MPLS VPN—SNMP MIB Support

This document describes the Simple Network Management Protocol (SNMP) agent support in Cisco IOS for Multiprotocol Label Switching (MPLS) Virtual Private Network (VPN) management, as implemented in the draft *MPLS/BGP Virtual Private Network Management Information Base Using SMIv2* (*draft-ietf-ppvpn-mpls-vpn-mib-03.txt*).

The MPLS VPN technology allows service providers to offer intranet and extranet VPN services that directly connect their customers' remote offices to a public network with the same security and service levels that a private network offers. Each VPN is associated with one or more VPN routing/forwarding instances (VRFs). A VRF is created for each VPN defined on a router and contains most of the information needed to manage and monitor MPLS VPNs: an IP routing table, a derived Cisco Express Forwarding (CEF) table, a set of interfaces that use this forwarding table, and a set of rules and routing protocol parameters that control the information that is included into the routing table. The Provider-Provisioned VPN (PPVPN)-MPLS-VPN MIB provides access to this VRF information, as well as interfaces included in the VRF, and other configuration and monitoring information.

The PPVPN-MPLS-VPN MIB provides the following benefits:

- A standards-based SNMP interface for retrieving information about critical MPLS VPN events.
- VRF information to assist in the management and monitoring of MPLS VPNs.
- Information, in conjunction with the Interfaces MIB, about interfaces assigned to VRFs.
- Performance statistics for all VRFs on a router.
- The generation and queuing of notifications that call attention to major changes in the operational status of MPLS VPN enabled interfaces; the forwarding of notification messages to a designated NMS for evaluation and action by network administrators.
- Advanced warning when VPN routing tables are approaching or exceed their capacity.
- Warnings about the reception of illegal labels on a VRF enabled interface. Such receptions may indicate misconfiguration or an attempt to violate security.

Feature Specifications for the MPLS VPN—SNMP MIB Support

Feature History	
Release	**Modification**
12.0(21)ST	This feature was introduced.
12.0(22)S	This feature was integrated into Cisco IOS Release 12.0(22)S.
12.2(14)S	This feature was integrated into Cisco IOS Release 12.2(14)S.
Supported Platforms	
Cisco 7200 series, Cisco 7500 series	

Determining Platform Support Through Cisco Feature Navigator

Cisco IOS software is packaged in feature sets that are supported on specific platforms. To obtain updated information about platform support for this feature, access Cisco Feature Navigator. Cisco Feature Navigator dynamically updates the list of supported platforms as new platform support is added for the feature.

Cisco Feature Navigator is a web-based tool that enables you to determine which Cisco IOS software images support a specific set of features and which features are supported in a specific Cisco IOS image. You can search by feature or release. In the release section, you can compare releases side by side to display both the features unique to each software release and the features that releases have in common.

To access Cisco Feature Navigator, you must have an account on Cisco.com. If you have forgotten or lost your account information, send a blank e-mail to cco-locksmith@cisco.com. An automatic check will verify that your e-mail address is registered with Cisco.com. If the check is successful, account details with a new random password will be e-mailed to you. Qualified users can establish an account on Cisco.com by following the directions found at this URL:

http://www.cisco.com/register

Cisco Feature Navigator is updated regularly when major Cisco IOS software releases and technology releases occur. For the most current information, go to the Cisco Feature Navigator home page at the following URL:

http://www.cisco.com/go/fn

Availability of Cisco IOS Software Images

Platform support for particular Cisco IOS software releases is dependent on the availability of the software images for those platforms. Software images for some platforms may be deferred, delayed, or changed without prior notice. For updated information about platform support and availability of software images for each Cisco IOS software release, refer to the online release notes or, if supported, Cisco Feature Navigator.

Contents

Prerequisites for MPLS VPN—SNMP MIB Support

The MPLS VPN MIB agent requires the following:

- SNMP is installed and enabled on the label switching routers.
- MPLS is enabled on the label switching routers.
- Multiprotocol Border Gateway Protocol (BGP) is enabled on the label switching routers.
- Cisco Express Forwarding is enabled on the label switching routers.

Restrictions for MPLS VPN—SNMP MIB Support

The following restrictions apply to the PPVPN-MPLS-VPN MIB for this release:

- Configuration of the MIB using the SNMP SET command is not supported in this release, except for trap-related objects, such as mplsVpnNotificationEnable and mplsVpnVrfSecIllegalLabelRcvThresh.
- The mplsVpnVrfBgpNbrPrefixTable is not supported in this version.

Information About MPLS VPN—SNMP MIB Support

SNMP agent code operating in conjunction with the PPVPN-MPLS-VPN MIB enables a standardized, SNMP-based approach to managing MPLS VPNs in Cisco IOS.

The PPVPN-MPLS-VPN MIB is based on the IETF draft MIB specification *draft-ietf-ppvpn-mpls-vpn-mib-03.txt*, which includes objects describing features that support MPLS VPN events. This IETF draft MIB, which undergoes revisions from time to time, is being evolved toward becoming a standard. Accordingly, the Cisco implementation of the PPVPN-MPLS-VPN MIB is expected to track the evolution of the IETF draft MIB, and may change accordingly.

Some slight differences between the IETF draft MIB and the actual implementation of MPLS VPNs within Cisco IOS require some minor translations between the PPVPN-MPLS-VPN MIB and the internal data structures of Cisco IOS. These translations are accomplished by means of the SNMP agent code. Also, while running as a low priority process, the SNMP agent provides a management interface to Cisco IOS. SNMP adds little overhead on the normal functions of the device.

The SNMP objects defined in the PPVPN-MPLS-VPN MIB can be viewed by any standard SNMP utility. The network administrator can retrieve information in the PPVPN-MPLS-VPN MIB using standard SNMP **get** and **getnext** operations for SNMP v1, v2, and v3.

All PPVPN-MPLS-VPN MIB objects are based on the IETF draft MIB; thus, no Cisco specific SNMP application is required to support the functions and operations pertaining to the PPVPN-MPLS-VPN MIB features.

This section contains the following topics:

- Capabilities Supported by PPVPN-MPLS-VPN MIB, page 4
- Functional Structure of the PPVPN-MPLS-VPN MIB, page 4
- Supported Objects in PPVPN-MPLS-VPN MIB, page 5
- MIB Objects Not Supported, page 18

Capabilities Supported by PPVPN-MPLS-VPN MIB

The following functionality is supported in Cisco Release 12.2(14)S for the PPVPN-MPLS-VPN MIB. The PPVPN-MPLS-VPN MIB provides you with the ability to do the following:

- Gather routing and forwarding information for MPLS VPNs on a router.
- Expose information in the VRF routing table.
- Gather information on BGP configuration related to VPNs and VRF interfaces and statistics.
- Emit notification messages that signal changes when critical MPLS VPN events occur.
- Enable, disable, and configure notification messages for MPLS VPN events by using extensions to existing SNMP CLI commands.
- Specify the IP address of a network management system (NMS) in the operating environment to which notification messages are sent.
- Write notification configurations into nonvolatile memory.

Functional Structure of the PPVPN-MPLS-VPN MIB

The SNMP agent code supporting the PPVPN-MPLS-VPN MIB follows the existing model for such code in Cisco IOS and is, in part, generated by the Cisco IOS tool set, based on the MIB source code.

The SNMP agent code, which has a layered structure that is common to MIB support code in Cisco IOS, consists of four layers:

- Platform-independent layer—This layer is generated primarily by the MIB development Cisco IOS tool set and incorporates platform- and implementation-independent functions. The Cisco IOS MIB development tool set creates a standard set of files associated with a MIB.
- Application interface layer—The functions, names, and template code for MIB objects in this layer are also generated by the MIB development Cisco IOS tool set.
- Application-specific layer—This layer provides an interface between the application interface layer and the API and data structures layer below and performs tasks needed to retrieve required information from Cisco IOS, such as searching through data structures.
- API and data structures layer—This layer contains the data structures or APIs within Cisco IOS that are retrieved or called in order to set or retrieve SNMP management information.

Supported Objects in PPVPN-MPLS-VPN MIB

The PPVPN-MPLS-VPN MIB contains numerous tables and object definitions that provide read-only SNMP management support for the MPLS VPN feature in Cisco IOS. The PPVPN-MPLS-VPN MIB conforms to Abstract Syntax Notation One (ASN.1), thus reflecting an idealized MPLS VPN database.

Using any standard SNMP network management application, you can retrieve and display information from the PPVPN-MPLS-VPN MIB using GET operations; similarly, you can traverse information in the MIB database for display using GETNEXT operations.

The PPVPN-MPLS-VPN MIB tables and objects supported in this Cisco IOS release are described briefly in the following sections:

- Scalar Objects, page 6
- MIB Tables, page 7
- Notifications, page 16

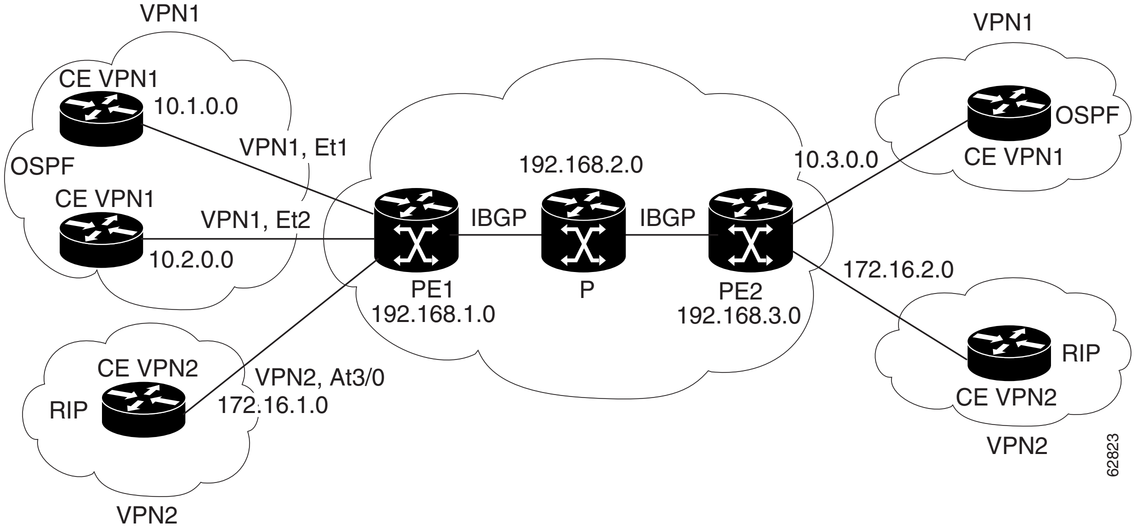
Objects that are not supported in this Cisco IOS release are listed in the "MIB Objects Not Supported" section on page 18.

Figure 1 shows a simple MPLS VPN configuration. This configuration includes two customer MPLS VPNs, labeled VPN1 and VPN2, and a simple provider network that consists of two provider edge routers, labeled PE1 and PE2, and a provider core router labeled P. Figure 1 shows the following sample configuration:

- VRF names—VPN1 and VPN2
- Interfaces associated with VRFs—Et1, Et2, and At3/0
- Routing protocols—OSPF, RIP, and IBGP
- Routes associated with VPN1—10.1.0.0, 10.2.0.0, and 10.3.0.0
- Routes associated with VPN2—172.16.1.0 and 172.16.2.0
- Routes associated with the provider network—192.168.1.0, 192.168.2.0, and 192.168.3.0

This configuration is used in this document to explain MPLS VPN events that are monitored and managed by the PPVPN-MPLS-VPN MIB.

Figure 1 Sample MPLS VPN Configuration



Scalar Objects

Table 1 shows the PPVPN-MPLS-VPN MIB scalar objects supported for this release.

Table 1 PPVPN-MPLS-VPN MIB Scalar Objects

MIB Object	Function
mplsVpnConfiguredVrfs	The number of VRFs configured on the router, including VRFs recently deleted.
mplsVpnActiveVrfs	The number of VRFs that are active on the router. An active VRF is assigned to at least one interface that is in the operationally up state.
mplsVpnConnectedInterfaces	The total number of interfaces assigned to any VRF.
mplsVpnNotificationEnable	A value that indicates whether all the PPVPN-MPLS-VPN MIB notifications are enabled. • Setting this object to true enables all notifications defined in the PPVPN-MPLS-VPN MIB. • Setting it to false disables all notifications defined in the MIB. This is one of the few objects that is writable.
mplsVpnVrfConfMaxPossibleRoutes	A number that indicates the amount of routes that this router is capable of storing. This value cannot be determined because it is based on the amount of available memory in the system. Therefore, this object is set to zero (0).

MIB Tables

The PPVPN-MPLS-VPN MIB implementation for this release supports the following tables described in this section:

- mplsVpnVrfTable, page 7
- mplsVpnInterfaceConfTable, page 8
- mlsVpnVrfRouteTargetTable, page 10
- mplsVpnVrfBgpNbrAddrTable, page 12
- mplsVpnVrfSecTable, page 12
- mplsVpnVrfPerfTable, page 13
- mplsVpnVrfRouteTable, page 13

mplsVpnVrfTable

Entries in the VRF configuration table (mplsVpnVrfTable) represent the VRFs that are defined on the router. This includes recently deleted VRFs. The information in this table is also displayed with the CLI **show ip vrf** EXEC command.

Each VRF is referenced by its VRF name (mplsVpnVrfName).

Table 2 lists the MIB objects and their functions for this table.

Table 2 PPVPN-MPLS-VPN MIB Objects for the mplsVpnVrfTable

MIB Object	Function
mplsVpnVrfName	The name associated with this VRF. When this object is used as an index to a table, the first octet is the string length, and subsequent octets are the ASCII codes of each character. For example, "vpn1" is represented as 4.118.112.110.49.

mplsVpnVrfDescription	The description of the VRF. This is specified with the following configuration command:
	`Router(config)# ip vrf vrf-name`
	`Router(config-vrf)# description vrf-description`

mplsVpnVrfRouteDistinguisher	The route distinguisher for this VRF. This is specified with the following configuration command:
	`Router(config)# ip vrf vrf-name`
	`Router(config-vrf)# rd route-distinguisher`

mplsVpnVrfCreationTime	The value of the sysUpTime when this VRF entry was created.
mplsVpnVrfOperStatus	The operational status of this VRF. A VRF is up (1) when at least one interface associated with the VRF is up. A VRF is down (2) when: • No interfaces exist whose ifOperStatus = up (1). • No interfaces are associated with this VRF.
mplsVpnVrfActiveInterfaces	The number of interfaces assigned to this VRF which are operationally up.
mplsVpnVrfAssociatedInterfaces	The number of interfaces assigned to this VRF, independent of the operational status.

Table 2 PPVPN-MPLS-VPN MIB Objects for the mplsVpnVrfTable (continued)

MIB Object	Function

mplsVpnVrfConfMidRouteThreshold	The middle route threshold. If the amount of routes in the VRF crosses this threshold, an mplsNumVrfRouteMidThreshExceeded notification is sent (if notifications are enabled and configured). You can set this value in configuration mode as a percentage of the maximum as follows:
	`Router(config)# ip vrf vrf-name`
	`Router(config-vrf)# maximum routes maximum mid-as-%-of-max`

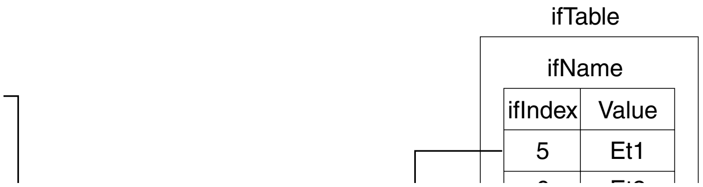
mplsVpnVrfConfHighRouteThreshold	The maximum route threshold. If the amount of routes in the VRF crosses this threshold, an mplsNumVrfRouteMaxThreshExceeded notification is sent (if notifications are enabled and configured). You can set this value in configuration mode as follows:
	`Router(config)# ip vrf vrf-name`
	`Router(config-vrf)# maximum routes maximum`

mplsVpnVrfConfMaxRoutes	This value is the same as the mplsVpnVrfConfHighRouteThreshold.
mplsVpnVrfConfLastChanged	The value of sysUpTime when the configuration of the VRF changes or interfaces are assigned or unassigned from the VRF. **Note** This object is updated only when values in this table change.
mplsVpnVrfConfRowStatus	Read-only implementation. This object normally reads “active (1),” but may read “notInService (2),” if a VRF was recently deleted.
mplsVpnVrfConfStorageType	Read-only implementation. This object always reads “volatile (2).”

mplsVpnInterfaceConfTable

In Cisco IOS, a VRF is associated with one MPLS VPN. Zero or more interfaces can be associated with a VRF. A VRF uses an interface that is defined in the ifTable of the Interfaces Group of MIB II (IFMIB). The IFMIB defines objects for managing interfaces. The ifTable of this MIB contains information on each interface in the network. The mplsVpnInterfaceConfTable associates a VRF from the mplsVpnVrfTable with a forwarding interface from the ifTable. Figure 2 shows the relationship between VRFs and interfaces defined in the ifTable and the mplsVpnInterfaceConfTable.

Figure 2 VRFs, the Interfaces MIB, and the mplsVpnInterfaceConfTable



Entries in the VPN interface configuration table (mplsVpnInterfaceConfTable) represent the interfaces that are assigned to each VRF. The information available in this table is also displayed with the CLI **show ip vrf** EXEC command.

The mplsVpnInterfaceConfTable shows how interfaces are assigned to VRFs. A label switch router (LSR) creates an entry in this table for every interface capable of supporting MPLS VPNs.

The mplsVpnInterfaceConfTable is indexed by the following:

- mplsVpnVrfName—The VRF name
- mplsVpnInterfaceConfIndex—An identifier that is the same as the ifIndex from the Interface MIB of the interface assigned to the VRF

Table 3 lists the MIB objects and their functions for this table.

Table 3 PPVPN-MPLS-VPN MIB Objects for the mplsVpnInterfaceConfTable

MIB Object	Function
mplsVpnInterfaceConfIndex	Provides the interface MIB ifIndex of this interface that is assigned to a VRF.
mplsVpnInterfaceLabelEdgeType	Indicates whether the interface is a provider edge interface (1) or a customer edge interface (2). This value is always providerEdge (1) because in Cisco IOS, customerEdge interfaces are not assigned to VRFs and do not appear in this table.

Table 3 PPVPN-MPLS-VPN MIB Objects for the mplsVpnInterfaceConfTable (continued)

MIB Object	Function
mplsVpnInterfaceVpnClassification	Specifies what type of VPN this interface is providing: carrier supporting carrier (CsC) (1), enterprise (2), or InterProvider (3). This value is set to enterprise (2) if MPLS is not enabled and to carrier supporting carrier (1) if MPLS is enabled on this interface.
mplsVpnInterfaceVpnRouteDistProtocol	Indicates the route distribution protocols that are being used to redistribute routes with BGP on this interface: BGP (2), OSPF (3), or RIP (4). In Cisco IOS, router processes are defined and redistributed on a per-VRF basis, not per-interface. Therefore, all interfaces assigned to the same VRF have the same value for this object.
mplsVpnInterfaceConfStorageType	Read-only implementation. This object always reads "volatile (2)."
mplsVpnInterfaceConfRowStatus	Read-only implementation. This object normally reads "active (1)," but may read "notInService (2)," if a VRF was recently deleted.

mlsVpnVrfRouteTargetTable

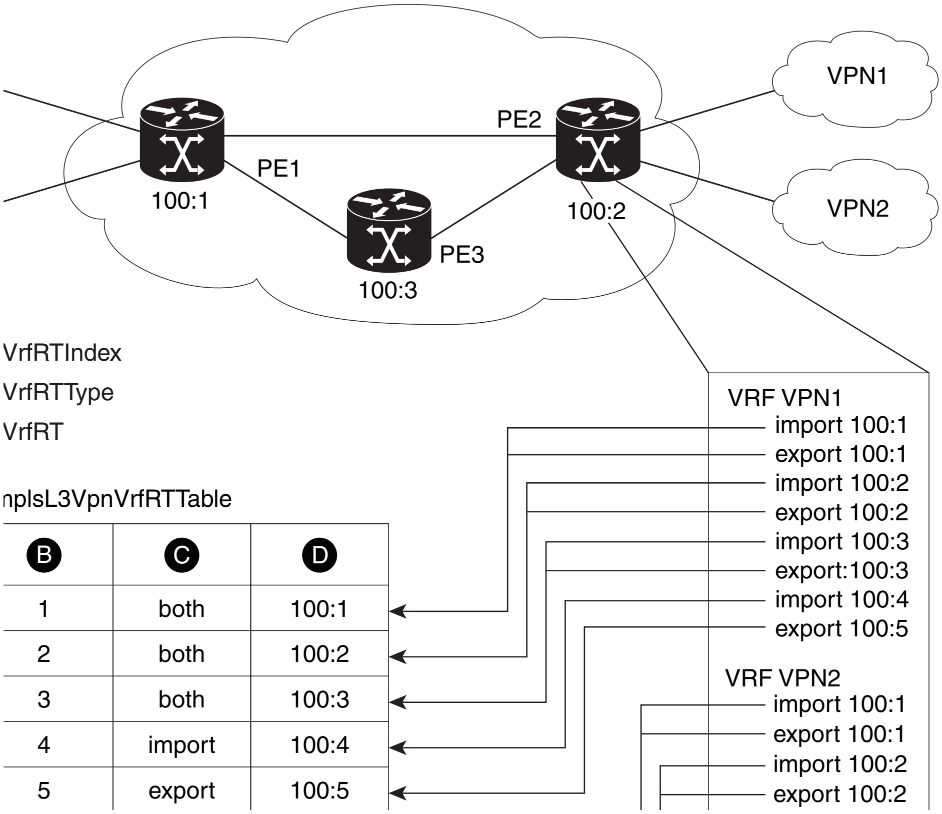
The route target table (mplsVpnVrfRouteTargetTable) describes the route target communities that are defined for a particular VRF. An LSR creates an entry in this table for each target configured for a VRF supporting an MPLS VPN instance.

The distribution of VPN routing information is controlled through the use of VPN route target communities, implemented by Border Gateway Protocol (BGP) extended communities. Distribution of VPN routing information works as follows:

- When a VPN route learned from a CE router is injected into BGP, a list of VPN route target extended community attributes are associated with it. Typically the list of route target community values is set from an export list of route targets associated with the VRF from which the route was learned.
- An import list of route target extended communities is associated with each VRF. The import list defines route target extended community attributes a route must have for the route to be imported into the VRF. For example, if the import list for a particular VRF includes route target communities A, B, and C, then any VPN route that carries any of those route target extended communities—A, B, or C—is imported into the VRF.

Figure 3 shows a sample configuration and its relationship to an mplsVpnVrfRouteTargetTable. A route target table exists on each PE router. Routers with route distinguishers (RDs) 100:1, 100:2, and 100:3 are shown in the sample configuration. Routers with RDs 100:4 and 100:5 are not shown in Figure 3, but are included in the route targets for PE2 and in the mplsVpnVrfRouteTargetTable.

Figure 3 Sample Configuration and the mplsVpnVrfRouteTargetTable



B	C	D
1	both	100:1
2	both	100:2
3	both	100:3
4	import	100:4
5	export	100:5

The mplsVpnVrfRouteTargetTable shows the import and export route targets for each VRF. The table is indexed by the following:

- mplsVpnVrfName—The VRF name
- mplsVpnVrfRouteTargetIndex—The route target entry identifier
- mplsVpnVrfRouteTargetType—A value specifying whether the entry is an import route target, export route target, or is defined as both

Table 4 lists the MIB objects and their functions for this table.

Table 4 PPVPN-MPLS-VPN MIB Objects for the mplsVpnVrfRouteTargetTable

MIB Object	Function
mplsVpnVrfRouteTargetIndex	A value that defines each route target's position in the table.
mplsVpnVrfRouteTargetType	Determines which type of route target the entry represents: import (1), export (2), or both (3).
mplsVpnVrfRouteTarget	Determines the route distinguisher for this target.
mplsVpnVrfRouteTargetDescr	Description of the route target. This object is not supported in this Cisco IOS release. Therefore, the object is the same as mplsVpnVrfRouteTarget.
mplsVpnVrfRouteTargetRowStatus	Read-only implementation. This object normally reads "active (1)," but may read "notInService (2)," if a VRF was recently deleted.

mplsVpnVrfBgpNbrAddrTable

The BGP neighbor address table (mplsVpnVrfBgpNbrAddrTable) represents the MPLS eBGP neighbors that are defined for a particular VRF. An LSR creates an entry for every BGP neighbor that is defined in the VRF's address-family.

The mplsVpnVrfBgpNbrAddrTable is indexed by the following:

- The VRF name (mplsVpnVrfName)
- An identifier that is the same as the ifIndex from the Interface MIB of the interface assigned to the VRF (mplsVpnInterfaceConfIndex)
- The IP address of the neighbor (mplsVpnVrfBgpNbrIndex)

Table 5 lists the MIB objects and their functions for this table.

Table 5 PPVPN-MPLS-VPN MIB Objects for the mplsVpnVrfBgpNbrAddrTable

MIB Object	Function
mplsVpnVrfBgpNbrIndex	The IPv4 address of the eBGP neighbor.
mplsVpnVrfBgpNbrRole	The role of this eBGP neighbor: customer edge (1) or provider edge (2). If the object mplsVpnInterfaceVpnClassification is carrier supporting carrier (CSC), then this value is provider edge (2), otherwise, this value is customer edge (1).
mplsVpnVrfBgpNbrType	Address type of this eBGP neighbor. The MIB only supports IPv4 (1). Therefore, this object returns "ipv4 (1)."
mplsVpnVrfBgpNbrAddr	IP address of the eBGP neighbor.
mplsVpnVrfBgpNbrRowStatus	Read-only implementation. This object normally reads "active (1)," but may read "notInService (2)" if a VRF was recently deleted.
mplsVpnVrfBgpNbrStorageType	Read-only implementation. This object always reads "volatile (2)."

mplsVpnVrfSecTable

The VRF security table (mplsVpnVrfSecTable) provides information about security for each VRF. An LSR creates an entry in this table for every VRF capable of supporting MPLS VPN.

The mplsVpnVrfSecTable *augments* the mplsVpnVrfTable and has the same indexing.

Table 6 lists the MIB objects and their functions for this table.

Table 6 PPVPN-MPLS-VPN MIB Objects for the mplsVpnVrfSecTable

MIB Object	Function
mplsVpnVrfSecIllegalLabelViolations	The number of illegally received labels on a VRF interface. Only illegal labels are counted by this object, therefore the object only applies to a VRF interface that is MPLS enabled (carrier supporting carrier [CsC] situation). This counter is incremented whenever a label is received that is above or below the valid label range, not in the global label forwarding table, or is received on the wrong VRF (that is, table IDs for the receiving interface and appropriate VRF label forwarding table do not match).
mplsVpnVrfSecIllegalLabelRcvThresh	Notification threshold for illegal labels received on this VRF. When the amount of illegal labels received on this interface crosses this threshold, an mplsNumVrfSecIllegalLabelThreshExceeded notification is sent (if the notification is enabled and configured). This object is one of the few in this MIB agent that supports the SNMP SET operation, which allows you to change this value.

mplsVpnVrfPerfTable

The VRF performance table (mplsVpnVrfPerfTable) provides statistical performance information for each VRF. An LSR creates an entry in this table for every VRF capable of supporting MPLS VPN.

The mplsVpnVrfPerfTable *augments* the mplsVpnVrfTable and has the same indexing.

Table 7 lists the MIB objects and their functions for this table.

Table 7 PPVPN-MPLS-VPN MIB Objects for the mplsVpnVrfPerfTable

MIB Objects	Functions
mplsVpnVrfPerfRoutesAdded	The number of routes added to this VRF over the course of its lifetime.
mplsVpnVrfPerfRoutesDeleted	The number of routes removed from this VRF.
mplsVpnVrfPerfCurrNumRoutes	The number of routes currently defined within this VRF.

mplsVpnVrfRouteTable

The VRF routing table (mplsVpnVrfRouteTable) provides the IP routing table information for each VRF. The information available in this table can also be accessed with the CLI **show ip route vrf** *vrf-name* command. For example, for PE1 in Figure 1:

- With the **show ip route vrf vpn1** command, you would see results like the following:

```
Router# show ip route vrf vpn1

Codes: C - connected, S - static, I - IGRP, R - RIP, M - mobile, B - BGP
    D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
    N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
    E1 - OSPF external type 1, E2 - OSPF external type 2, E - EGP
    i - IS-IS, L1 - IS-IS level-1, L2 - IS-IS level-2, ia - IS-IS inter area
    * - candidate default, U - per-user static route, o - ODR
    P - periodic downloaded static route
    !
```

```
Gateway of last resort is not set
!
     10.0.0.0/32 is subnetted, 3 subnets
B       10.3.0.0 [200/0] via 192.168.2.1, 04:36:33
C       10.1.0.0/16 is directly connected, Ethernet1
C       10.2.0.0/16 [200/0] directly connected Ethernet2, 04:36:33
```

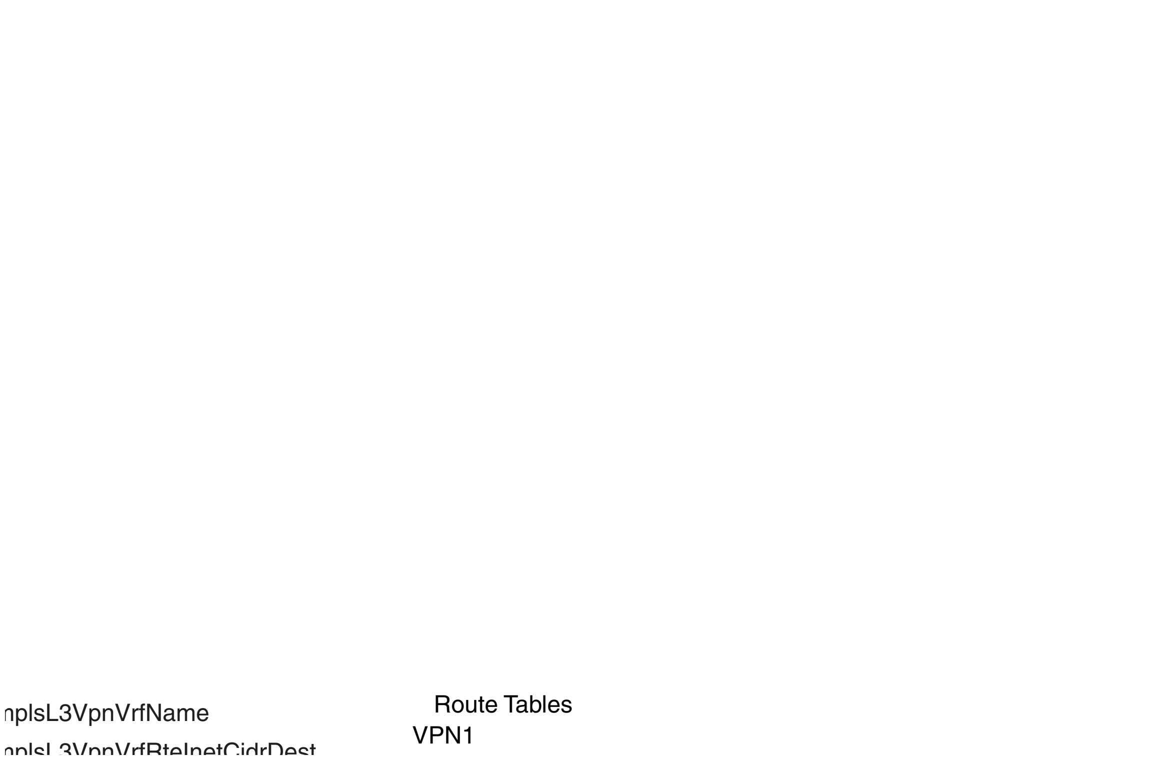
- With the **show ip route vrf vpn2** command, you would see results like the following:

```
Router# show ip route vrf vpn2

Codes: C - connected, S - static, I - IGRP, R - RIP, M - mobile, B - BGP
    D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
    N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
    E1 - OSPF external type 1, E2 - OSPF external type 2, E - EGP
    i - IS-IS, L1 - IS-IS level-1, L2 - IS-IS level-2, ia - IS-IS inter area
    * - candidate default, U - per-user static route, o - ODR
    P - periodic downloaded static route
    !
Gateway of last resort is not set
!
     172.16.0.0/32 is subnetted, 2 subnets
B       172.16.2.0 [200/0] via 192.168.2.1, 04:36:33
C       172.16.1.0 is directly connected, ATM 3/0
```

Figure 4 shows the relationship of the routing tables, the VRFs, and the mplsVpnVrfRouteTable. You can view information about the VPN1 and VPN2 route tables using the **show ip route vrf** *vrf-name* CLI command. The global route table is the same as ipCidrRouteTable in the IP-FORWARD-MIB. You can view information about the global route table with the **show ip route** command.

Figure 4 Route Table, VRFs, and the mplsVpnVrfRouteTable



An LSR creates an entry in this table for every route that is configured, either dynamically or statically, within the context of a specific VRF capable of supporting MPLS VPN.

The mplsVpnVrfRouteTable is indexed by the following:

- mplsVpnVrfName—The VRF name, which provides the VRF routing context
- mplsVpnVrfRouteDest—The IP destination address
- mplsVpnVrfRouteMask—The IP destination mask
- mplsVpnVrfRouteTos—The IP header ToS bits
- mplsVpnVrfRouteNextHop—The IP address of the next hop for each route entry

Note The ToS bits are not supported in this Cisco IOS release and, therefore, are always 0.

Table 8 lists the MIB objects and their functions for the mplsVpnVrfRouteTable. This table represents VRF-specific routes. The global routing table is the ipCidrRouteTable in the IP-FORWARD-MIB.

Table 8 PPVPN-MPLS-VPN MIB Objects for the mplsVpnVrfRouteTable

MIB Object	Function
mplsVpnVrfRouteDest	The destination IP address defined for this route.
mplsVpnVrfRouteDestAddrType	The address type of the IP destination address (mplsVpnVrfRouteDest). This MIB implementation only supports IPv4 (1). Therefore, this object has a value of “ipv4 (1).”
mplsVpnVrfRouteMask	The destination IP address mask defined for this route.
mplsVpnVrfRouteMaskAddrType	The address type of the destination IP address mask. This MIB implementation only supports IPv4 (1). Therefore, this object has a value of “ipv4 (1).”
mplsVpnVrfRouteTos	The ToS bits from the IP header for this route. Cisco IOS only supports ToS bits of zero. Therefore, the object is always 0.
mplsVpnVrfRouteNextHop	The next hop IP address defined for this route.
mplsVpnVrfRouteNextHopAddrType	The address type of the next hop IP address. This MIB implementation only supports IPv4 (1). Therefore, this object has a value of “ipv4 (1).”
mplsVpnVrfRouteIfIndex	The interface MIB ifIndex for the interface through which this route is forwarded. The object is 0 if no interface is defined for the route.
mplsVpnVrfRouteType	Defines if this route is a local or remotely defined route.
mplsVpnVrfRouteProto	The routing protocol that was responsible for adding this route to the VRF.
mplsVpnVrfRouteAge	The number of seconds since this route was last updated.
mplsVpnVrfRouteInfo	A pointer to more information from other MIBs. This object is not supported and always returns “nullOID (0.0).”
mplsVpnVrfRouteNextHopAS	The autonomous system number of the next hop for this route. This object is not supported and is always 0.
mplsVpnVrfRouteMetric1	The primary routing metric used for this route.
mplsVpnVrfRouteMetric2 mplsVpnVrfRouteMetric3 mplsVpnVrfRouteMetric4 mplsVpnVrfRouteMetric5	Alternate routing metrics used for this route. These objects are supported only for Cisco IGRP and Cisco EIGRP. These objects display the bandwidth metrics used for the route. Otherwise, these values are set to –1.

Table 8 PPVPN-MPLS-VPN MIB Objects for the mplsVpnVrfRouteTable (continued)

MIB Object	Function
mplsVpnVrfRouteRowStatus	Read-only implementation.This object normally reads “active (1),” but may read “notInService (2),” if a VRF was recently deleted.
mplsVpnVrfRouteStorageType	Read-only implementation. This object always reads “volatile (2).”

Notifications

This section provides the following information about PPVPN-MPLS-VPN MIB notifications supported in this release:

- Notification Generation Events, page 16
- Notification Specification, page 18
- Monitoring the PPVPN-MPLS-VPN MIB Notifications, page 18

Notification Generation Events

The following notifications of the PPVPN-MPLS-VPN MIB are implemented for this release:

- **mplsVrfIfUp**—Sent to an NMS when an interface comes up and is assigned a VPN routing/forwarding table instance (VRF).
- **mplsVrfIfDown**—Generated and sent to the NMS when a VRF is removed from an interface or the interface transitions from an operationally “up” state to a “down” state.

Note For the mplsVrfIfUp or mplsVrfIfDown notifications to be issued on ATM or Frame Relay subinterfaces, you must configure the **snmp-server traps atm subif** command or the **snmp-server traps frame-relay subif** command on the subinterfaces, respectively.

- **mplsNumVrfRouteMidThreshExceeded**—Generated and sent when the middle (warning) threshold is crossed. You can configure this threshold in the CLI by using the following commands:

  ```
  Router(config)# ip vrf vrf-name

  Router(config-vrf)# maximum routes <max-thresh> <mid-thresh (% of max)>
  ```

  This notification is sent to the NMS only at the time the threshold is exceeded. (See Figure 5 for a comparison of the warning and maximum thresholds.) Whenever the number of routes falls below this threshold and exceeds the threshold again, a notification is sent to the NMS.

- **mplsNumVrfRouteMaxThreshExceeded**—Generated and sent when you attempt to create a route on a VRF that already contains the maximum number of routes as defined by the following CLI commands:

  ```
  Router(config)# ip vrf vrf-name

  Router(config-vrf)# maximum routes <max-thresh> <mid-thresh (% of max)>
  ```

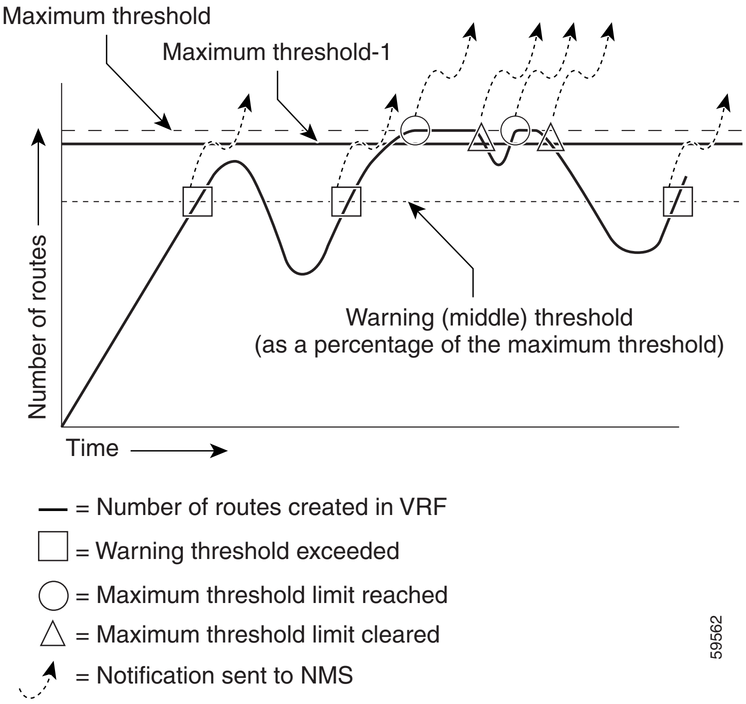
  A trap notification is sent to the NMS when you attempt to exceed the maximum threshold. Another notification is not sent until the number of routes falls below the maximum threshold and reaches the maximum threshold again. (See Figure 5 for an example of how this notification works and for a comparison of the maximum and warning thresholds.)

Note The **maximum routes** command sets the number of routes for a VRF. You *cannot* exceed the number of routes in the VRF that you set with the **maximum routes** *max-thresh* CLI command.

Prior to this implementation of the PPVPN-MPLS-VPN MIB, you were not notified when this threshold (or the warning threshold) was reached.

- **mplsNumVrfSecIllegalLabelThreshExceeded**—Generated and sent when the amount of illegal labels received on a VRF interface exceeds the threshold *mplsVpnVrfSecIllegalLabelRcvThresh*. This threshold is defined with a value of 0. Therefore, a notification is sent when the first illegal label is received on a VRF. Labels are considered illegal if they are outside of the valid label range, do not have a Label Forwarding Information Base (LFIB) entry, or the table ID of the message does not match the table ID for the label in the LFIB.

Figure 5 Comparison of Warning and Maximum Thresholds


Maximum threshold
Maximum threshold-1
Warning (middle) threshold
(as a percentage of the maximum threshold)
Number of routes
Time
= Number of routes created in VRF
= Warning threshold exceeded
= Maximum threshold limit reached
= Maximum threshold limit cleared
= Notification sent to NMS
59562

For information on the Cisco IOS CLI commands for configuring PPVPN-MPLS-VPN MIB notifications that are to be sent to an NMS, see the "How to Configure MPLS VPN—SNMP MIB Support" section on page 19 and the "Command Reference" section on page 29.

Notification Specification

In an SNMPv1 notification, each VPN notification has a generic type identifier and an enterprise-specific type identifier for identifying the notification type.

- The generic type for all VPN notifications is "enterpriseSpecific" as this is not one of the generic notification types defined for SNMP.
- The enterprise-specific type is identified as follows:
  - 1 for *mplsVrfIfUp*
  - 2 for *mplsVrfIfDown*
  - 3 for *mplsNumVrfRouteMidThreshExceeded*
  - 4 for *mplsNumVrfRouteMaxThreshExceeded*
  - 5 for *mplsNumVrfSecIllegalLabelThreshExceeded*

In SNMPv2, the notification type is identified by an **SnmpTrapOID** varbind (variable binding consisting of an object identifier [OID] type and value) included within the notification message.

Each notification also contains two additional objects from the PPVPN-MPLS-VPN MIB. These objects provide additional information about the event, as follows:

- The VRF interface up/down notifications provide additional variables—*mplsVpnInterfaceConfIndex* and *mplsVpnVrfName*—in the notification. These variables describe the SNMP interface index and the VRF name, respectively.
- The mid and max threshold notifications include the *mplsVpnVrfName* variable (VRF name) as well as the *mplsVpnVrfPerfCurrNumRoutes* variable that indicates the current number of routes within the VRF.
- The illegal label notification includes the *mplsVpnVrfName* variable (VRF name) and the *mplsVpnVrfSecIllegalLabelViolations* variable that maintains the current count of illegal labels on a VPN.

Monitoring the PPVPN-MPLS-VPN MIB Notifications

When PPVPN-MPLS-VPN MIB notifications are enabled (see the **snmp-server enable traps** command), notification messages relating to specific MPLS VPN events within Cisco IOS are generated and sent to a specified NMS in the network. Any utility that supports SNMPv1 or SNMPv2 notifications can receive notification messages.

To monitor PPVPN-MPLS-VPN MIB notification messages, log in to an NMS that supports a utility that displays SNMP notifications, and start the display utility.

MIB Objects Not Supported

The following objects from the mplsVpnVrfBgpPathAttrTable are not supported for this release:

- mplsVpnVrfBgpPathAttrPeer
- mplsVpnVrfBgpPathAttrIpAddrPrefixLen
- mplsVpnVrfBgpPathAttrIpAddrPrefix
- mplsVpnVrfBgpPathAttrOrigin
- mplsVpnVrfBgpPathAttrASPathSegment
- mplsVpnVrfBgpPathAttrNextHop

- mplsVpnVrfBgpPathAttrMultiExitDisc
- mplsVpnVrfBgpPathAttrLocalPref
- mplsVpnVrfBgpPathAttrAtomicAggregate
- mplsVpnVrfBgpPathAttrAggregatorAS
- mplsVpnVrfBgpPathAttrAggregatorAddr
- mplsVpnVrfBgpPathAttrCalcLocalPref
- mplsVpnVrfBgpPathAttrBest
- mplsVpnVrfBgpPathAttrUnknown

How to Configure MPLS VPN—SNMP MIB Support

This section describes configuration tasks for the MPLS VPN MIB agent. Each task in the list is identified as either required or optional.

- Enable SNMP Master Agent, page 19 (required)
- Verify the Status of SNMP Master Agent, page 21 (optional)
- Configure the Router to Send SNMP Traps, page 21 (required)
- Configure Threshold Values for MPLS VPN-MIB Notifications, page 24 (required)

The MPLS VPN notifications are enabled or disabled using the extended CLI commands (see the "Command Reference" section on page 29).

Enable SNMP Master Agent

The SNMP master agent for the PPVPN-MPLS-VPN MIB notifications is disabled by default.

Perform this task to enable the SNMP master agent for the PPVPN-MPLS-VPN MIB notifications.

SUMMARY STEPS

1. **telnet** *xxx.xxx.xxx.xxx*
2. **enable**
3. **show running-config**
4. **configure** {**terminal** | **memory** | **network**}
5. **snmp-server community** *string* [**view** *view-name*] [**ro** | **rw**] [*number*]
6. **exit**
7. **write memory**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**telnet** *xxx.xxx.xxx.xxx* **Example:** `Prompt# telnet xxx.xxx.xxx.xxx`	Telnets to the router identified by the specified IP address (represented as *xxx.xxx.xxx.xxx*).
Step 2	**enable** **Example:** `Router> enable`	Enables higher privilege levels, such as privileged EXEC mode. Enter your password if prompted.
Step 3	**show running-config** **Example:** `Router# show running-config`	Displays the running configuration to determine if an SNMP agent is already running. 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 needed.

Step 4	**configure** {**terminal** | **memory** | **network**}
	Example:
	`Router# configure terminal`
		Enters global configuration mode.

Step 5	**snmp-server community** *string* [**view** *view-name*] [**ro** | **rw**] [*number*] **Example:** `Router(config)# snmp-server community comaccess ro`	Sets up the community access string to permit access to the SNMP protocol. • The *string* argument acts like a password and permits access to the SNMP protocol. • The **view** *view-name* keyword argument pair specifies the name of a previously defined view. The view defines the objects available to the community. • The **ro** keyword specifies read-only access. Authorized management stations are only able to retrieve MIB objects. • The **rw** keyword specifies read-write access. Authorized management stations are able to both retrieve and modify MIB objects. • The *number* argument is an integer from 1 to 99 that specifies an access list of IP addresses that are allowed to use the community string to gain access to the SNMP agent.
Step 6	**exit** **Example:** `Router(config)# exit`	Exits the global configuration mode and returns you to the privileged EXEC mode.
Step 7	**write memory** **Example:** `Router# write memory`	Writes the modified configuration to nonvolatile memory (NVRAM), permanently saving the settings.

Verify the Status of SNMP Master Agent

Perform this task to verify that the SNMP master agent is enabled on a host network device.

SUMMARY STEPS

1. **telnet** *xxx.xxx.xxx.xxx*
2. **enable**
3. **show running-config** [**interface** | **map-class**]
4. **disable**

DETAILED STEPS

	Command or Action	Purpose

Step 1	`telnet` *`xxx.xxx.xxx.xxx`*
	Example:
	`Prompt# telnet 10.10.10.10`
		(Optional) Telnets to the router identified by the specified IP address (represented as *xxx.xxx.xxx.xxx*).

Step 2	`enable`
	Example:
	`Router> enable`
		(Optional) Enables higher privilege levels, such as privileged EXEC mode.
		Enter your password if prompted.

Step 3	`show running-config [interface	map-class]` **Example:** `Router# show running-config`	(Optional) Displays the configuration information currently running on the router, the configuration for a specific interface, or map class information. Use the **show running-config** command to check that **snmp-server** statements appear in the output.

Step 4	`disable`
	Example:
	`Router# disable`
		(Optional) Exits to user EXEC mode.

Configure the Router to Send SNMP Traps

Perform this task to configure the router to send traps to a host.

The **snmp-server host** command specifies which hosts receive traps. The **snmp-server enable traps** command globally enables the trap production mechanism for the specified traps.

For a host to receive a trap, an **snmp-server host** command must be configured for that host, and, generally, the trap must be enabled globally through the **snmp-server enable traps** command.

Note Although you can set the *community-string* argument using the **snmp-server host** command by itself, we recommend you define this string using the **snmp-server community** command prior to using the **snmp-server host** command.

SUMMARY STEPS

1. **telnet** *xxx.xxx.xxx.xxx*
2. **enable**
3. **configure** {**terminal** | **memory** | **network**}
4. **snmp-server host** *host-addr* [**traps** | **informs**] [**version** {**1** | **2c** | **3** [**auth** | **noauth** | **priv**]}] *community-string* [**udp-port** *port*] [*notification-type*]
5. **snmp-server enable traps atm** [**pvc** | **subif**] or **snmp-server enable traps frame-relay** [**subif**]
6. **snmp-server enable traps** [*notification-type*] [*notification-option*]
7. **end**

DETAILED STEPS

	Command or Action	Purpose

Step 1	**`telnet`** *`xxx.xxx.xxx.xxx`*
	Example:
	`Prompt# telnet xxx.xxx.xxx.xxx`
		Telnets to the router identified by the specified IP address (represented as *xxx.xxx.xxx.xxx*).

Step 2	**`enable`**
	Example:
	`Router> enable`
		Enables higher privilege levels, such as privileged EXEC mode.
		Enter your password if prompted.

Step 3	**`configure`** `{`**`terminal`** `|` **`memory`** `|` **`network`**`}`
	Example:
	`Router# configure terminal`
		Enters global configuration mode.

	Command or Action	Purpose
Step 4	**snmp-server host** *host-addr* [**traps** | **informs**] [**version** {**1** | **2c** | **3** [**auth** | **noauth** | **priv**]}] *community-string* [**udp-port** *port*] [*notification-type*] **Example:** `Router(config)# snmp-server host 172.20.2.160 traps comaccess mpls-vpn`	Specifies the recipient of an SNMP notification operation. • The *host-addr* argument specifies the name or Internet address of the host (the targeted recipient). • The **traps** keyword sends SNMP traps to this host. This is the default. • The **informs** keyword sends SNMP informs to this host. • The **version** keyword specifies the version of the SNMP used to send the traps. Version 3 is the most secure model, as it allows packet encryption with the **priv** keyword. If you use the **version** keyword, you must specify one of the following: – **1** —SNMPv1. This option is not available with informs. – **2c** —SNMPv2C. – **3** —SNMPv3. The following three optional keywords can follow the **version 3** keyword (**auth**, **noauth**, **priv**). • The *community-string* argument is a password-like community string sent with the notification operation. • The **udp-port** *port* keyword argument pair names the UDP port of the host to use. The default is 162. • The *notification-type* argument specifies the type of notification to be sent to the host. If no type is specified, all notifications are sent.

Step 5	**snmp-server enable traps atm** [**pvc** | **subif**]
	or
	snmp-server enable traps frame-relay [**subif**]
	Example:
	`Router(config)# snmp-server enable traps atm subif`
	or
	Example:
	`Router(config)# snmp-server enable traps frame-relay subif`
		(For ATM subinterfaces only) Enables the sending of ATM SNMP notifications.
		• The **pvc** keyword enables SNMP ATM permanent virtual circuit (PVC) traps.
		• The **subif** keyword enables SNMP ATM subinterface traps.
		or
		(For Frame Relay subinterfaces only) Enables Frame Relay DLCI link status SNMP notifications.
		• The **subif** keyword enables SNMP Frame Relay subinterface traps.
		Note For mplsVrfIfUp or mplsVrfIfDown notifications to be issued on ATM or Frame Relay subinterfaces, you must configure the appropriate **snmp-server enable traps** command with the **subif** keyword.

	Command or Action	Purpose
Step 6	**snmp-server enable traps** [*notification-type*] [*notification-option*] **Example:** `Router(config)# snmp-server enable traps mpls vpn vrf-up vrf-down`	Enables the router to send SNMP traps or informs (SNMP notifications). • The *notification-type* argument specifies the type of notification to enable. If no type is specified, all notifications available on your device are sent. • *The notification-option* argument specifies the notification option.

Step 7	**end**
	Example:
	`Router(config)# end`
		(Optional) Exits to user EXEC mode.

Configure Threshold Values for MPLS VPN-MIB Notifications

Perform this task to configure threshold values for MPLS VPN-MIB notifications.

The **mplsNumVrfRouteMidThreshExceeded** notification event is generated and sent when the middle (warning) threshold is crossed. You can configure this threshold in the CLI by using the **maximum routes** command in VRF configuration mode. This notification is sent to the NMS only at the time the threshold is exceeded. Whenever the number of routes falls below this threshold and exceeds the threshold again, a notification is sent to the NMS.

The **mplsNumVrfRouteMaxThreshExceeded** notification event is generated and sent when you attempt to create a route on a VRF that already contains the maximum number of routes as defined by the **maximum routes** command in VRF configuration mode. A trap notification is sent to the NMS when you attempt to exceed the maximum threshold. Another notification is not sent until the number of routes falls below the maximum threshold and reaches the maximum threshold again.

(See Figure 5 for an example of how this notification works and for a comparison of the maximum and warning thresholds.)

Note The **maximum routes** command sets the number of routes for a VRF. You *cannot* exceed the number of routes in the VRF that you set with the **maximum routes** *max-thresh* CLI command.

Prior to this implementation of the PPVPN-MPLS-VPN MIB, you were not notified when this threshold (or the warning threshold) was reached.

SUMMARY STEPS

1. **enable**
2. **configure** {**terminal** | **memory** | **network**}
3. **ip vrf** *vrf-name*
4. **maximum routes** *limit* {*warn threshold* | **warn-only**}
5. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables higher privilege levels, such as privileged EXEC mode. Enter your password if prompted.

Step 2	**configure** {**terminal** | **memory** | **network**}
	Example:
	`Router# configure terminal`
		Enters global configuration mode.

Step 3	**ip vrf** *vrf-name* **Example:** `Router(config)# ip vrf vpn1`	Configures a VRF routing table. • The *vrf-name* argument specifies the name assigned to a VRF.

Step 4	**maximum routes** *limit* {*warn threshold* | **warn-only**}
	Example:
	`Router(config-vrf)# maximum routes 10000 80`
		Limits the maximum number of routes in a VRF to prevent a PE router from importing too many routes.
		• The *limit* argument specifies the maximum number of routes allowed in a VRF. You may select from 1 to 4,294,967,295 routes to be allowed in a VRF.
		• The *warn threshold* argument specifies when the threshold limit is reached and routes are rejected. The threshold limit is a percentage of the limit specified, from 1 to 100.
		• The **warn-only** keyword specifies that a SYSLOG error message is issued when the maximum number of routes allowed for a VRF exceeds the threshold. However, additional routes are still allowed.

Step 5	**end**
	Example:
	`Router(config-vrf)# end`
		(Optional) Exits to privileged EXEC mode.

Configuration Examples for MPLS VPN—SNMP MIB Support

This section contains the following configuration examples for the MPLS VPN-SNMP MIB Support feature:

- Enable SNMP Master Agent Examples, page 26
- Verify the Status of SNMP Master Agent Example, page 26
- Configure the Router to Send SNMP Traps Examples, page 26
- Configure Threshold Values for MPLS VPN-MIB Notifications Example, page 26

Enable SNMP Master Agent Examples

The following example shows how to enable an SNMP agent on a host network device:

```
Router# config terminal

Router(config)# snmp-server community
```

The following example shows how to enable SNMPv1 and SNMPv2C. The configuration permits any SNMP agent to access all PPVPN-MPLS-VPN MIB objects with read-only permissions using the community string public.

```
Router(config)# snmp-server community public
```

Verify the Status of SNMP Master Agent Example

The following example verifies that the SNMP master agent is enabled for the MPLS VPN-MIB Notifications feature.

```
Router# show running-config

Building configuration...
....
....
snmp-server community public RO
snmp-server community private RW
....
```

Note If you do not see any “snmp-server” statements, SNMP has not been enabled on the router.

Configure the Router to Send SNMP Traps Examples

The following example shows how to allow read-only access to all PPVPN-MPLS-VPN MIB notifications relating to members of access list 4 that specify the comaccess community string. No other SNMP agents will have access to any PPVPN-MPLS-VPN MIB notifications.

```
Router(config)# snmp-server community comaccess ro 4

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

The following example shows you how to enable all SNMP traps defined for MPLS VPN on host 172.31.156.34 using the comaccess community string.

```
Router(config)# snmp-server host 172.31.156.34 comaccess mpls-vpn

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

Configure Threshold Values for MPLS VPN-MIB Notifications Example

The following example shows how to set a maximum threshold of 10000 routes and a warning threshold that is 80 percent of the maximum threshold for a VRF named vpn1 on a router:

```
Router(config)# ip vrf vpn1

Router(config)# maximum routes 10000 80
```

Additional References

For additional information related to the MPLS VPN—SNMP MIB Support feature, refer to the following references:

- Related Documents, page 28
- Standards, page 28
- MIBs, page 28
- RFCs, page 29
- Technical Assistance, page 29

Related Documents

Related Topic	Document Title
Multiprotocol Label Switching (MPLS) Label Distribution Protocol (LDP) configuration tasks	*MPLS Label Distribution Protocol*
A description of SNMP agent support in Cisco IOS for the MPLS Label Switching Router MIB (MPLS-LSR-MIB)	*MPLS Label Switching Router MIB*
MPLS Virtual Private Network (VPN) configuration tasks	*MPLS Virtual Private Networks (VPNs)*
Configuration tasks for MPLS ATM network enhancements	*MPLS Scalability Enhancements for the ATM LSR*
MPLS automatic bandwidth adjustment configuration tasks	*MPLS Traffic Engineering (TE)—Automatic Bandwidth Adjustment for TE Tunnels*
MPLS traffic engineering scalability enhancements configuration tasks	*MPLS Traffic Engineering (TE)—Scalability Enhancements*
A description of SNMP agent support in Cisco IOS for the MPLS Traffic Engineering MIB (MPLS TE MIB)	*MPLS Traffic Engineering (TE) MIB*
Basic MPLS VPN Carrier Supporting Carrier configuration tasks	*MPLS VPN Carrier Supporting Carrier*
Overview and configuration tasks for the Multiprotocol Label Switching (MPLS) distribution protocol.	"Multiprotocol Label Switching" chapter in the *Cisco IOS Switching Services Configuration Guide, Release 12.2*

Standards

Standards	Title
No new or modified standards are supported by this feature.	—

MIBs

MIBs[1]	MIBs Link
• PPVPN-MPLS-VPN MIB **Note** The structure and content of the PPVPN-MPLS-VPN MIB is in full conformance with the provisions of Section 10 of RFC 2026. The Internet Engineering Task Force (IETF) document entitled *draft-ietf-ppvpn-mpls-vpn-mib-03.txt* includes objects that support the MPLS VPN MIB agent.	To obtain lists of supported MIBs by platform and Cisco IOS release, and to download MIB modules, go to the Cisco MIB website on Cisco.com at the following URL: http://www.cisco.com/public/sw-center/netmgmt/cmtk/mibs.shtml

[1] Not all supported MIBs are listed.

To locate and download MIBs for selected platforms, Cisco IOS releases, and feature sets, use Cisco MIB Locator found at the following URL:

http://tools.cisco.com/ITDIT/MIBS/servlet/index

If Cisco MIB Locator does not support the MIB information that you need, you can also obtain a list of supported MIBs and download MIBs from the Cisco MIBs page at the following URL:

http://www.cisco.com/public/sw-center/netmgmt/cmtk/mibs.shtml

To access Cisco MIB Locator, you must have an account on Cisco.com. If you have forgotten or lost your account information, send a blank e-mail to cco-locksmith@cisco.com. An automatic check will verify that your e-mail address is registered with Cisco.com. If the check is successful, account details with a new random password will be e-mailed to you. Qualified users can establish an account on Cisco.com by following the directions found at this URL:

http://www.cisco.com/register

RFCs

RFCs[1]	**Title**
RFC 2233	*The Interfaces Group MIB using SMIv2*
RFC 2547bis	*BGP/MPLS VPNs*

[1] Not all supported RFCs are listed.

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, tools, and lots more. 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 modified commands. All other commands used with this feature are documented in the Cisco IOS Release 12.2 command reference publications.

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

snmp-server enable traps

To enable the router to send Simple Network Management Protocol (SNMP) traps and informs, use the **snmp-server enable traps** command in global configuration mode. To disable SNMP notifications, 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) Type of notification (trap or inform) to enable. If no type is specified, all notifications are sent. The notification type can be one of the following keywords:

- **bgp**—Enables Border Gateway Protocol (BGP) state change notifications.
- **config**—Enables configuration notifications.
- **entity**—Enables Entity MIB modification notifications.
- **envmon**—Enables Cisco enterprise-specific environmental monitor notifications when an environmental threshold is exceeded. When the **envmon** keyword is used, you can specify a notification-option value.
- **frame-relay**—Enables Frame Relay notifications.
- **hsrp**—Enables Hot Standby Router Protocol (HSRP) notifications.
- **isdn**—Enables Integrated Services Digital Network (ISDN) notifications. When the **isdn** keyword is used on Cisco 1600 series routers, you can specify a notification-option value.
- **mpls traffic-eng**—Sends notifications about changes in the status of MPLS traffic engineering tunnels.

Note The *notification-type* keyword for MPLS traffic engineering tunnels is specified as **mpls traffic-eng** (containing an intervening space and a dash). This particular keyword syntax (which is interpreted by the CLI as a two-word construct) has been adopted to maintain consistency with other MPLS traffic engineering commands.

- **mpls vpn**—Enables MPLS VPN-MIB notifications.

Note The *notification-type* keyword for MPLS VPNs is specified as **mpls vpn** (containing an intervening space). This particular keyword syntax (which is interpreted by the CLI as a two-word construct) has been adopted to maintain consistency with other MPLS VPN commands.

- **repeater**—Enables Ethernet hub repeater notifications. When the **repeater** keyword is selected, you can specify a notification-option value.
- **rsvp**—Enables Resource Reservation Protocol (RSVP) notifications.
- **rtr**—Enables response time reporter (RTR) notifications.

notification-type (continued)

- **snmp** [**authentication**]—Enables RFC 1157 SNMP notifications. Use of the **authentication** keyword produces the same effect as not using the **authentication** keyword. Both the **snmp-server enable traps snmp** and **snmp-server enable traps snmp authentication** forms of this command globally enable (or, if using the **no** form, disable) the following SNMP traps:
  - authentication Failure
  - linkUP
  - linkDown
  - warmstart
- **syslog**—Enables Cisco Syslog MIB error message notification. Specify the level of messages to send with the **logging history level** command.

notification-option

(Optional) Notification options.

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

  When the **envmon** keyword is used, you can enable a specific environmental notification type, or accept all notification types from the environmental monitor system. If no option is specified, all environmental notifications are enabled. The option can be one or more of the following:
  - **voltage**
  - **shutdown**
  - **supply**
  - **fan**
  - **temperature**
- **isdn** [**call-information** | **isdn u-interface**]

  When the **isdn** keyword is used, you can specify the **call-information** keyword to enable an SNMP ISDN call information notification for the ISDN MIB subsystem, or you can specify the **isdn u-interface** keyword to enable an SNMP ISDN U interface notification for the ISDN U interface MIB subsystem.
- **mpls traffic-eng** [**up** | **down** | **reroute**]

  When you specify the **mpls traffic-eng** keyword, it enables the sending of notifications to indicate changes in the status of MPLS traffic engineering tunnels.

  Any one of the following can be specified as an argument to the **mpls traffic-eng** keyword:
  - **up**
  - **down**
  - **reroute**

notification-option (continued)

If you do not specify a specific argument for the **mpls traffic-eng** keyword, all three types of MPLS traffic engineering tunnel notifications are sent.

Note The *notification-option* keyword for MPLS traffic engineering tunnels is specified as **mpls traffic-eng** (containing an intervening space and a dash). This particular keyword syntax (which is interpreted by the CLI as a two-word construct) has been adopted to maintain consistency with other MPLS traffic engineering commands.

- **mpls vpn** [**vrf-up** | **vrf-down** | **mid-threshold** | **max-threshold** | **illegal-label**]

  When you specify the **mpls vpn** keyword, it enables the sending of notifications to indicate the status of MPLS VPN forwarding and routing. You can specify one or more of the following keywords:

  - **vrf-up**—Enables a notification of the assignment of a VRF to an interface that is operational or the transition of a VRF interface to the operationally up state.
  - **vrf-down**—Enables a notification of the removal of a VRF from an interface or the transition of an interface to the down state.
  - **mid-threshold**—Enables a notification of a warning that the number of routes created has crossed a defined threshold. This warning is sent only at the time the threshold is exceeded.
  - **max-threshold**—Enables a notification that the route created exceeded the maximum defined threshold. Every route created that exceeds this threshold generates and sends a notification.
  - **illegal-label**—Enables a notification that the number of illegal labels received on a VRF interface exceeded the defined threshold. Labels are illegal if they are outside the legal range, do not have an LFIB entry, or do not match table IDs for the label.

  If no option is specified, all MPLS VPN-MIB notifications are enabled.

Note The *notification-type* keyword for MPLS VPN is specified as **mpls vpn** (containing an intervening space). This particular keyword syntax (which is interpreted by the CLI as a two-word construct) has been adopted to maintain consistency with other MPLS VPN commands.

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

  When you specify the **repeater** keyword, you can also specify the repeater option. If no option is specified, all repeater notifications are enabled. The specified option can be either of the following keywords:

  - **health**—Enables IETF Repeater Hub MIB (RFC 1516) health notification.
  - **reset**—Enables IETF Repeater Hub MIB (RFC 1516) reset notification.

Defaults

This command is disabled by default. Most notification types are disabled. However, some notification types cannot be controlled by means of this command.

If you enter this command with no *notification-type* keywords, the default is to enable all notification types controlled by 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 replaced the **snmp-server trap-authentication** command.
12.0(17)ST	This command was integrated into Cisco IOS Release 12.0(17)ST. The **mpls traffic-eng** keyword was added for use with the *notification-type* and *notification-option* parameters of the **snmp-server enable traps** command.
12.0(21)ST	The **mpls vpn** keyword was added 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(13)T	This command was integrated into Cisco IOS Release 12.2(13)T.
12.2(14)S	This command was integrated into Cisco IOS Release 12.2(14)S.

Usage Guidelines

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

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

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

To enable multiple types of notifications, you must issue a separate **snmp-server enable traps** command for each notification type and notification option.

The **snmp-server enable traps** command is used with the **snmp-server host** command. Use the **snmp-server host** command to specify the NMS host (or hosts) to receive 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, using the community string 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 using the community string 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, using the community string 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 using the community string public:

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

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

In the following example, MPLS VPN trap notifications are sent to the host specified as 172.31.156.34 using the community string public, if a VRF transitions from a down state to an up state or from an up state to a down state:

```
Router(config)# snmp-server enable traps mpls vpn vrf-up vrf-down

Router(config)# snmp-server host 172.31.156.34 traps public mpls-vpn
```

Related Commands

Command	Description
snmp-server host	Specifies the recipient of SNMP notifications.

snmp-server host

To specify the recipient of a Simple Network Management Protocol (SNMP) notification, use the **snmp-server host** command in global configuration mode. To remove the specified host, use the **no** form of this command.

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

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

Syntax Description

host-addr	Name or Internet address of the host (the targeted recipient of SNMP notifications).
traps	(Optional) Sends SNMP notifications (traps) to this host. This is the default if the [**traps** | **informs**] keyword choice is not specified.
informs	(Optional) Sends SNMP informs to this host.
version	(Optional) Version of the SNMP used to send the notifications. Version 3 is the most secure model, because it allows packet encryption by means of the **priv** keyword. If you use the **version** keyword, you must specify one of the following: • **1**—SNMPv1. This option is not available with informs. • **2c**—SNMPv2C. • **3**—SNMPv3. The following optional keywords can be used in conjunction with the version 3 keyword: – **auth** (Optional) Enables Message Digest 5 (MD5) and Secure Hash Algorithm (SHA) packet authentication. – **noauth** (Default) The noAuthNoPriv security level. This is the default if the [**auth** | **noauth** | **priv**] keyword choice is not specified. – **priv** (Optional) Enables Data Encryption Standard (DES) packet encryption (also called “privacy”).
community-string	Password-like community string sent with the notification 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 prior to using the **snmp-server host** command.
udp-port *port*	(Optional) UDP port of the host to which SNMP notifications are to be sent. The default is 162.
notification-type	(Optional) Specifies the type of SNMP notification to be sent to the host. If no type is specified, all notifications are sent. Any one or more of the following can be specified as keywords in the *notification-type* parameter: • **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 an environmental threshold is exceeded.

notification-type (continued)

- **frame-relay**—Sends Frame Relay notifications.
- **hsrp**—Sends Hot Standby Router Protocol (HSRP) notifications.
- **isdn**—Sends Integrated Services Digital Network (ISDN) notifications.
- **llc2**—Sends Logical Link Control, Type 2 (LLC2) notifications.
- **mpls-traffic-eng**—Sends MPLS traffic engineering notifications indicating changes in the status of MPLS traffic engineering tunnels.

Note The *notification-type* keyword applicable to MPLS traffic engineering tunnels is specified as **mpls-traffic-eng** (containing two dashes and no intervening spaces). This syntax is necessary to ensure that the CLI interprets this parameter as a unified, single-word construct, thus preserving the capability of the **snmp-server host** command to accept multiple *notification-type* keywords in the CLI command line (subject only to the requirement that all such keywords specified be separated by a space).

The corresponding parameter in the **snmp-server enable traps** command, however, is not subject to this requirement and is, therefore, specified as **mpls traffic-eng** (containing an intervening space and a dash). The keyword syntax for the *notification-type* and *notification-option* parameters in the case of the **snmp-server enable traps** command is interpreted by the CLI as a two-word construct and must be so specified in order to maintain consistency with other MPLS traffic engineering commands.

- **repeater**—Sends standard repeater (hub) notifications.
- **mpls-vpn**—Sends MPLS VPN-MIB trap notifications to the specified host that indicate whether the VPN is up or down, whether the number of VPN routes has exceeded a warning or maximum defined threshold, and whether the number of label errors has exceeded a defined threshold.

Note The *notification-type* keyword applicable to MPLS VPNs is specified as **mpls-vpn** (containing a dash and no intervening spaces). This syntax is necessary to ensure that the CLI interprets this parameter as a unified, single-word construct, thus preserving the capability of the **snmp-server host** command to accept multiple *notification-type* keywords in the CLI command line (subject only to the requirement that all such keywords specified be separated by a space).

The corresponding parameter in the **snmp-server enable traps** command, however, is not subject to this requirement and is, therefore, specified as **mpls vpn** (containing an intervening space). The keyword syntax for the *notification-type* and *notification-option* parameters in the case of the **snmp-server enable traps** command is interpreted by the CLI as a two-word construct and must be so specified in order to maintain consistency with other MPLS VPN commands.

notification-type (continued)	• **rsrb**—Sends remote source-route bridging (RSRB) notifications. • **rsvp**—Sends Resource Reservation Protocol (RSVP) notifications. • **rtr**—Sends Service Assurance (SA) Agent RTR notifications. • **sdlc**—Sends Synchronous Data Link Control (SDLC) notifications. • **sdllc**—Sends SDLLC notifications. • **snmp**—Sends SNMP notifications (as defined in RFC 1157). • **stun**—Sends serial tunnel (STUN) notifications. • **syslog**—Sends error message notifications (Cisco Syslog MIB). Specify the level of messages to be sent using the **logging history level** command. • **tty**—Sends Cisco enterprise-specific notifications when a Transmission Control Protocol (TCP) connection closes. • **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 with no keywords, the default is to send all types of notifications (traps) to the host. No informs will be sent to this host.

If no **version** keyword is present, the default is version 1. The **no snmp-server host** command with no keywords will disable notifications, but not informs, to the host. To disable informs, use the **no snmp-server host informs** command.

Note If the *community-string* is not defined using the **snmp-server community** command prior to using this command, the default form of the **snmp-server community** command will automatically be inserted into the configuration. The password (*community-string*) used for this automatic configuration of the **snmp-server community** will be the same as 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	This command was integrated into Cisco IOS Release 12.0(17)ST. The **mpls-traffic-eng** keyword was added for use with the *notification-type* parameter of the **snmp-server host** command to enable sending SNMP notifications reflecting status changes in MPLS traffic engineering tunnels.
12.0(21)ST	The **mpls-vpn** keyword was added for use with the *notification-type* parameter of the **snmp-server host** command to enable sending SNMP notifications reflecting changes in MPLS VPNs status.
12.0(22)S	This command was integrated into Cisco IOS Release 12.0(22)S.

Release	Modification
12.2(13)T	This command was integrated into Cisco IOS Release 12.2(13)T.
12.2(14)S	This command was integrated into Cisco IOS Release 12.2(14)S.

Usage Guidelines

To configure a label switch router (LSR) to send SNMP notifications, 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 network management system (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 enable multiple NMS hosts, you must issue a separate **snmp-server host** command for each targeted NMS. 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 enter an **snmp-server host inform** command for an NMS host, and then enter another **snmp-server host inform** command for the same NMS host, the second command overrides the first.

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

Examples

If you want to configure a unique SNMP community string for notifications, but you want to prevent SNMP polling access with this 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 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 Cisco environmental monitor enterprise-specific notifications are sent to the 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 router is enabled to send all notifications to the host identified as myhost.cisco.com using the community string public:

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

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

In the following example, notifications will not be sent to any host. The BGP notifications are enabled for all hosts, but only the ISDN notifications are enabled for sending to a host.

```
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 using the community string 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
```

Related Commands

Command	Description
snmp-server enable traps	Enables the sending of SNMP notifications.

Glossary

AS—autonomous system. A collection of networks that share the same routing protocol and that are under the same system administration.

ASN.1—Abstract Syntax Notation One. OSI language for describing data types independent of particular computer structures and representation techniques. Described by ISO International Standard 8824.

BGP—Border Gateway Protocol. The exterior Border Gateway Protocol used to exchange routing information between routers in separate autonomous systems. BGP uses Transmission Control Protocol (TCP). Because TCP is a reliable protocol, BGP does not experience problems with dropped or fragmented data packets.

BGP prefixes—A route announcement using the BGP. A prefix is composed of a path of autonomous system (AS) numbers, indicating which networks the packet must pass through, and the IP block that is being routed. A BGP prefix would look something like: 701 1239 42 206.24.14.0/24. (The /24 part is referred to as a CIDR mask.) The /24 indicates that there are 24 ones in the netmask for this block starting from the left side. A /24 corresponds to the natural mask 255.255.255.0.

CEF—Cisco Express Forwarding. An advanced Layer 3 IP switching technology. CEF optimizes network performance and scalability for networks with large and dynamic traffic patterns.

CE router—customer edge router. A router on the border between a VPN provider and a VPN customer that belongs to the customer.

CIDR—classless interdomain routing. Technique supported by BGP4 and based on route aggregation. CIDR allows routers to group routes together to reduce the quantity of routing information carried by the core routers. With CIDR, several IP networks appear to networks outside the group as a single, larger entity. With CIDR, IP addresses and their subnet masks are written as four octets, separated by periods, followed by a forward slash and a two-digit number that represents the subnet mask.

community—In SNMP, a logical group of managed devices and NMSs in the same administrative domain.

community name—*See* community string.

community string—Text string that acts as a password and is used to authenticate messages sent between a managed station and a router containing an SNMP agent. The community string is sent in every packet between the manager and the client. Also called a community name.

IETF—Internet Engineering Task Force. Task force consisting of over 80 working groups responsible for developing Internet standards. The IETF operates under the auspices of ISOC. *See also* ISOC.

informs—A type of notification message that is more reliable than a conventional trap notification message, because the informs message notification requires acknowledgment, and a trap notification does not.

ISOC—Internet Society. International nonprofit organization, founded in 1992, that coordinates the evolution and use of the Internet. In addition, ISOC delegates authority to other groups related to the Internet, such as the IAB. ISOC is headquartered in Reston, Virginia (United States).

label—A short, fixed-length data construct that tells switching nodes how to forward data (packets or cells).

Label Distribution Protocol—*See* LDP.

Label Forwarding Information Base—*See* LFIB.

label switching router—*See* LSR.

LDP—Label Distribution Protocol. A standard protocol between MPLS-enabled routers that is used for the negotiation of the labels (addresses) used to forward packets.

LFIB—Label Forwarding Information Base. In the Cisco Label Switching system, the data structure for storing information about incoming and outgoing tags (labels) and associated equivalent packets suitable for labeling.

LSR—label switching router. A device that forwards MPLS packets based on the value of a fixed-length label encapsulated in each packet.

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

MPLS—Multiprotocol Label Switching. A method for forwarding packets (frames) through a network. It enables routers at the edge of a network to apply labels to packets (frames). ATM switches or existing routers in the network core can switch packets according to the labels with minimal lookup overhead.

MPLS interface—An interface on which MPLS traffic is enabled.

MPLS VPN—Multiprotocol Label Switching Virtual Private Network. Using MPLS VPNs in a Cisco IOS network provides the capability to deploy and administer scalable Layer 3 VPN backbone services including applications, data hosting network commerce, and telephony services to business customers. A VPN is a secure IP-based network that shares resources on one or more physical networks. A VPN contains geographically dispersed sites that can communicate securely over a shared backbone.

For an MPLS VPN Solution, an MPLS VPN is a set of provider edge routers that are connected by means of a common “backbone” network to supply private IP interconnectivity between two or more customer sites for a given customer. Each VPN has a set of provisioning templates and policies and can span multiple provider administrative domains (PADs).

Multiprotocol Label Switching—*See* MPLS.

notification—A message sent by an SNMP agent to a network management station, console, or terminal to indicate that a significant event within Cisco IOS has occurred. *See also* trap.

NMS—network management system. A powerful, well-equipped computer (typically an engineering workstation) that is used by a network administrator to communicate with other devices in the network. An NMS is typically used to manage network resources, gather statistics, and perform a variety of network administration and configuration tasks.

PE router—provider edge router. A router on the border between a VPN provider and a VPN customer that belongs to the provider.

PPVPN—Provider-Provisioned VPN. The name of the IETF working group that is developing the PPVPN-MPLS-VPN MIB.

QoS—quality of service. Measure of performance for a transmission system that reflects its transmission quality and service availability.

route target—*See* RT.

RSVP—Resource Reservation Protocol. Protocol for reserving network resources to provide Quality of Service guarantees to application flows.

RT—route target. An extended community attribute that identifies a group of routers and, in each router of that group, a subset of forwarding tables maintained by the router that can be populated with a BGP route carrying that extended community attribute. The RT is a 64-bit value by which Cisco IOS discriminates routes for route updates in VRFs.

Simple Network Management Protocol—*See* SNMP.

SNMP—Simple Network Management Protocol. Network management protocol used almost exclusively in TCP/IP networks. SNMP provides a means to monitor and control network devices, and to manage configurations, statistics collection, performance, and security. See *also* SNMP2.

SNMP2—SNMP Version 2. Version 2 of the popular network management protocol. SNMP2 supports centralized as well as distributed network management strategies, and includes improvements in the SMI, protocol operations, management architecture, and security. *See also* SNMP.

traffic engineering—The techniques and processes used to cause routed traffic to travel through the network on a path other than the one that would have been chosen if standard routing methods had been used.

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

VPN—Virtual Private Network. A group of sites that, as the result of a set of administrative policies, are able to communicate with each other over a shared backbone network. *See* MPLS VPN.

VPN ID—A mechanism that identifies a VPN based on RFC 2685. A VPN ID consists of an Organizational Unique Identifier (OUI), a three-octet hex number assigned by the IEEE Registration Authority, and a VPN index, a four-octet hex number, which identifies the VPN within the company.

VRF—VPN routing/forwarding instance. A VRF consists of an IP routing table, a derived forwarding table, a set of interfaces that use the forwarding table, and a set of rules and routing protocols that determine what goes into the forwarding table. In general, a VRF includes the routing information that defines a customer VPN site that is attached to a PE router.

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