Pseudowire Emulation Edge-to-Edge MIBs for Ethernet Services

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

Feature History for the PWE3 MIBs

Release	Modification
12.0(29)S	This feature was introduced.

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

CISCO SYSTEMS

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

Prerequisites for the PWE3 MIBs

- SNMP must be installed and enabled on the label switch routers (LSRs).

- MPLS must be enabled on the LSRs.

- Pseudowires must be configured with Ethernet access circuits. (For more detailed information, see the *Any Transport over MPLS* feature module.)

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

Information About the PWE3 MIBs

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

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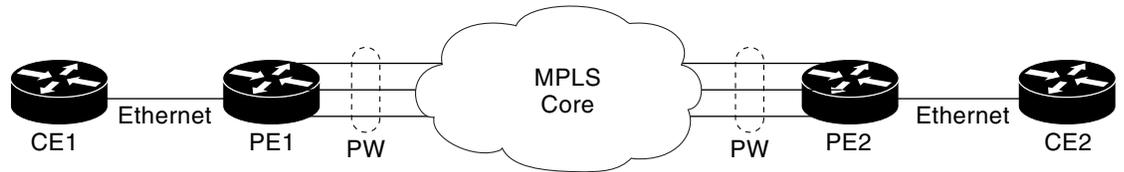
PW = Pseudowire
CE = Customer Edge Router
PE = Provider Edge Router
MPLS = Multiprotocol Label Switching

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
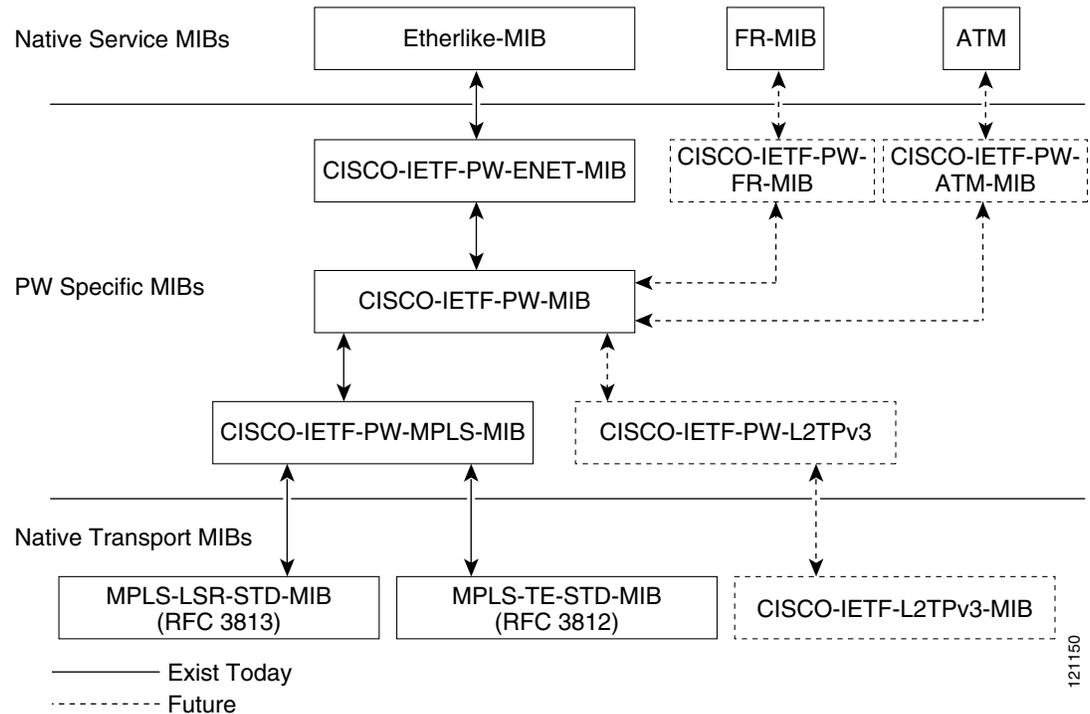
The architecture is modular in nature, in that once deployed, new emulated service MIB modules or additional transport MIB modules simply "plug in" to or extend the existing infrastructure rather than require a new and unique one. This allows you to build management applications without the concern of a new service requiring the deployment of a completely new management strategy. Since the architecture is in essence 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 association with the pseudowire are absent. For example, although the PWE3 Frame Relay (CISCO-IETF-PW-FR-MIB) and the Asynchronous Transfer Mode (CISCO-IETF-PW-ATM) MIBs are not available in this implementation, you may still see entries in the corresponding native service MIB modules such as the Frame Relay Forum, Internet Engineering Task Force (IETF) standard and Cisco proprietary Frame Relay MIB modules, as well 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.

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, etc. • 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.

Table 1   *cpwVcTable Objects and Descriptions (continued)*

Objects	Description
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.
cpwVcPeerAddrType	Denotes the address type of the peer node maintenance protocol (signaling) address if PW maintenance protocol is used for the VC creation. It should be set to unknown if 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 PWES 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.

Table 1   *cpwVcTable Objects and Descriptions (continued)*

Objects	Description
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.
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 represent 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.

Table 1   cpwVcTable Objects and Descriptions (continued)

Objects	Description
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.
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 exceed 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.

Table 1   cpwVcTable Objects and Descriptions (continued)

Objects	Description
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 represent 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 specific a 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.
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.

Table 7   cpwVcMplsInboundTable Objects and Descriptions (continued)

Objects	Description
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.
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.

Table 9   cpwVcMplsTeMappingTable Objects and Descriptions (continued)

Objects	Description
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.

Table 10   cpwVcEnetTable Objects and Descriptions (continued)

Objects	Description
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.
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).

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—This object is used to indicate the next cpwVcIndex value when adding rows to the cpwVcTable.

- cpwVcNotifRate—This object indicates the rate at which cpwVcUp/Down notifications can be issued from the device.

- cpwVcMplsOutboundIndexNext—This object 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 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.

- cpwVcMplsInboundIndexNext—This object 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 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 access circuits.

How to Configure the PWE3 MIBs

This section contains the following procedures:

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 Circuits Across a Network

For detailed information, see the *Any Transport over MPLS* feature module.

Verifying the PWE3 MIBs Configuration

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

Note This release supports SNMPv1 and SNMPv2c.

Configuration Examples for the PWE3 MIBs

This section provides the following configuration examples:

PWE3 MIB: Examples

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.

There are notifications specific to the PWE3 MIBs. For detailed information on the commands used to configure them, see the "Command Reference" section on page 22.

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
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.
	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*

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 section documents revised commands. All other commands used with this feature are documented in the Cisco IOS Release 12.3 command reference publications.

- **snmp-server enable traps**
- **snmp-server host**

snmp-server enable traps

To enable a label switch router (LSR) to send Simple Network Management Protocol (SNMP) notifications or informs to an SNMP host, use the **snmp-server enable traps** command in global configuration mode. To disable notifications or informs, use the **no** form of this command.

snmp-server enable traps [*notification-type*] [*notification-option*]

no snmp-server enable traps [*notification-type*] [*notification-option*]

Syntax Description	*notification-type*	(Optional) Specifies the particular type of SNMP notification(s) to be enabled on the LSR. If a notification type is not specified, all SNMP notifications applicable to the LSR are enabled and sent to the SNMP host. Any one or all of the following keywords can be specified in any combination as the *notification-type* (family name) in the **snmp-server enable traps** command:
		• **bgp**—Sends Border Gateway Protocol (BGP) state change notifications.
		• **config**—Sends configuration notifications.
		• **entity**—Sends entity MIB modification notifications.
		• **envmon**—Sends Cisco enterprise-specific environmental monitor notifications whenever certain environmental thresholds are exceeded. *Notification-option* arguments (below) can be specified in combination with this keyword.
		• **frame-relay**—Sends Frame Relay notifications.
		• **hsrp**—Sends Hot Standby Routing Protocol (HSRP) notifications.
		• **isdn**—Sends ISDN notifications. *Notification-option* arguments (see below) can be specified in combination with this keyword.
		• **repeater**—Sends Ethernet repeater (hub) notifications. *Notification-option* arguments (see below) can be specified in combination with this keyword.
		• **rsvp**—Sends Resource Reservation Protocol (RSVP) notifications.
		• **rtr**—Sends Service Assurance Agent/Response Time Reporter (RTR) notifications.
		• **snmp** [**authentication**]—Sends RFC 1157 SNMP notifications. Using the **authentication** keyword produces the same effect as not using it. Both the **snmp-server enable traps snmp** and the **snmp-server enable traps snmp authentication** forms of this command globally enable the following SNMP notifications (or, if you are using the **no** form of the command, disables such notifications): **authenticationFailure**, **linkUp**, **linkDown**, and **warmstart**.
		• **syslog**—Sends system error message (Syslog) notifications. You can specify the level of messages to be sent using the **logging history level** command.

notification-type (continued)	• **mpls ldp**—Sends notifications about status changes in LDP sessions. Note that this keyword is specified as **mpls ldp**. This syntax, which the CLI interprets as a two-word construct, has been implemented in this manner to maintain consistency with other MPLS commands. *Notification-option* arguments (below) can be specified in combination with this keyword.
	• **mpls traffic-eng**—Sends notifications about status changes in MPLS label distribution tunnels. This keyword is specified as **mpls traffic-eng**. This syntax, which the CLI interprets as a two-word construct, has been implemented in this manner to maintain consistency with other MPLS commands. *Notification-option* arguments (below) can be specified in combination with this keyword.
	• **pw vc**—Sends notifications about status changes in pseudowire (PW) virtual circuits (VCs).

notification-option	(Optional) Defines the particular options associated with the specified *notification-type* that are to be enabled on the LSR.

- **envmon** [**voltage** | **shutdown** | **supply** | **fan** | **temperature**]

  When you specify the **envmon** keyword, you can enable any one or all of the following environmental notifications in any combination: **voltage**, **shutdown**, **supply**, **fan**, or **temperature**. If you do not specify an argument with the **envmon** keyword, all types of system environmental notifications are enabled on the LSR.

- **isdn** [**call-information** | **isdn u-interface**]

  When you specify the **isdn** keyword, you can use either the **call-information** argument (to enable an SNMP ISDN call information option for the ISDN MIB subsystem) or the **isdn u-interface** argument (to enable an SNMP ISDN U interface option for the ISDN U Interfaces MIB subsystem), or both. If you do not specify an argument with the **isdn** keyword, both types of isdn notifications are enabled on the LSR.

- **repeater** [**health** | **reset**]

  When you specify the **repeater** keyword, you can use either the **health** argument or the **reset** argument, or both (to enable the IETF Repeater Hub MIB [RFC 1516] notification). If you do not specify an argument with the **repeater** keyword, both types of notifications are enabled on the LSR.

- **mpls ldp** [**session-up** | **session-down** | **pv-limit** | **threshold**]

  When you specify the **mpls ldp** keyword, you can use any one or all of the following arguments in any combination to indicate status changes in LDP sessions: **session-up**, **session-down**, **pv-limit**, or **threshold**. If you do not specify an argument with the **mpls ldp** keyword, all four types of LDP session notifications are enabled on the LSR.

- **mpls traffic-eng** [**up** | **down** | **reroute**]

  When you specify the **mpls traffic-eng** keyword, you can use any one or all of the following arguments in any combination to enable the sending of notifications regarding status changes in MPLS label distribution tunnels: **up**, **down**, or **reroute**. If you do not specify an argument with the **mpls traffic-eng** keyword, all three types of tunnel notifications are enabled on the LSR.

- **pw vc** [**up** | **down**]

  When you specify the **pw vc** keyword, you can use either the **up** or the **down** notification. If you do not specify a notification with the **pw vc** keyword, then both are sent.

Defaults   If you issue this command on an LSR without specifying any *notification-type* keywords, the default behavior of the LSR is to enable all notification types controlled by the command (some notification types cannot be controlled by means of this command).

Command Modes   Global configuration

Command History	Release	Modification
	11.1	This command was introduced.
	11.3	The **snmp-server enable traps snmp authentication** form of this command was introduced to replace the **snmp-server trap-authentication** command.
	12.0(17)ST	The **mpls traffic-eng** keyword was added to define a class or family of specific SNMP notifications for use with the *notification-type* and *notification-option* parameters of the **snmp-server enable traps** command.
	12.0(21)ST	The **mpls ldp** keyword was added to define a class or family of specific SNMP notifications for use with the *notification-type* and *notification-option* parameters of the **snmp-server enable traps** command.
	12.0(22)S	This command was integrated into Cisco IOS Release 12.0(22)S.
	12.2(18)S	This command was integrated into Cisco IOS Release 12.2(18)S.
	12.3(8)T	This command was integrated into Cisco IOS Release 12.3(8)T.
	12.0(29)S	The **pw vc** keyword was added to define a class or family of specific SNMP notifications for use with the *notification-type* and *notification-option* parameters of the **snmp-server enable traps** command.

Usage Guidelines

To configure an LSR to send SNMP LDP notifications, you must issue at least one **snmp-server enable traps** command on the router.

To configure an LSR to send either notifications (traps) or informs to a designated network management station (NMS), you must issue the **snmp-server host** command on that device, using the keyword (**traps** or **informs**) that suits your purposes.

If you issue the **snmp-server enable traps** command without keywords, all SNMP notification types are enabled on the LSR. If you issue this command with specific keywords, only the notification types associated with those particular keywords are enabled on the LSR.

The **snmp-server enable traps** command is used in conjunction with the **snmp-server host** command. You use the latter command to specify the NMS host (or hosts) targeted as the recipient(s) of the SNMP notifications generated by SNMP-enabled LSRs in the network. To enable an LSR to send such notifications, you must issue at least one **snmp-server host** command on the LSR.

Examples

In the following example, the router is enabled to send all notifications to the host specified as myhost.cisco.com. The community string is defined as public.

```
Router(config)# snmp-server enable traps

Router(config)# snmp-server host myhost.cisco.com public
```

In the following example, the router is enabled to send Frame Relay and environmental monitor notifications to the host specified as myhost.cisco.com. The community string is defined as public.

```
Router(config)# snmp-server enable traps frame-relay

Router(config)# snmp-server enable traps envmon temperature

Router(config)# snmp-server host myhost.cisco.com public
```

In the following example, notifications are not sent to any host. BGP notifications are enabled for all hosts, but the only notifications enabled to be sent to a host are ISDN notifications (which are not enabled in this example).

```
Router(config)# snmp-server enable traps bgp

Router(config)# snmp-server host bob public isdn
```

In the following example, the router is enabled to send all inform requests to the host specified as myhost.cisco.com. The community string is defined as public.

```
Router(config)# snmp-server enable traps

Router(config)# snmp-server host myhost.cisco.com informs version 2c public
```

In the following example, HSRP MIB notifications are sent to the host specified as myhost.cisco.com. The community string is defined as public.

```
Router(config)# snmp-server enable hsrp

Router(config)# snmp-server host myhost.cisco.com traps version 2c public hsrp
```

In the following example, pseudowire up traps are enabled:

```
Router(config)# snmp-server enable traps pw vc up
```

In the following example, pseudowire down traps are enabled:

```
Router(config)# snmp-server enable traps pw vc down
```

Related Commands	Command	Description
	snmp-server host	Specifies the intended recipient of an SNMP notification (that is, the designated NMS workstation in the network).

snmp-server host

To specify a network management station (NMS) in the network as the intended recipient of Simple Network Management Protocol (SNMP) notifications or informs, use the **snmp-server host** command in global configuration mode. To disable the configuration of the NMS as intended recipient, use the **no** form of this command.

> **snmp-server host** *host-addr* [**vrf** *vrf-name*] [**traps** | **informs**] [**version** {**1** | **2c** | **3** [**auth** | **noauth** | **priv**]}] *community-string* [**udp-port** *port*] [*notification-type*]

> **no snmp-server host** *host-addr* [**traps** | **informs**]

Syntax Description		
host-addr	Specifies the name or the IP address of the host NMS workstation on which the SNMP agent is running. The specified NMS thus serves as the recipient of SNMP notifications or informs.	
vrf *vrf-name*	(Optional) Specifies the name of the Virtual Private Network (VPN) routing/forwarding instance (VRF) for which the NMS is to receive SNMP notifications or informs.	
traps	(Optional) Sends SNMP notifications to the specified NMS host. This is the default of the **snmp-server host** command.	
informs	(Optional) Sends SNMP informs to the specified NMS host.	

version	(Optional) Indicates the SNMP version to be used in sending LDP notifications or informs to the NMS host. Version 3 is the most secure, because it allows packet encryption by means of the **priv** keyword (below). If you use the **version** keyword, you must also specify one of the following arguments:
	
	• **1**—SNMPv1. This option is not available when you are using the **informs** keyword.
	
	• **2c**—SNMPv2c.
	
	• **3**—SNMPv3. The following optional arguments can be used with the **version 3** keyword:
	
	  – **auth**—Enables Message Digest 5 (MD5) and Secure Hash Algorithm (SHA) packet authentication.
	
	  – **noauth**—Indicates the noAuthNoPriv security level. This argument is the default if the [**auth**
		noauth	**priv**] keyword argument is not specified.
			
			  – **priv**—Enables Data Encryption Standard (DES) packet encryption (also called privacy encryption).	

community-string	The community string, functioning much like a password, is sent with the notification or informs operation. Although you can set this string using the **snmp-server host** command by itself, we recommend that you define this string using the **snmp-server community** command before you use the **snmp-server host** command.	
udp-port *port*	User Datagram Protocol (UDP) port number of the NMS host to which SNMP notifications or informs are to be sent. The default UDP port number is 162.	

notification-type	(Optional) Specifies the particular type of SNMP notifications or informs to be sent to the NMS host. If no notification type is specified, all applicable SNMP notifications or informs are sent. One or more of the following can be specified as a keyword in this command:

- **bgp**—Sends Border Gateway Protocol (BGP) state change notifications.

- **config**—Sends configuration notifications.

- **dspu**—Sends downstream physical unit (DSPU) notifications.

- **entity**—Sends entity MIB modification notifications.

- **envmon**—Sends Cisco enterprise-specific environmental monitor notifications when a specified system environmental threshold is exceeded.

- **frame-relay**—Sends Frame Relay notifications.

- **hsrp**—Sends Hot Standby Routing Protocol (HSRP) notifications.

- **isdn**—Sends ISDN notifications.

- **llc2**—Sends Logical Link Control, Type 2 (LLC2) notifications.

- **mpls-ldp**—Sends notifications about status changes in LDP sessions. This keyword (including a hyphen) is seen as one word, enabling you to specify multiple keywords in the **snmp-server host** command (if you delimit each keyword with a space). The same parameter in the **snmp-server enable traps** command is specified as **mpls ldp** (including a space), which results in two words in the *notification-type* and *notification-option* parameters in the **snmp-server enable traps** command. A two-word parameter is consistent with other MPLS commands.

- **mpls-traffic-eng**—Sends notifications indicating status changes in label distribution tunnels. This keyword (including hyphens) is seen as one word, enabling you to specify multiple keywords in the **snmp-server host** command (if you delimit each keyword with a space). The same parameter in the **snmp-server enable traps** command is specified as **mpls traffic-eng** (including a space and a hyphen), which results in two words in the *notification-type* and *notification-option* parameters in the **snmp-server enable traps** command. A two-word parameter is consistent with other MPLS commands.

- **pw-vc**—Sends pseudowire (PW) virtual circuit (VC) notifications.

- **repeater**—Sends standard repeater (hub) notifications.

- **rsrb**—Sends remote source-route bridging (RSRB) notifications.

- **rsvp**—Sends Resource Reservation Protocol (RSVP) notifications.

- **rtr**—Sends SA Agent (Response Time Reporter [RTR]) notifications.

- **sdlc**—Sends Synchronous Data Link Control (SDLC) notifications.

- **sdllc**—Sends SDLC Logical Link Control (SDLLC) notifications.

- **snmp**—Sends SNMP notifications (as defined in RFC 1157).

- **stun**—Sends serial tunnel (STUN) notifications.

- **syslog**—Sends system error message (Syslog) notifications. You can specify the level of messages to be sent using the **logging history level** command.

notification-type (continued)	•	**tty**—Sends Cisco enterprise-specific notifications when a TCP connection terminates.
	•	**x25**—Sends X.25 event notifications.

Defaults

This command is disabled by default, in which case, no SNMP notifications are sent.

If you enter this command without keywords, the default is to send all notification types to the NMS host. No informs are sent to the host.

If no **version** keyword is specified, the default is Version 1. Issuing the **no snmp-server host** command without keywords disables notifications, but not informs, to the NMS host. To disable informs, use the **no snmp-server host informs** command.

Note If the *community-string* is not defined if you use the **snmp-server community** command prior to you issuing the **snmp-server host** command, the default form of the **snmp-server community** command is automatically inserted into the configuration. The password (*community-string*) used for this automatic configuration of the **snmp-server community** is the same as the one specified in the **snmp-server host** command. This is the default behavior for Cisco IOS Release 12.0(3) and later.

Command Modes

Global configuration

Command History

Release	Modification
10.0	This command was introduced.
12.0(17)ST	The **mpls-traffic-eng** keyword was added as a *notification-type* parameter in the **snmp-server host** command to enable the sending of traffic engineering notifications that reflect status changes in label distribution tunnels.
12.0(21)ST	The **mpls-ldp** keyword was added as a *notification-type* parameter in the **snmp-server host** command to enable the sending of LDP notifications that reflect status changes in LDP sessions.
12.0(22)S	This command was integrated into Cisco IOS Release 12.0(22)S.
12.2(18)S	This command was integrated into Cisco IOS Release 12.2(18)S.
12.0(27)S	The **vrf** *vrf-name* keyword argument pair was added to support multiple LDP contexts for VPNs.
12.3(8)T	This command was integrated into Cisco IOS Release 12.3(8)T.
12.0(29)S	The **pw-vc** keyword was added as a notification-type parameter in the **snmp-server host** command to enable the sending of PW VC notifications.

Usage Guidelines

To configure a label switch router (LSR) to send SNMP notifications to an NMS, you must enter at least one **snmp-server host** command on the LSR.

If you issue the **snmp-server host** command without keywords, all SNMP notification types are enabled for the specified NMS host. If you issue this command with specific keywords, only the notification types associated with those particular keywords are enabled for the NMS host.

To specify multiple NMS hosts as recipients of SNMP notifications or informs, you must issue a separate **snmp-server host** command for each targeted NMS host. You can specify multiple notification types in the command for each NMS.

When multiple **snmp-server host** commands are issued for the same NMS host and notification type (trap or inform request), each succeeding command overwrites the previous command. For example, if you issue an **snmp-server host inform** command for an NMS host and then issue another **snmp-server host inform** command for the same NMS host, the second command overrides the first command.

The **snmp-server host** command is used in conjunction with the **snmp-server enable** command. You use the **snmp-server enable** command to specify which SNMP notifications are to be sent globally. For an NMS host to receive most SNMP notifications, at least one **snmp-server enable** command and the **snmp-server host** command for that NMS host must be enabled on the LSR.

Examples

If you want to configure a unique SNMP community string for notifications, but you want to prevent SNMP polling access with this particular string, the configuration should include an access-list. In the following example, the community string is named comaccess and the access list is numbered 10:

```
Router(config)# snmp-server community comaccess ro 10

Router(config)# snmp-server host 172.20.2.160 comaccess

Router(config)# access-list 10 deny any
```

In the following example, SNMP notifications are sent to the NMS host specified as myhost.cisco.com. The community string is defined as comaccess.

```
Router(config)# snmp-server enable traps

Router(config)# snmp-server host myhost.cisco.com comaccess snmp
```

In the following example, SNMP and the Cisco environmental monitor (envmon) enterprise-specific notifications are sent to the NMS host identified by IP address 172.30.2.160:

```
Router(config)# snmp-server enable traps

Router(config)# snmp-server host 172.30.2.160 public snmp envmon
```

In the following example, the LSR is enabled to send all notifications to the SNMP host identified as myhost.cisco.com. The community string is defined as public.

```
Router(config)# snmp-server enable traps

Router(config)# snmp-server host myhost.cisco.com public
```

In the following example, notifications are not sent to any SNMP host. The BGP notifications are enabled for all hosts, but only the ISDN notifications are sent to the host NMS.

```
Router(config)# snmp-server enable traps bgp

Router(config)# snmp-server host bob public isdn
```

In the following example, the LSR is enabled to send all inform requests to the NMS host specified as myhost.cisco.com. The community string is defined as public.

```
Router(config)# snmp-server enable traps

Router(config)# snmp-server host myhost.cisco.com informs version 2c public
```

In the following example, Hot Standby Router Protocol (HSRP) MIB notifications are sent to the NMS host specified as myhost.cisco.com. The community string is defined as public.

```
Router(config)# snmp-server enable hsrp

Router(config)# snmp-server host myhost.cisco.com traps version 2c public hsrp
```

In the following example, the community name is defined as public and SNMP pw-vc-specific notifications are sent to the NMS host identified by IP address 172.19.192.50:

```
Router(config)# snmp-server host 172.19.192.50 public pw-vc
```

Related Commands	Command	Description
	snmp-server enable traps	Enables the LSR on which this command is executed to send SNMP notifications to a designated NMS host.

Glossary

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

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 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—Tunnel whose 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.