpwVcMplsOutboundRowStatus | A read-only implementation that is always active(1). It is used for creating, modifying, and deleting. |
| cpwVcMplsOutboundStorageType | The storage type for this object is a read-only implementation that is always volatile(2). |

## cpwVcMplsInboundTable

Table 7 lists the cpwVcMplsInboundTable objects and their descriptions.

*Table 7 cpwVcMplsInboundTable Objects and Descriptions*

| Objects | Description |
|---|---|
| cpwVcMplsInboundIndex | An arbitrary index for enabling multiple rows per VC in this table. The next available free index can be retrieved using cpwVcMplsInboundIndexNext. the value = 1, because this object is not supported. |
| cpwVcMplsInboundLsrXcIndex | If the outer label is defined in the MPL-LSR-MIB; that is, set by LDP or manually, this object points to the xconnect index of the outer tunnel. The xconnect index represents the pseudowire in the inbound direction retrieving 0 if information for this object is not known. |
| cpwVcMplsInboundTunnelIndex | Part of the set of indexes for an inbound tunnel, specifically an MPLS-TE outer tunnel; value = 0. This object does not support TE tunnels at the ingress router. |
| cpwVcMplsInboundTunnelInstance | Part of the set of indexes for an inbound tunnel, specifically an MPLS-TE outer tunnel; value = 0. This object does not support TE tunnels at the ingress router. |
| cpwVcMplsInboundTunnelLclLSR | Part of the set of indexes for an inbound tunnel, specifically an MPLS-TE outer tunnel; otherwise, set to NULL. This object does not support TE tunnels at the ingress router. |

*Table 7 cpwVcMplsInboundTable Objects and Descriptions (continued)*

| Objects | Description |
|---|---|
| cpwVcMplsInboundTunnelPeerLSR | Part of the set of indexes for an inbound tunnel, specifically an MPLS-TE outer tunnel; otherwise, this object is set to NULL. This object does not support TE tunnels at the ingress router. |
| cpwVcMplsInboundIfIndex | In the case of a VC only (no outer tunnel), this object holds the ifIndex of the inbound port. The value = 0. |
| cpwVcMplsInboundRowStatus | A read-only implementation that is always active(1). It is used for creating, modifying, and deleting. |
| cpwVcMplsInboundStorageType | The storage type for this object is a read-only implementation that is always volatile(2). |

## cpwVcMplsNonTeMappingTable

Table 8 lists the cpwVcMplsNonTeMappingTable objects and their descriptions.

*Table 8 cpwVcMplsNonTeMappingTable Objects and Descriptions*

| Objects | Description |
|---|---|
| cpwVcMplsNonTeMappingTunnelDirection | Identifies if the row represents an outbound or inbound mapping. |
| cpwVcMplsNonTeMappingXcTunnelIndex | XC index in the MPLS-LSR-MIB of the pseudowire LDP-generated XC entry. |
| cpwVcMplsNonTeMappingIfIndex | Identifies the port on which the VC is carried for VC only; the value = 0. |
| cpwVcMplsNonTeMappingVcIndex | Represents the VC in the cpwVcTable. |

## cpwVcMplsTeMappingTable

Table 9 lists the cpwVcMplsTeMappingTable objects and their descriptions.

*Table 9 cpwVcMplsTeMappingTable Objects and Descriptions*

| Objects | Description |
|---|---|
| cpwVcMplsTeMappingTunnelDirection | Identifies if the row represents an outbound mapping. |
| cpwVcMplsTeMappingTunnelIndex | Index for the conceptual row identifying an MPLS-TE tunnel. |
| cpwVcMplsTeMappingTunnelInstance | Identifies an instance of an MPLS-TE tunnel. |
| cpwVcMplsTeMappingTunnelPeerLsrID | Identifies a peer LSR when the outer tunnel is MPLS-TE based. |
| cpwVcMplsTeMappingTunnelLocalLsrID | Identifies the local LSR. |
| cpwVcMplsTeMappingVcIndex | Represents the VC in the cpwVcTable. |

# Tables in the PW-ENET-MIB

The PW-ENET-MIB consists of the following table:

- cpwVcEnetTable (Table 10)—Provides Ethernet port mapping and VLAN configuration for each Ethernet emulated virtual connection. This table is indexed on cpwVcIndex, which uniquely identifies a singular connection. The second level index for this table is cpwVcEnetPwVlan, which indicates VLANs on this VC. This table is used only for Ethernet VC types—ethernetVLAN, ethernet, or ethernet virtual private LAN service (VPLS), and is implemented as read-only.

## cpwVcEnetTable

Table 10 lists the cpwVcEnetTable objects and their descriptions.

***Table 10 cpwVcEnetTable Objects and Descriptions***

| Objects | Description |
|---|---|
| cpwVcEnetPwVlan | The VLAN value for frames on a VC. This is one of the indexes to the table so multiple VLAN values can be configured for a PW VC. This value is 4096 to indicate untagged frames; that is, if the cpwVcEnetVlanMode value is removeVlan. This value is the VLAN value of the access circuit if the cpwVcEnetVlanMode value is noChange. The value of 4097 is used if the object is not applicable; for example, when mapping all packets from an Ethernet port to the VC. |
| cpwVcEnetVlanMode | Indicates the way the VLAN field is handled between the access circuit and the PW VC. The possible values for this field are as follows:<br>• noChange—Indicates that the VC contains the original user VLAN, as specified in cpwVcEnetPortVlan.<br>• changeVlan—Indicates that the VLAN field on the VC may be different from the VLAN field on the user's port.<br>• removeVlan—Indicates that the encapsulation on the VC does not include the original VLAN field.<br>• addVlan—Indicates that a VLAN field is added on the PSN bound direction (that is, on the PW). |
| cpwVcEnetPortVlan | Defines the VLAN value on the physical port (or VPLS virtual port) if a change is required to the VLAN value between the VC and the physical or virtual port. It is equal to cpwVcEnetPwVlan if the cpwVcEnetVlanMode value is noChange. A value of 4096 indicates that no VLAN is associated with the VC; that is, assigning Default VLAN to untagged frames. If all traffic from the VC is being forwarded to the port, then this value is 4097 indicating it is not relevant. |

*Table 10 cpwVcEnetTable Objects and Descriptions (continued)*

| Objects | Description |
|---|---|
| cpwVcEnetPortIfIndex | The ifIndex value of the Ethernet port associated with this PW VC for point-to-point Ethernet service. For VPLS, this value is an ifIndex value for a virtual interface for the VPLS instance. |
| cpwVcEnetVcIfIndex | Models the VC as a virtual interface in the ifTable. This value is always 0 to indicate no virtual interface is created. |
| cpwVcEnetRowStatus | A read-only implementation that is always active(1). It is used for creating, modifying, and deleting. |
| cpwVcEnetStorageType | The storage type for this object is a read-only implementation that is always volatile(2). |

# Tables in the PW-FR-MIB

The PW-FR-MIB consists of the following table:

- cpwVcFrTable (Table 11)—Contains entries that represent an FRoPW connection operating in one-to-one mapping mode in which there is a one-to-one correspondence between a Frame Relay VC and a pair of unidirectional pseudowires.

## cpwVcFrTable

Table 11 lists the cpwVcFrTable objects and their descriptions.

*Table 11 cpwVcFrTable Objects and Descriptions*

| Objects | Description |
|---|---|
| cpwVcFrIfIndex | Returns the interface ifIndex of the Frame Relay (FR) segment of the FRoPW connection. |
| cpwVcFrDlci | Returns the data-link connection identifier (DLCI) of the Frame Relay segment of an FRoPW connection. |
| cpwVcFrAdminStatus | Returns the administrative status of an FRoPW connection. |
| cpwVcFrOperStatus | Returns the combined operational status of an FRoPW connection. |
| cpwVcFrPw2FrOperStatus | Returns the operational status of the PW-to-FR direction in an FRoPW connection. |
| cpwVcFrRowStatus | A read-only implementation that is always active(1). It is used for creating, modifying, and deleting. |
| cpwVcFrStorageType | The storage type for this object is a read-only implementation that is always volatile(2). |

# Tables in the PW-ATM-MIB

The PW-ATM-MIB consists of the following tables:

- cpwVcAtmTable (Table 12)—Specifies information for an ATM VC to be carried over the PSN.
- cpwVcAtmPerfTable (Table 13)—Specifies performance-related attributes for an ATM VC.

## cpwVcAtmTable

Table 12 lists the cpwVcAtmTable objects and their descriptions.

***Table 12 cpwVcAtmTable Objects and Descriptions***

| Objects | Description |
|---|---|
| cpwAtmIf | Specifies the ATM interface that sends and receives cells from the ATM network. |
| cpwAtmVpi | Specifies the VPI value of the ATM VC. |
| cpwAtmVci | Specifies the VCI value of the ATM VC. |
| cpwAtmClpQosMapping | Indicates the presence of cell loss priority (CLP) bits determining the value in quality of service (QoS) fields of the encapsulating protocol. The value could be used only for outbound traffic, which means traffic going out to the PSN. |
| cpwAtmRowStatus | A read-only implementation that is always active(1). It is used for creating, modifying, and deleting. |
| cpwAtmOamCellSupported | Indicates whether operation, administration, and maintenance (OAM) cells are transported on this VC. |
| cpwAtmQosScalingFactor | Represents the scaling factor to be applied to ATM QoS rates when calculating QoS rates for the PSN domain. |
| cpwAtmCellPacking | Identifies if the VC is configured to do cell packing. |
| cpwAtmMncp | Identifies the number of cells that need to be packed. |
| cpwAtmEncap | Provides information on whether MPLS or Layer 2 Tunneling Protocol Version 3 (L2TPv3) is used as the transport. |
| cpwAtmPeerMncp | Represents the maximum number of cells that can be packed in one packet for a peer interface. |
| cpwAtmMcptTimeout | Represents the maximum cell packing timeout (MCPT) value used. |

## cpwVcAtmPerfTable

Table 13 lists the cpwVcAtmPerfTable objects and their descriptions.

*Table 13 cpwVcAtmPerfTable Objects and Descriptions*

| Objects | Description |
|---|---|
| cpwAtmCellsReceived | Obtains information on the number of cells that were received and sent to the PSN. |
| cpwAtmCellsSent | Provides information on the number of cells sent to the ATM network. |
| cpwAtmCellsRejected | Indicates the number of cells that were rejected by this VC because of policing. |
| cpwAtmCellsTagged | Indicates the number of cells that were tagged. |
| cpwAtmHCCellsReceived | Provides the high-capacity counter for the number of cells received by this VC. |
| cpwAtmHCCellsRejected | Provides the high-capacity counter for the number of cells rejected by this VC. |
| cpwAtmHCCellsTagged | Provides the high-capacity counter for number of cells that were tagged. |
| cpwAtmAvgCellsPacked | Provides the average number of cells that were packed. |
| cpwAtmPktsReceived | Indicates the number of ATM AAL5 packets that are actually sent into the ATM network as packets when the VC is configured to do AAL5 over PW. |
| cpwAtmPktsSent | Gets the number of packets that are reconstructed from the cells, assigns a VC label, and sends the packets into the PSN. |
| cpwAtmPktsRejected | Indicates the number of packets that were rejected because of policing. |

# Objects in the PWE3 MIBs

The PWE3 MIBs represent an ASN.1 notation reflecting specific components of the pseudowire services. The MIBs enable a network management application using SNMP to get this information for display. The MIBs support the standard GETNEXT and GETBULK functionality, but do not support configuration capabilities (via SET) in the current implementation.

# Scalar Objects in the PWE3 MIBs

The PWE3 MIBs contain the following supported scalar object:

- cpwVcUpDownNotifEnable—This object reflects the configuration of the cpwVcUp and cpwVcDown notifications. If either of the notifications is configured via the command-line interface (CLI), then this object has a value of true(1). If this object is set via SNMP to true(1), then it enables the emission of both the cpwVcUp and cpwVcDown notifications; if the object is set via SNMP to false(2), these notifications are not emitted.

**Note** cpwVcUpDownNotifEnable can be set only if RW is configured for the **snmp-server community** *string* [**view** *view-name*] [**ro** | **rw**] [**ipv6** *nacl*] [*access-list-number*] command.

The PWE3 MIBs contain the following unsupported scalar objects:

- cpwVcIndexNext—Indicates the next cpwVcIndex value to use when you add rows to the cpwVcTable.
- cpwVcNotifRate—Indicates the rate at which cpwVcUp/Down notifications can be issued from the device.
- cpwVcMplsOutboundIndexNext—Contains an appropriate value to be used for cpwVcMplsOutboundIndex when you create entries in the cpwVcMplsOutboundTable. The value 0 indicates that no unassigned entries are available. To obtain the cpwVcMplsOutboundIndex value for a new entry, the manager issues a management protocol retrieval operation to obtain the current value of this object. After each retrieval, the software agent should modify the value to the next unassigned index; however, the software agent *must not* assume such retrieval will be done for each row created.
- cpwVcMplsInboundIndexNext—Contains an appropriate value to be used for cpwVcMplsInboundIndex when you create entries in the cpwVcMplsInboundTable. The value 0 indicates that no unassigned entries are available. To obtain the cpwVcMplsInboundIndex value for a new entry, the manager issues a management protocol retrieval operation to obtain the current value of this object. After each retrieval, the software agent should modify the value to the next unassigned index; however, the agent *must not* assume such retrieval will be done for each row created.

## Notifications in the PWE3 MIBs

The cpwVcUp and cpwVcDown notifications in the PW-MIB indicate when the operStatus values for a range of PW VCs have changed state.

The definition of these objects in the PW-MIB indicates that events of the same type, either up or down, must be able to be correlated into ranges. The implementation of these notifications does not do any of this correlation. A notification is generated for each individual VC that has an operational state change if that notification is enabled. A notification does not signal an operational state change for a group of VCs as described in the MIB.

## Benefits of the PWE3 MIBs

The PWE3 MIBs provide the ability to manage pseudowire emulation edge-to-edge by providing MPLS-related information about the service and a mechanism to monitor the Ethernet, Frame Relay, or ATM access circuits.

# How to Configure Pseudowire Emulation Edge-to-Edge MIBs for Ethernet, Frame Relay, and ATM Services

This section contains the following procedures:

- Enabling the SNMP Agent for the PWE3 MIBs, page 21 (required)
- Configuring AToM, Frame Relay, or ATM Circuits Across a Network for the PWE3 MIBs, page 22 (required)

# Enabling the SNMP Agent for the PWE3 MIBs

Perform this task to enable the SNMP agent.

## SUMMARY STEPS

1. **enable**
2. **show running-config** [**interface | map-class**]
3. **configure terminal**
4. **snmp-server community** *string* [**view** *view-name*] [**ro | rw**] [**ipv6** *nacl*] [*access-list-number*]
5. **end**
6. **write memory**

## DETAILED STEPS

| | Command or Action | Purpose |
|---|---|---|
| **Step 1** | **enable**<br><br>**Example:**<br>`Router> enable` | Enables privileged EXEC mode.<br>• Enter your password if prompted. |
| **Step 2** | **show running-config** [**interface** \| **map-class**]<br><br>**Example:**<br>`Router# show running-config` | Displays the running configuration of the router so that you can determine if an SNMP agent is already running on the device.<br>If no SNMP information is displayed, continue with the next step.<br>If any SNMP information is displayed, you can modify the information or change it as desired.<br>• The optional **interface** keyword displays interface-specific configuration information.<br>• The optional **map-class** keyword displays dialer or Frame Relay map-class information. |
| **Step 3** | **configure terminal**<br><br>**Example:**<br>`Router# configure terminal` | Enters global configuration mode. |

| | Command or Action | Purpose |
|---|---|---|
| **Step 4** | **snmp-server community** *string* [**view** *view-name*] [**ro** \| **rw**] [**ipv6** *nacl*] [*access-list-number*]<br><br>**Example:**<br>`Router(config)# snmp-server community public ro` | Sets up the community access string to permit access to SNMP for the MIBs.<br>• The *string* argument consists of 1 to 32 alphanumeric characters and functions much like a password, permitting access to SNMP. Blank spaces are not permitted in the community string.<br>• The optional **view** *view-name* keyword argument combination specifies a previously defined view. The view defines the objects available to the SNMP community.<br>• The optional **ro** keyword configures read-only (ro) access to the objects in the MIBs.<br>• The optional **rw** keyword specifies read-write access. Authorized management stations can both retrieve and modify MIB objects.<br>• The optional **ipv6** *nacl* keyword argument combination specifies an IPv6 named access list.<br>• The optional *access-list-numbe*r argument is an integer from 1 to 99 that specifies a standard access list of IP addresses or a string (not to exceed 64 characters) that is the name of a standard access list of IP addresses allowed access to the SNMP agent. Alternatively, it is an integer from 1300 to 1999 that specifies a list of IP addresses in the expanded range of standard access list numbers that are allowed to use the community string to gain access to the SNMP agent. |
| **Step 5** | **end**<br><br>**Example:**<br>`Router(config)# end` | Exits to privileged EXEC mode. |
| **Step 6** | **write memory**<br><br>**Example:**<br>`Router# write memory` | Writes the modified SNMP configuration into NVRAM of the router, permanently saving the SNMP settings. |

# Configuring AToM, Frame Relay, or ATM Circuits Across a Network for the PWE3 MIBs

This section contains the following procedure:

• Configuring the Pseudowire Class, page 23

# Configuring the Pseudowire Class

The successful transmission of the Layer 2 frames between PE routers is due to the configuration of the PE routers. You configure the connection, called a pseudowire, between the routers.

**Note** In simple configurations, this task is optional. You do not need to specify a pseudowire class if you specify the tunneling method as part of the **xconnect** command.

The pseudowire-class configuration group specifies the following characteristics of the tunneling mechanism:

- Encapsulation type
- Control protocol
- Payload-specific options

For more information, see the **pseudowire-class** command in the following feature module:

*Layer 2 Tunnel Protocol Version 3*

You must specify the **encapsulation mpls** command as part of the pseudowire class or as part of the **xconnect** command for the AToM VCs to work properly. If you omit the **encapsulation mpls** command as part of the **xconnect** command, you receive the following error:

```
% Incomplete command.
```

Once you specify the **encapsulation mpls** command, you cannot remove it using the **no encapsulation mpls** command. Nor can you change the command's setting using the **encapsulation l2tpv3** command. Those methods result in the following error message:

```
Encapsulation changes are not allowed on an existing pw-class.
```

To remove the command, you must delete the pseudowire with the **no pseudowire-class** command. To change the type of encapsulation, remove the pseudowire with the **no pseudowire-class** command and reestablish the pseudowire and specify the new encapsulation type.

**Note** There are many options that you can configure. For detailed information, see the *Any Transport over MPLS* feature module or the *Cisco IOS Wide-Area Networking Configuration Guide*, Release 12.4.

## SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **pseudowire-class** *name*
4. **encapsulation mpls**

**DETAILED STEPS**

| | Command or Action | Purpose |
|---|---|---|
| **Step 1** | **enable**<br><br>**Example:**<br>`Router> enable` | Enables privileged EXEC mode.<br>• Enter your password if prompted. |
| **Step 2** | **configure terminal**<br><br>**Example:**<br>`Router# configure terminal` | Enters global configuration mode. |
| **Step 3** | **pseudowire-class** *name*<br><br>**Example:**<br>`Router(config)# pseudowire-class atom` | Establishes a pseudowire class with a name that you specify and enters pseudowire class configuration mode. |
| **Step 4** | **encapsulation mpls**<br><br>**Example:**<br>`Router(config-pw)# encapsulation mpls` | Specifies the tunneling encapsulation. For AToM, the encapsulation type is **mpls**. |

## What to Do Next

Perform a MIB walk using your SNMP management tool on cpwVcMIB, cpwVcMplsMIB, cpwVcEnetMIB, cpwVcFrMIB, and cpwVcAtmMIB to verify that the PW-MIB, the PW-MPLS-MIB, the PW-ENET-MIB, the PW-FR-MIB, and the PW-ATM-MIB objects, respectively, are populated correctly.

**Note** SNMPv1 and SNMPv2c are supported.

# Configuration Examples for the Pseudowire Emulation Edge-to-Edge MIBs for Ethernet, Frame Relay, and ATM Services

This section provides the following configuration example:

- PWE3 MIBs: Example, page 24

## PWE3 MIBs: Example

In the following example, the configuration permits any SNMP manager to access all objects with read-only permissions using the community string public.

```
Router# configure terminal
```

```
Enter configuration commands, one per line.  End with CNTL/Z.
Router(config)# snmp-server community public ro
```

**Note** There is no explicit way to configure the PWE3 MIBs. However, for information on AToM configuration tasks and examples, see the *Any Transport over MPLS* feature module; for Frame Relay and ATM, see the *Cisco IOS Wide-Area Networking Configuration Guide*, Release 12.4.

There are notifications specific to the PWE3 MIBs. For detailed information on the commands used to configure them, see the "Additional References" section on page 25.

# Additional References

The following sections provide references related to the Pseudowire Emulation Edge-to-Edge MIBs for Ethernet, Frame Relay, and ATM Services feature.

## Related Documents

| Related Topic | Document Title |
|---|---|
| SNMP commands | *Cisco IOS Network Management Command Reference*, Release 12.2SX<br>*Cisco IOS Network Management Command Reference*, Release 12.2SR |
| SNMP configuration tasks | Configuring SNMP Support" chapter in the *Cisco Network Management Configuration Guide*, Release 12.4 |
| SNMP support for VPNs | *SNMP Notification Support for VPNs* |
| Ethernet over MPLS configuration tasks | The "How to Configure Any Transport over MPLS" section in the *Any Transport over MPLS* feature module |
| Frame Relay configuration tasks | *Cisco IOS Wide-Area Networking Configuration Guide*, Release 12.4 |

| Related Topic | Document Title |
|---|---|
| ATM configuration tasks | *Cisco IOS Asynchronous Transfer Mode Configuration Guide*, Release 12.4 |
| Pseudowire-related Internet drafts | • *An Architecture for Multi-Segment Pseudo Wire Emulation Edge-to-Edge*, Internet draft, September 25, 2008 [draft-ietf-pwe3-ms-arch-05.txt]<br>• *Definitions for Textual Conventions and OBJECT-IDENTITIES for Pseudo-Wires Management*, Internet draft, August 10, 2007 [draft-ietf-pwe3-pw-tc-mib-09.txt]<br>• *Ethernet Pseudo Wire (PW) Management Information Base*, Internet draft, August 30, 2007 [draft-pwe3-enet-mib-10.txt]<br>• *Managed Objects for ATM over Packet Switched Network (PSN)*, Internet draft, August 8, 2007 [draft-ietf-pwe3-pw-atm-mib-02.txt]<br>• *Pseudo Wire (PW) Management Information Base*, Internet draft, January 8, 2008 [draft-ietf-pwe3-pw-mib-14.txt]<br>• *Pseudo Wire (PW) over MPLS PSN Management Information Base*, Internet draft, August 11, 2007 [draft-ietf-pwe3-pw-mpls-mib-11.txt]<br>**Note** For information on using SNMP MIB features, see the appropriate documentation for your network management system. |

# Standards

| Standard | Title |
|---|---|
| No new or modified standards are supported by this feature, and support for existing standards has not been modified by this feature. | — |

# MIBs

| MIB | MIBs Link |
|---|---|
| SNMP-VACM-MIB | To locate and download MIBs for selected platforms, Cisco IOS releases, and feature sets, use Cisco MIB Locator found at the following URL:<br>http://www.cisco.com/go/mibs |

## RFCs

| RFC | Title |
|---|---|
| RFC 1156 | *Management Information Base for Network Management of TCP/IP-based Internets* |
| RFC 1157 | *A Simple Network Management Protocol (SNMP)* |
| RFC 1213 | *Management Information Base for Network Management of TCP/IP-based Internets: MIB-II* |
| RFC 1315 | *Management Information Base for Frame Relay DTEs* |
| RFC 3815 | *Definitions of Managed Objects for the Multiprotocol Label Switching (MPLS), Label Distribution Protocol (LDP)* |
| RFC 3916 | *Requirements for Pseudo-Wire Emulation Edge-to-Edge (PWE3)* |
| RFC 4619 | *Encapsulation Methods for Transport of Frame Relay over Multiprotocol Label Switching (MPLS) Networks* |

## 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.<br><br>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.<br><br>Access to most tools on the Cisco Support website requires a Cisco.com user ID and password. | http://www.cisco.com/techsupport |

# Command Reference

This feature uses no new or modified commands.

# Feature Information for Pseudowire Emulation Edge-to-Edge MIBs for Ethernet, Frame Relay, and ATM Services

Table 14 lists the release history for this feature.

Not all commands may be available in your Cisco IOS software release. For release information about a specific command, see the command reference documentation.

Use Cisco Feature Navigator to find information about platform support and software image support. Cisco Feature Navigator enables you to determine which Cisco IOS and Catalyst OS software images support a specific software release, feature set, or platform. To access Cisco Feature Navigator, go to http://www.cisco.com/go/cfn. An account on Cisco.com is not required.

**Note** Table 14 lists only the Cisco IOS software release that introduced support for a given feature in a given Cisco IOS software release train. Unless noted otherwise, subsequent releases of that Cisco IOS software release train also support that feature.

*Table 14 Feature Information for Pseudowire Emulation Edge-to-Edge MIBs for Ethernet, Frame Relay, and ATM Services*

| Feature Name | Releases | Feature Information |
|---|---|---|
| Pseudowire Emulation Edge-to-Edge MIBs for Ethernet, Frame Relay, and ATM Services | 12.0(29)S<br>12.0(30)S<br>12.0(31)S<br>12.2(28)SB<br>12.2(33)SRA<br>12.4(11)T<br>12.2(33)SXH | The Pseudowire Emulation Edge-to-Edge MIBs for Ethernet, Frame Relay, and ATM Services feature provides Simple Network Management Protocol (SNMP) support within an Any Transport over Multiprotocol Label Switching (AToM) infrastructure emulating Ethernet, Frame Relay, and ATM services over packet switched networks (PSNs).<br><br>In Cisco IOS Release 12.0(29)S, this feature was introduced as Pseudowire Emulation Edge-to-Edge MIBs for Ethernet Services.<br><br>In Cisco IOS Release 12.0(30)S, the title changed to Pseudowire Emulation Edge-to-Edge MIBs for Ethernet and Frame Relay Services because Frame Relay was added as a transport. Support was added for the Cisco 12000 series routers and for the CISCO-IETF-PW-FR-MIB (PW-FR-MIB).<br><br>In Cisco IOS Release 12.0(31)S, the CISCO-IETF-PW-MPLS-MIB (PW-MPLS-MIB) was modified regarding how cross-connect (XC) and tunnel indexes appear for virtual circuits (VCs).<br><br>In Cisco IOS Release 12.2(28)SB, the title changed to Pseudowire Emulation Edge-to-Edge MIBs for Ethernet, Frame Relay, and ATM Services because ATM was added as a transport. Support was added for the CISCO-IETF-PW-ATM-MIB (PW-ATM-MIB).<br><br>In Cisco IOS Releases 12.2(33)SRA, 12.4(11)T, and 12.2(33)SXH, this feature was integrated into the releases as the Pseudowire Emulation Edge-to-Edge MIBs for Ethernet and Frame Relay Services feature because ATM is not supported as a transport. |

# Glossary

**AAL**—ATM adaptation layer. AAL defines the conversion of user information into cells. AAL1 and AAL2 handle isochronous traffic, such as voice and video; AAL3/4 and AAL5 pertain to data communications through the segmentation and reassembly of packets.

**ATM**—Asynchronous Transfer Mode. A cell-based data transfer technique in which channel demand determines packet allocation. This is an international standard for cell relay in which multiple service types (such as voice, video, or data) are conveyed in fixed-length (53-byte) cells. Fixed-length cells allow cell processing to occur in hardware, thereby reducing transit delays. ATM is designed to take advantage of high-speed transmission media such as E3, SONET, and T3.

**CE router**—customer edge router. A router that is part of a customer network and that interfaces to a provider edge (PE) router.

**DLCI**—data-link connection identifier. A unique number assigned to a PVC endpoint in a Frame Relay network. Identifies a particular PVC endpoint within an access channel in a Frame Relay network and has local significance only to that channel.

**encapsulation**—Wrapping of data in a particular protocol header. For example, Ethernet data is wrapped in a specific Ethernet header before network transit. Also, when bridging occurs in dissimilar networks, the entire frame from one network is simply placed in the header used by the data link layer protocol of the other network.

**EoMPLS**—Ethernet over Multiprotocol Label Switching (MPLS). A tunneling mechanism that allows a service provider to tunnel customer Layer 2 traffic through a Layer 3 MPLS network. EoMPLS is a point-to-point solution only. EoMPLS is also known as Layer 2 tunneling.

**Frame Relay**—The industry standard, switched data link layer protocol that handles multiple virtual circuits using High-Level Data Link Control (HDLC) encapsulation between connected devices. Frame Relay is more efficient than X.25, the protocol for which it is generally considered a replacement.

**IETF**—Internet Engineering Task Force. A task force (consisting of more than 80 working groups) that is developing standards for the Internet and the IP suite of protocols.

**LDP**—Label Distribution Protocol. The protocol that supports MPLS hop-by-hop forwarding and the distribution of bindings between labels and network prefixes. The Cisco proprietary version of this protocol is the Tag Distribution Protocol (TDP).

**LSP**—label-switched path. A configured connection between two label switch routers (LSRs) in which label-switching techniques are used for packet forwarding; also a specific path through an MPLS network.

**LSR**—label switch router. A Multiprotocol Label Switching (MPLS) node that can forward native Layer 3 packets. The LSR forwards a packet based on the value of a label attached to the packet.

**MIB**—Management Information Base. A database of network management information that is used and maintained by a network management protocol such as Simple Network Management Protocol (SNMP). The value of a MIB object can be changed or retrieved by using SNMP commands, usually through a network management system. MIB objects are organized in a tree structure that includes public (standard) and private (proprietary) branches.

**MPLS**—Multiprotocol Label Switching. A switching method for the forwarding of IP traffic through the use of a label. This label instructs the routers and the switches in the network where to forward the packets based on preestablished IP routing information.

**MTU**—maximum transmission unit. Maximum packet size, in bytes, that a particular interface can handle.

**NMS**—network management system. System responsible for managing at least part of a network. An NMS is generally a reasonably powerful and well-equipped computer, such as an engineering workstation. An NMS communicates with agents to help keep track of network statistics and resources.

**notification**—A message sent by a Simple Network Management Protocol (SNMP) agent to a network management station, console, or terminal to indicate that a significant network event has occurred. *See also* trap.

**OSPF**—Open Shortest Path First. A link-state hierarchical Interior Gateway Protocol routing algorithm, derived from the IS-IS protocol. OSPF features include least-cost routing, multipath routing, and load balancing.

**PE router**—provider edge router. A router that is part of a service provider's network and is connected to a customer edge (CE) router.

**primary tunnel**—A tunnel whose label-switched path (LSP) may be fast rerouted if there is a failure. Backup tunnels cannot be primary tunnels.

**pseudowire**—PW. A mechanism that carries the elements of an emulated service from one provider edge (PE) to one or more PEs over a packet switched network (PSN).

**SNMP**—Simple Network Management Protocol. A management protocol used almost exclusively in TCP/IP networks. SNMP provides a means for monitoring and controlling network devices, and for managing configurations, statistics collection, performance, and security.

**trap**—A message sent by an SNMP agent to a network management station, console, or terminal, indicating that a significant event occurred. Traps are less reliable than notification requests because the receiver does not send an acknowledgment when it receives a trap. The sender cannot determine if the trap was received.

**tunnel**—A secure communication path between two peers, such as routers.

**VC**—virtual circuit. A logical circuit created to ensure reliable communication between two network devices. A virtual circuit can be either permanent (PVC) or switched (SVC).

---