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

The Pseudowire Emulation Edge-to-Edge (PWE3) MIBs for Ethernet and Frame Relay Services provide Simple Network Management Protocol (SNMP) support within an Any Transport over Multiprotocol Label Switching (AToM) infrastructure emulating Ethernet and Frame Relay services over packet switched networks (PSNs). The PWE3 MIBs include the CISCO-IETF-PW-MIB (PW-MIB), the CISCO-IETF-PW-MPLS-MIB (PW-MPLS-MIB), the CISCO-IETF-PW-ENET-MIB (PW-ENET-MIB), and the CISCO-IETF-PW-FR-MIB (PW-FR-MIB).

This release introduces support for the CISCO-IETF-PW-FR-MIB (PW-FR-MIB), which provides network management information specific to a Frame Relay over pseudowire (FRoPW) connection in an MPLS AToM or an IP network.

Feature History for the PWE3 MIBs

Release	Modification
12.0(29)S	This feature was introduced.
12.0(30)S	The title was modified to include Frame Relay as a transport. Support was added for the Cisco 12000 Series Routers and for the CISCO-IETF-PW-FR-MIB (PW-FR-MIB).

Finding Support Information for Platforms and Cisco IOS Software Images

Use Cisco Feature Navigator to find information about platform support and Cisco IOS software image support. Access Cisco Feature Navigator at http://www.cisco.com/go/fn. You must have an account on Cisco.com. If you do not have an account or have forgotten your username or password, click **Cancel** at the login dialog box and follow the instructions that appear.

Contents

- Prerequisites for the PWE3 MIBs, page 2
- Restrictions for the PWE3 MIBs, page 2
- Information About the PWE3 MIBs, page 2

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

- How to Configure the PWE3 MIBs, page 18
- Configuration Examples for the PWE3 MIBs, page 20
- Additional References, page 21
- Command Reference, page 23
- Glossary, page 24

Prerequisites for the PWE3 MIBs

- SNMP must be enabled on the label switch routers (LSRs).
- MPLS must be enabled on the LSRs.
- Pseudowires must be configured with Ethernet or Frame Relay access circuits. (For more detailed information, see the *Any Transport over MPLS* feature module or the *Cisco IOS Wide-Area Networking Configuration Guide*, Release 12.3.)

Restrictions for the PWE3 MIBs

This implementation of the PWE3 MIBs is limited to read-only (RO) permission for MIB objects except for the cpwVcUp and cpwVcDown notification enable object, cpwVcUpDownNotifEnable, which, for purposes of this release, has been extended to be writable by the SNMP agent.

- The following tables in the PW-MIB are not supported in this release:
  - cpwVcPerfCurrentTable
  - cpwVcPerIntervalTable
- The following objects in the PW-MPLS-MIB are not supported in this release:
  - cpwVcMplsOutboundIndexNext
  - cpwVcMplsInboundIndexNext
- The following tables in the PW-ENET-MIB are not supported in this release:
  - cpwVcEnetMplsPriMappingTable
  - cpwVcEnetStatsTable
- The following table in the PW-FR-MIB is not supported in this release:
  - cpwVcFrPMTable

Information About the PWE3 MIBs

To configure the PWE3 MIBs, you need to understand the following concepts:

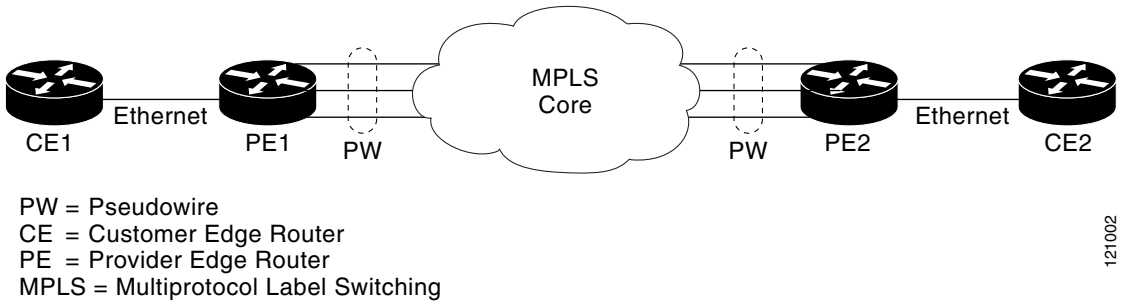
- What is a Pseudowire, page 3
- PWE3 MIBs Architecture, page 3
- Components and Functions, page 4
- MIB Tables in the PW-MIB, page 5
- MIB Tables in the PW-MPLS-MIB, page 11

- MIB Tables in the PW-ENET-MIB, page 14
- MIB Tables in the PW-FR-MIB, page 16
- Objects, page 16
- Scalar Objects, page 17
- Notifications, page 17
- Benefits, page 17

What is a Pseudowire

A pseudowire is a point-to-point connection between pairs of provider edge (PE) routers (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.

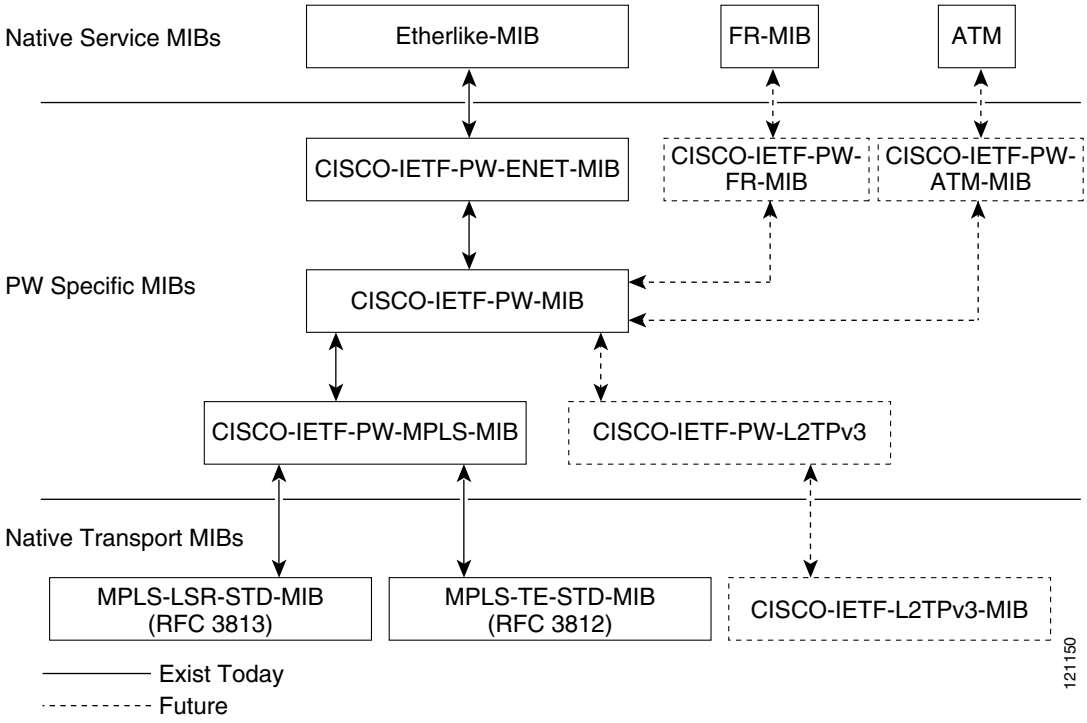
Figure 1 Sample Pseudowire Topology



PWE3 MIBs Architecture

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

Figure 2 PWE3 MIBs Architecture



The architecture is modular in nature, in that once deployed, new emulated service MIB modules or additional transport MIB modules simply "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 new management strategy. Because the architecture is a generalized association mechanism between existing service and transport MIB modules, it should be noted that the 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. For example, although the Asynchronous Transfer Mode (CISCO-IETF-PW-ATM) MIB is not available in this implementation, you may still see entries in the corresponding native service MIB modules such as the ATM Forum and Cisco proprietary ATM MIB modules.

When the corresponding PWE3-specific MIB modules are implemented in the future, you will be able to see those pseudowire associations at that time.

Components and Functions

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 by using counters for monitoring and configuration of services, statistics, and notifications by using the SNMP protocol.

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

This MIB provides 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, by using the SNMP protocol.

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

  This MIB provides managed objects that can be used by a network manager to monitor pseudowire emulation Ethernet services by using the SNMP protocol.

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

  This MIB provides managed objects that can be used by a network manager to monitor pseudowire emulation Frame Relay services by using the SNMP protocol.

  This MIB uses an 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 (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 virtual circuit (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. Currently, each PW VC index uniquely maps into an existing VC entry in the PW-MIB and the PW-FR-MIB.

MIB Tables in the PW-MIB

The PW-MIB consists of the following tables:

- cpwVcTable (Table 1)—Contains high-level generic parameters related to virtual circuit (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 include: • 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. • 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 Layer 2 Tunneling Protocol (L2TP) control protocol. • 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 include: • strict(1) • loose(2) 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 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. 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/PE maintenance protocol is used for the VC creation. It should be set to unknown if the PE/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/PE 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 pair (AVP) for the L2TP.
cpwVcLocalGroupID	Use in the Group ID field sent to the peer PW/PE 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 on the ifTable in the name object as part of the maintenance protocol. If 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. 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 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 optionally received from the remote node via the maintenance protocol. Should be 0 if this parameter is not available or not used.
cpwVcRemoteIfString	Indicates the interface description string as received by the maintenance protocol; must be 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 owner is manual or automatically otherwise. Examples: for MPLS PSN, the label represents the 20 bits of VC tag, for L2TP it represents the 32 bits of 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 owner is manual or automatically otherwise. Examples: for MPLS PSN, the label represents the 20 bits of VC tag; for L2TP the label represents the 32 bits of 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 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 severance from most severe to less severe is as follows: unknown, notPresent, down, lowerLayerDown, dormant, testing, and up. The operator can consult the per direction OperStatus for fault isolation per direction.
cpwVcInboundOperStatus	Indicates the actual operational status of this VC in the inbound direction. Values include: • up—The VC is established and ready to pass packets. • down—PW signaling has not yet finished or indications available at the service level show that the VC is not passing packets. • testing—AdminStatus at the VC level is set to test. • dormant—The VC is not available because the required resources are occupied by higher priority VCs. • notPresent—Some component is missing to accomplish the setup of the VC. • 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 include: • up—The VC is established and ready to pass packets. • down—PW signaling has not yet finished or indications available at the service level show that the VC is not passing packets. • testing—AdminStatus at the VC level is set to test. • dormant—The VC is not available because the required resources are occupied by higher priority VCs. • notPresent—Some component is missing to accomplish the setup of the VC. • 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 *n* intervals. The minimum value of *n* is 4; the default of *n* is 32 and the maximum value of *n* is 96. The value is *n* unless the measurement was (re)started within the last *n**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. For the current implementation, this 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 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.

MIB 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 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 in this implementation; consequently, its value is 1. The operator creates at least one entry in this table for each PW VC that requires an MPLS PSN. This implementation does not support the VC-only case or the cpwVcMplsOutboundIndex.
- 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 in this implementation; 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. This implementation does not support the VC-only case or the cpwVcMplsInboundIndex.
- 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 quickly the PW carried over a specific non-TE MPLS outer tunnel. This table in indexed by the XC 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 quickly the PW carried over a specific TE MPLS outer tunnel. This table in 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. For this implementation, values include: • mplsTe(0)—Used if the outer tunnel were set up by MPLS-TE. • mplsNonTe(1)—Used if the outer tunnel were 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. For this implementation, values include: • 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 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. As 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 if 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. Next available free index can be retrieved using cpwVcMplsOutboundIndexNext. In this implementation, the value = 1.
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 XC index of the outer tunnel. Otherwise, it 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, set to 0.
cpwVcMplsOutboundTunnelInstance	Part of the set of indexes for an outbound tunnel, specifically an MPLS-TE outer tunnel; otherwise, set to 0.
cpwVcMplsOutboundTunnelLclLSR	Part of the set of indexes for an outbound tunnel, specifically an MPLS-TE outer tunnel; otherwise, set to NULL.
cpwVcMplsOutboundTunnelPeerLSR	Part of the set of indexes for an outbound tunnel, specifically an MPLS-TE outer tunnel; otherwise, set to NULL.
cpwVcMplsOutboundIfIndex	For a VC only with no outer tunnel, this object holds the ifIndex of the outbound port. For this implementation, 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. Next available free index can be retrieved using cpwVcMplsInboundIndexNext. In this implementation, the value = 1.
cpwVcMplsInboundLsrXcIndex	If the outer label is defined in the MPL-LSR-MIB; that is, set by LDP or manually, this object points to the XC index of the outer tunnel. The XC index represents the pseudowire in the inbound direction retrieving 0 if information 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.
cpwVcMplsInboundTunnelPeerLSR	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
cpwVcMplsInboundIfIndex	In case of a VC only (no outer tunnel), this object holds the ifIndex of the inbound port. In this implementation, 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; for this implementation, 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.

MIB 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 indices 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 in the current implementation are as follows: • noChange—Indicates that the VC contains the original user VLAN, as specified in cpwVcEnetPortVlan. • changeVlan—Indicates that the VLAN field on the VC may be different from the VLAN field on the user's port. • removeVlan—Indicates that the encapsulation on the VC does not include the original VLAN field.
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.
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. In this implementation, this value is always 0 to indicate no virtual interface is created.

Table 10 cpwVcEnetTable Objects and Descriptions (continued)

Objects	Description
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).

MIB 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 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).

Objects

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

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 **snmp-server community** *string* [**view** *view-name*] [**ro**] [*number*].

The PWE3 MIBs contain the following unsupported scalar objects:

- cpwVcIndexNext—Indicates the next cpwVcIndex value when adding 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 creating 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 creating 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

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

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 or Frame Relay access circuits.

How to Configure the PWE3 MIBs

This section contains the following procedures:

- Enabling the SNMP Agent, page 18 (required)
- Setting Up AToM or Frame Relay Circuits Across a Network, page 19 (required)
- Verifying the PWE3 MIBs Configuration, page 19 (optional)

Enabling the SNMP Agent

Perform this task to enable the SNMP agent.

SUMMARY STEPS

1. **enable**
2. **show running-configuration**
3. **configure terminal**
4. **snmp-server community** *string* [**view** *view-name*] [**ro**] [*number*]
5. **end**
6. **write memory**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**show running-configuration** **Example:** `Router# show running-configuration`	Displays the running configuration of the router so that you can determine if an SNMP agent is already running on the device. If no SNMP information is displayed, continue with the next step. If any SNMP information is displayed, you can modify the information or change it as desired.
Step 3	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.

	Command or Action	Purpose
Step 4	**snmp-server community** *string* [**view** *view-name*] [**ro**] [*number*] **Example:** `Router(config)# snmp-server community public ro`	Configures read-only (ro) community strings for the MIBs. • The *string* argument functions like a password, permitting access to SNMP functionality on LSRs in an MPLS network. • The optional **ro** keyword configures read-only (ro) access to the objects in the MIBs.
Step 5	**end** **Example:** `Router(config)# end`	Exits to privileged EXEC mode.
Step 6	**write memory** **Example:** `Router# write memory`	Writes the modified SNMP configuration into NVRAM of the router, permanently saving the SNMP settings.

Setting Up AToM or Frame Relay Circuits Across a Network

For detailed information, see the *Any Transport over MPLS* feature module or the *Cisco IOS Wide-Area Networking Configuration Guide*, Release 12.3.

Verifying the PWE3 MIBs Configuration

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

Note This release supports SNMPv1 and SNMPv2c.

Configuration Examples for the PWE3 MIBs

This section provides the following configuration example:

- PWE3 MIBs: Example, page 20

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, see the *Cisco IOS Wide-Area Networking Configuration Guide*, Release 12.3.

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 21.

Additional References

The following sections provide references related to the PWE3 MIBs.

Related Documents

Related Topic	Document Title
SNMP commands	*Cisco IOS Configuration Fundamentals and Network Management Command Reference,* Release 12.3
SNMP configuration	"Configuring SNMP Support" chapter in the *Cisco IOS Configuration Fundamentals and Network Management Configuration Guide*
SNMP Support for VPNs	*SNMP Notification Support for VPNs*
EoMPLS 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 12.3
Other documentation	*Pseudo Wire (PW) Management Information Base*, Internet draft, February 2004 [draft-ietf-pwe3-pw-mib-04.txt]; Zelig, D., Nadeau, T.D., Danenberg, D. and Mantin, S. *Ethernet Pseudo Wire (PW) Management Information Base*, Internet draft, February 2004 [draft-pwe3-enet-mib-04.txt]; Zelig, D. and Nadeau, T.D. *Pseudo Wire (PW) over MPLS PSN Management Information Base*, Internet draft, February 2004 [draft-ietf-pwe3-pw-mpls-mib-05.txt]; Zelig, D., Nadeau, T.D., Danenberg, D., Mantin, S. and Malis, A. *Definitions for Textual Conventions and OBJECT-IDENTITIES for Pseudo-Wires Management*, Internet draft, February 2003 [draft-ietf-pwe3-pw-tc-mib-04.txt]; Nadeau, T.D., Danenberg, D., Zelig, D. and Malis, A. *Frame Relay over Pseudo-Wires*, Internet draft, July 2003 [draft-ietf-pwe3-frame-relay-01.txt]; Martini, L., et al. **Note** For information on using SNMP MIB features, see the appropriate documentation for your network management system.

Standards

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

MIBs

MIBs	MIBs Link
• *MPLS Label Distribution Protocol MIB* (draft-ietf-mpls-ldp-mib-08.txt) • SNMP-VACM-MIB *The View-based Access Control Model (ACM) MIB for SNMP*	To locate and download MIBs for selected platforms, Cisco IOS releases, and feature sets, use Cisco MIB Locator found at the following URL: http://www.cisco.com/go/mibs

RFCs

RFCs	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*

Technical Assistance

Description	Link
Technical Assistance Center (TAC) home page, containing 30,000 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/public/support/tac/home.shtml

Command Reference

This feature uses no new or modified commands. All commands used with this feature are documented in the Cisco IOS Release 12.3 command reference publications.

Glossary

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 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 though a Layer 3 MPLS network. EoMPLS is a point-to-point solution only. EoMPLS is also known as Layer 2 tunneling.

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. NMSs communicate 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.

Note Refer to *Internetworking Terms and Acronyms* for terms not included in this glossary.