Cisco Prime Fulfillment User Guide 6.2

Americas Headquarters
Cisco Systems, Inc.
170 West Tasman Drive
San Jose, CA 95134-1706
USA
http://www.cisco.com
Tel: 408 526-4000
800 553-NETS (6387)
Fax: 408 527-0883

Text Part Number: OL-25458-01

Cisco Prime Fulfillment User Guide 6.2

CONTENTS

About This Guide

This preface contains the following sections:

Objective

The *Cisco Prime Fulfillment User Guide 6.2* contains detailed explanations of Prime Fulfillment services and components across all applications.

Note With this release, Prime Fulfillment can be used as a standalone product or as part of the Cisco Prime for IP Next Generation Network (IP NGN) Suite. When installed as part of the suite, you can launch Prime Fulfillment from the Prime Central portal. For more information about Prime Central, see the documentation for Cisco Prime Central 1.0.

Audience

This guide is designed for service provider network managers and operators who are responsible for provisioning Prime Fulfillment services for their customers.

Network managers and operators should be familiar with the following topics, as required for the services being configured:

- Basic concepts and terminology used in internetworking.
- Network topologies and protocols.
- Layer 2 Virtual Private Network (L2VPN), Virtual Private LAN Service (VPLS), VPN, Multiprotocol Label Switching (MPLS), and terms and technology.
- MPLS VPN terms and technology.
- A general understanding of Multiprotocol Label Switching Traffic Engineering (MPLS TE) concepts and traffic engineering is also required.

Organization

This guide is organized as follows:

- Chapter 1, "Prime Fulfillment GUI Overview," describes how to get started with the Prime Fulfillment graphical user interface (GUI).
- Chapter 2, "Before Setting Up Prime Fulfillment," describes how to set up the Cisco Prime Fulfillment services
- Chapter 3, "Managing L2VPN and Carrier Ethernet Services," describes how to manage L2VPN and carrier ethernet services.
- Chapter 4, "Managing RAN Backhaul Services," describes how to manage RAN backhaul service.
- Chapter 5, "Managing MPLS VPN Services," describes how to manage MPLS VPN services.
- Chapter 6, "Managing MPLS Transport Profile Services," describes how to manage MPLS Transport Profile services.
- Chapter 7, "Managing MPLS Traffic Engineering Services," describes how to manage MPLS traffic engineering services.
- Chapter 8, "Managing Service Requests," describes how to manage service requests.
- Chapter 9, "Managing Templates and Data Files," describes how to manage template and data files.
- Chapter 10, "Monitoring," describes how to monitor Prime Fulfillment.
- Chapter 11, "Performing Diagnostics," describes performing diagnostic in Prime Fulfillment.
- Chapter 12, "Using the Topology Tool," describes using the topology tool in Prime Fulfillment.
- Chapter 13, "Using Inventory Manager,", describes how to use the inventory manager in Prime Fulfillment.
- Chapter 14, "Administration Tasks," describes the administrative tasks that can be performed in Prime Fulfillment.
- Appendices provide supplementary information.

Related Documentation

The entire documentation set for Cisco Prime Fulfillment, can be accessed at:

http://www.cisco.com/en/US/products/ps12199/tsd_products_support_series_home.html

or at:

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

The following documents comprise the Cisco Prime Fulfillment 6.2 documentation set:

General Documentation (in suggested reading order)

- *Cisco Prime Fulfillment Getting Started and Documentation Guide 6.2*

  http://www.cisco.com/en/US/docs/net_mgmt/prime/fulfillment/6.2/roadmap/docguide.html
- *Release Notes for Cisco Prime Fulfillment 6.2*

  http://www.cisco.com/en/US/docs/net_mgmt/prime/fulfillment/6.2/release/notes/relnotes.html

- *Cisco Prime Fulfillment Installation Guide 6.2*

  http://www.cisco.com/en/US/docs/net_mgmt/prime/fulfillment/6.2/installation/guide/installation.html

- *Cisco Prime Fulfillment Supported Devices 6.2*

  http://www.cisco.com/en/US/docs/net_mgmt/prime/fulfillment/6.2/supported/devices/supported_devices_table.xls

- *Cisco Prime Fulfillment User Guide 6.2*

  http://www.cisco.com/en/US/docs/net_mgmt/prime/fulfillment/6.2/user/guide/prime_fulfill.html

- *Cisco Prime Fulfillment Theory of Operations Guide 6.2*

  http://www.cisco.com/en/US/docs/net_mgmt/prime/fulfillment/6.2/theory/operations/guide/theory.html

- *Cisco Prime Fulfillment Third Party and Open Source Copyrights 6.2*

  http://www.cisco.com/en/US/docs/net_mgmt/prime/fulfillment/6.2/third_party/open_source/copyright/Prime_Fulfillment_Third_Party_and_Open_Source_Copyrights62.pdf

API Documentation

- *Cisco Prime Fulfillment API Programmer Guide 6.2*

  http://www.cisco.com/en/US/docs/net_mgmt/prime/fulfillment/6.2/developer/guide/apipg.html

- *Cisco Prime Fulfillment API Programmer Reference 6.2*

  http://www.cisco.com/en/US/docs/net_mgmt/prime/fulfillment/6.2/developer/reference/xmlapi.zip

Note All documentation *might* be upgraded over time. All upgraded documentation will be available at the same URLs specified in this document.

Other Cisco Prime Product Documentation

See also the documentation for the following Cisco Prime products:

- *Cisco Prime Central 1.0*

  http://www.cisco.com/en/US/products/ps11754/tsd_products_support_series_home.html

- *Cisco Prime Network 3.8*

  http://www.cisco.com/en/US/products/ps11879/tsd_products_support_series_home.html

- *Cisco Prime Optical 9.3.1*

  http://www.cisco.com/en/US/products/ps11670/tsd_products_support_series_home.html

- *Cisco Prime Performance Manager 1.0*

  http://www.cisco.com/en/US/products/ps11715/tsd_products_support_series_home.html

Obtaining Documentation and Submitting a Service Request

For information on obtaining documentation, submitting a service request, and gathering additional information, see the monthly *What's New in Cisco Product Documentation*, which also lists all new and revised Cisco technical documentation, at:

http://www.cisco.com/en/US/docs/general/whatsnew/whatsnew.html

Subscribe to the *What's New in Cisco Product Documentation* as a Really Simple Syndication (RSS) feed and set content to be delivered directly to your desktop using a reader application. The RSS feeds are a free service and Cisco currently supports RSS Version 2.0.

CHAPTER 1

Prime Fulfillment GUI Overview

This chapter provides information about how to get started to use Cisco Prime Fulfillment and gives a structural overview of this guide. It contains the following sections:

System Recommendations

The system recommendations and requirements are listed in Chapter 1, “System Recommendations” of the *Cisco Prime Fulfillment Installation Guide 6.2* and the *Release Notes for Cisco Prime Fulfillment 6.2*. The recommendation is to thoroughly review this list before even planning your installation, to be sure that you have all the hardware and software you must successfully install.

Introduction

Cisco Prime Fulfillment 6.2 is an evolution of Cisco IP Solution Center (ISC) that includes the powerful capabilities of that offering combined with significant enhancements to the user interface, to adding and updating devices and technologies, and to extending the powerful diagnostic workflows. The changes in Prime Fulfillment are listed in the *Release Notes for Cisco Prime Fulfillment 6.2*.

This guide lists many features that are common among multiple applications, which are sold and licensed separately. The applications and their respective *User Guides* reference this document for setup steps necessary before creating a policy and then a service request specific to the application and for other common features.

Before explaining the tabs in the Graphical User Interface (GUI), see the “Structural Overview” section on page 1-2. It explains elements common to many windows in Prime Fulfillment.

The GUI is separated into the following large sections (tabs):

- "Operate" section on page 1-8
- "Inventory" section on page 1-9
- "Service Design" section on page 1-10
- "Traffic Engineering" section on page 1-10
- "Diagnostics" section on page 1-11
- "Administration" section on page 1-11

The remaining sections in this chapter explain the sections and subsections of this guide that explain the functionality available from these tabs.

Note The terminology used in this guide and this product can be used interchangeably or preferably with other terms.

Structural Overview

After you log into Cisco Prime Fulfillment, the first window to appear is the Home window, as shown in Figure 1-1, "Home Window."

Figure 1-1 Home Window

Note The tabs and the choices navigating within the tabs that appear depend on the user permission, explained in Chapter 14, "Manage Security"(**Administration > Security > User Roles**). The choices shown in this guide are for all permissions (**admin**).

There are two new charts available in the home screen, which provides a count of SR's in different states and list the SR's deployed for the past seven days:

- **Pie chart**—The pie chart provides an overall view of Service Requests in Prime Fulfillment with various states. If you click on any state in the pie chart it would redirect to the service manager screen with a list of all Service Requests on the selected state.
- **Bar chart**—The bar chart displays the last seven days Service requests added, modified, or deleted in Prime Fulfillment. If you click of the Bar, it would redirect to the service manager screen with a list of all Service Requests on the selected day.

This overview includes the following sections:

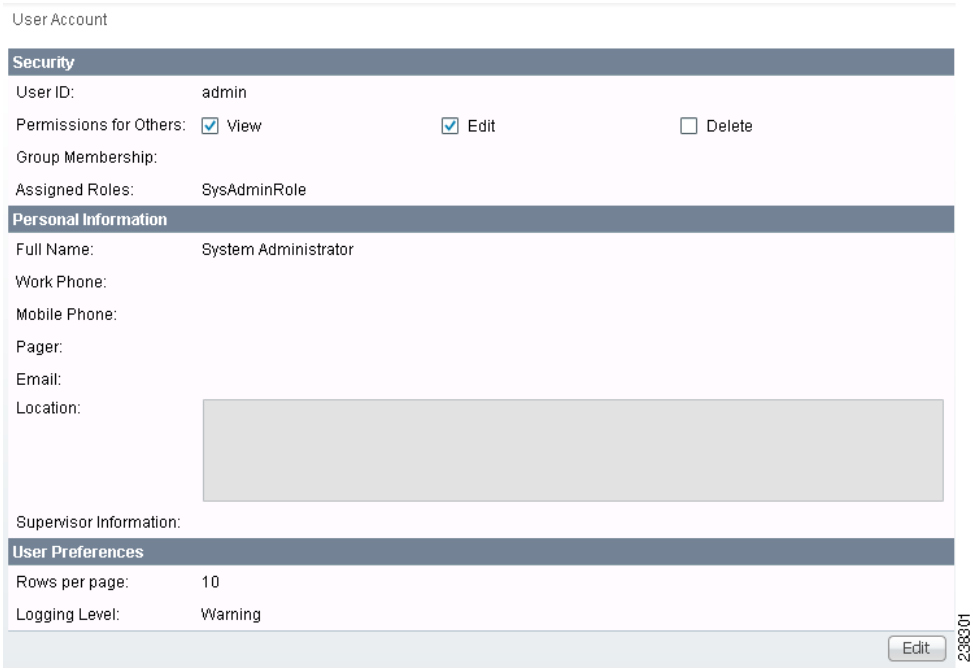
Links

In the upper right-hand corner of the Home window (Figure 1-1), additional links appear that function as follows:

User

The **User** in the Home page is **User:** followed by **admin** (default) or a username. When you click User: admin the following window appears:

Figure 1-2 User: admin window



You can change your password without the SysAdmin or UserAdmin privileges when you click the Edit button. This allows you to edit the user profile, including changing the password.

Customer

The **Customer** in the Home page is **Customer:** followed by **None** (default) or a customer name. This is referred to as Customer Context. The advantage of Customer Context is to focus only on information for a specified customer. To set the Customer Context, follow these steps:

Step 1 Click on the name after **Customer: None** and the following window appears.

Figure 1-3 Customer Context

Step 2 Click the **Select** button and you receive a list of all the currently created customers.

Step 3 Click the radio button for the customer for which you want information and click **Select**.

Figure 1-3, reappears with the name of the selected customer. Click **Save** or highlight the customer name and click **Clear** to reset the customer for which you want information.

The customer you chose now appears after **Customer:** on the Home window and it is the only customer for which information appears.

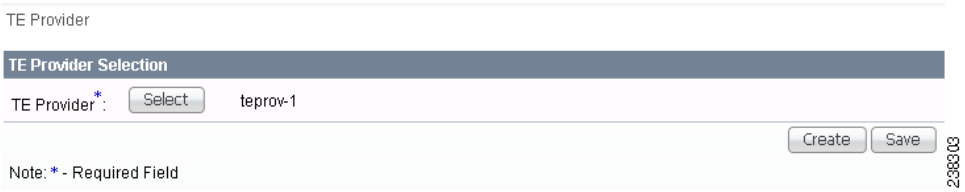
Step 4 You can reset the Customer Context by clearing and reselecting.

TE Provider

The **TE Provider** in the Home page is **TE Provider:** followed by **None** (default) or a TE provider name. This is referred to as TE Provider Context. The advantage of TE Provider Context is to focus only on information for a specified provider. To set the Provider Context, follow these steps:

Step 1 Click on the name after **TE Provider: None** and the following window appears.

Figure 1-4 TE Provider Context



Step 2 Click the **Select** button and you receive a list of all the currently created provider.

Step 3 Click the radio button for the customer for which you want information and click **Select**.

Figure 1-4, reappears with the name of the selected TE provider. Click **Save** or highlight the TE provider name and click **Clear** to reset the TE provider for which you want information.

The TE provider you chose now appears after **TE Provider:** on the Home window and it is the only TE Provider for which information appears.

Step 4 You can reset the TE Provider Context by clearing and reselecting.

Logout

When you click **Logout**, you log out of the product.

Feedback

When you click **Feedback**, you receive the feedback window to provide you feedback on this product.

About

When you click **About**, you receive the product name and version.

Help

When you click **Help**, you receive a pointer to the Prime Fulfillment documentation:

http://www.cisco.com/en/US/products/ps12199/tsd_products_support_series_home.html

From that location, you can choose the type of Prime Fulfillment document you want to see.

Common GUI Components

GUI components that are common on many windows are as follows:

Filters

At the top of many windows you can filter information that appears in the window. As shown in Figure 1-5, you can click the drop-down list for categories, then in the **matching** field enter the search criteria, using * if you want to indicate anything is a match (you can enter only * or you can place * before other characters, in the middle of other characters, at the end of other characters, or in multiple locations), and click **Find**. In some cases you might also have a field after the **matching** field from which you can select or enter more specifics for your **Find**.

Header Row Check Box

Many windows have a check box in the header row, where the column names exist, as shown in Figure 1-5. If you check this check box, then all check boxes in the window are chosen.

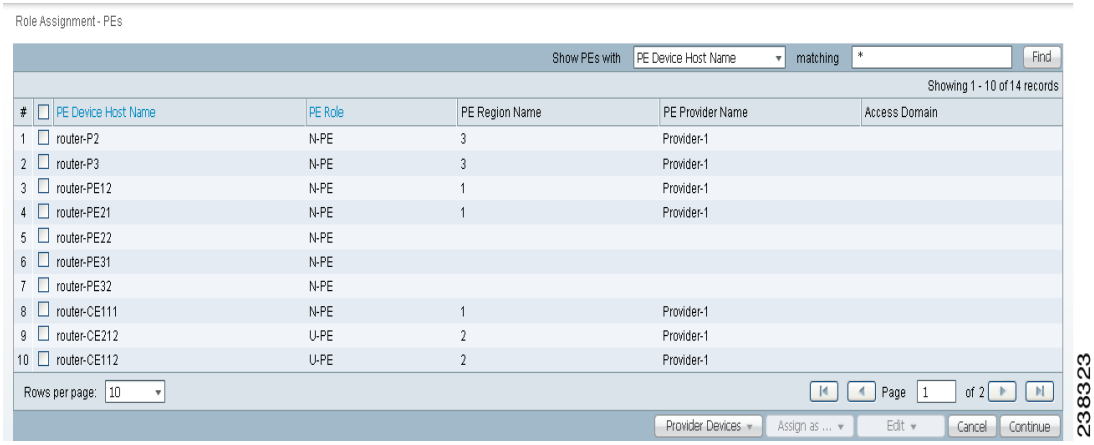
Rows per Page

In the bottom left corner of many windows, as shown in Figure 1-5, you can change the number of rows shown on this window in **Rows per page**. Click the drop-down list and you can select **5**, **10**, **20**, **30**, **40**, **50**, **100**, **500**, **1000**, or **2500**.

Go To Page

Near the bottom in the right corner of many windows, as shown in Figure 1-5, there is **Go to page** *field* **of** ***y***. In the *field*, you can enter the page you want to choose and then click the **Go** button to get there. The ***y*** indicates the last page for this topic. Another way to choose a specific page is to use the arrows. You can click the **>** arrow to choose the next page or the furthest arrow to the right **>|** to choose the last page. You can click the **<** arrow to choose the previous page or the furthest arrow to the left **|<** to choose the first page.

Figure 1-5 Example of Filtering, Header Row Check Box, Rows per Page, and Changing Pages



Auto Refresh

At the bottom left corner of several windows, there is a check box used to enable or disable the **Auto Refresh** feature, as shown in Figure 1-6. Checking this check box causes the window and its data to refresh every **n** milliseconds. The amount of time between refresh cycles can be set in the DCPL property: GUI.srRefreshRate. By default, the **Auto Refresh** feature is enabled to 30000 milliseconds.

Color Coding

In the Service Request table, the Task table, and the Device table, the colors you see indicate the state of the items, as shown in Figure 1-6.

In the **Service Request** table, the states have the following colors:

- BROKEN is bright yellow
- CLOSED is no color
- DEPLOYED is bright green
- FAILED AUDIT is bright yellow
- FAILED DEPLOY is bright red
- FUNCTIONAL is bright green
- INVALID is bright red
- LOST is bright yellow
- PENDING is bright green
- IN-PROGRESS is bright yellow
- REQUESTED is cream
- WAIT DEPLOYED is cream

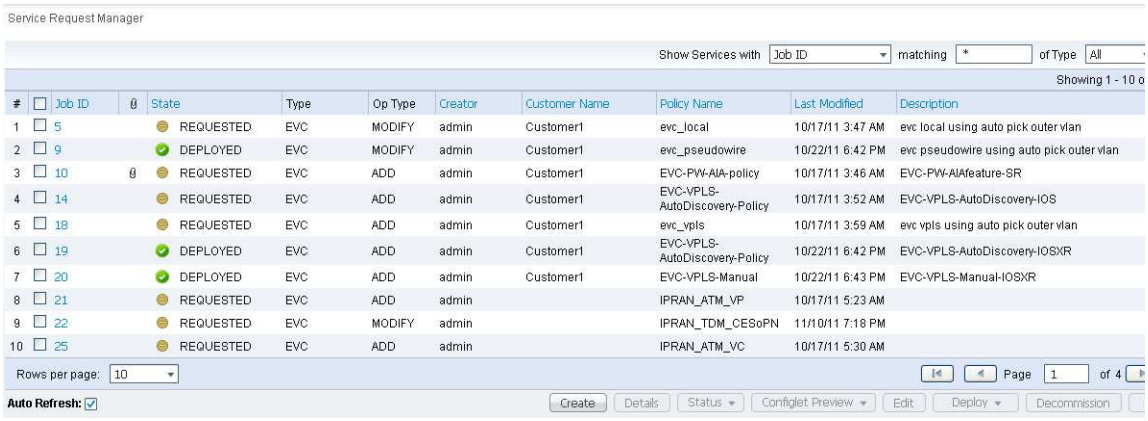
In the **Task** table, the states have the following colors:

- ABORTED is orange
- RUNNING is bright green
- WAITING_TO_RUN is cream
- errors is bright red
- successfully is bright green
- warnings is cyan

In the **devices** table, the states have the following colors:

- device returns anything other than **success** or **no result**, then the color is bright red
- device returns **success**, then the color is bright green
- **no result** from device, then the color is dark blue

Figure 1-6 Colors as Identifiers

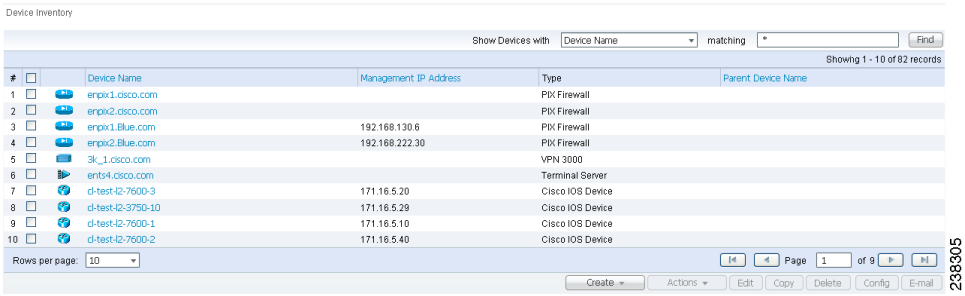


Icons

In some windows with tables of information, icons appear to show the type of device, as shown in Figure 1-7.

Note A list of possible icons can be found in Table 12-1 in the Launching Topology Tool, page 12-2 section of Chapter 12, "Using the Topology Tool."

Figure 1-7 Devices—Icons



Operate

Operate contains tools to create and manage Service Requests and the various tasks of Prime Fulfillment.

From the Home window you receive upon logging in, click the **Operate** tab and you receive a window as shown in Figure 1-8.

Figure 1-8 Operate Selections

The selections are as follows:

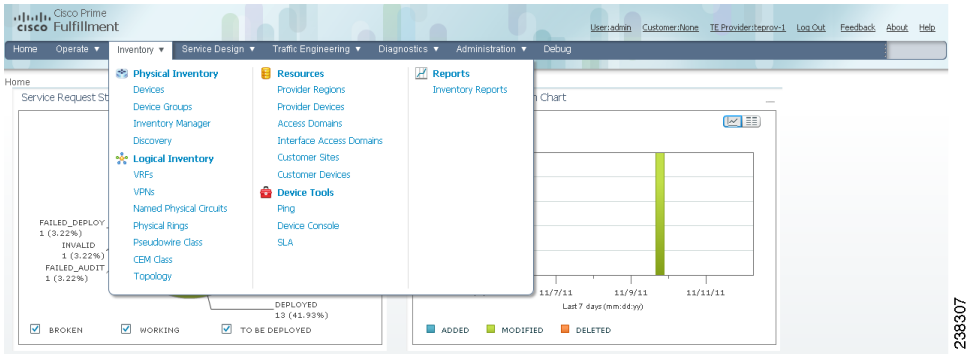
- **Service Requests**—Create, deploy, and manage service requests (SRs). This is explained in detail in Chapter 8, "Managing Service Requests".
- **Tasks**—Create and manage the tasks associated with Prime Fulfillment. This is explained in detail in Task Manager, page 10-23 section of Chapter 10, "Monitoring".

Inventory

Inventory contains tools to manage physical and logical inventory elements, resources, device tools, and reports.

From the Home window you receive upon logging in, click the **Inventory** tab and you receive a window as shown in Figure 1-9.

Figure 1-9 Inventory Selections



The selections are as follows:

- **Physical Inventory**—Create and manage Devices, Device Groups, Inventory Manager, and Discovery.
  - **Devices**—Create and manage devices (explained in detail in Devices, page 2-1 section of Chapter 2, "Before Setting Up Prime Fulfillment").
  - **Device Groups**—Create and manage device groups (explained in detail in Device Groups, page 2-28 section of Chapter 2, "Before Setting Up Prime Fulfillment").
  - **Inventory Manager**—Bulk-manage inventory elements (explained in detail in Chapter 13, "Using Inventory Manager").
  - **Discovery—**Discover devices, connections, and services (explained in detail in Appendix G, "Inventory - Discovery").
- **Logical Inventory**—Create and manage VRFs, VPNs, Named Physical Circuits, Physical Rings, and Pseudowire Class. This is explained in detail in Setting Up Logical Inventory, page 2-53 section of Chapter 2, "Before Setting Up Prime Fulfillment".
- **Resources**—Create and manage Customer Sites and Devices, Provider Regions and Devices, and Access Domains. This is explained in detail in Setting Up Resources, page 2-40 section of Chapter 2, "Before Setting Up Prime Fulfillment":
- **Device Tools**—Contains the following choices:
  - **Ping**—Perform Ping connectivity tests (explained in detail in Ping, page 10-1 section of Chapter 10, "Monitoring").
  - **SLA**—Manage Service Level Agreement (SLA) probes (explained in detail in SLA, page 10-3 section of Chapter 10, "Monitoring").
  - **Device Console**—Download commands and configlets to devices and view device configuration (explained in detail in Inventory - Device Console, page 13-1 section of Chapter 13, "Using Inventory Manager").

- **Reports**—Create and manage various reports of Prime Fulfillment. This is explained in Reports, page 10-27 section of Chapter 10, "Monitoring".

Service Design

Service Design contains management tools for creating and managing resources, policies, and templates.

From the Home window you receive upon logging in, click the **Service Design** tab and you receive a window as shown in Figure 1-10.

Figure 1-10 Service Design Selections

The selections are as follows:

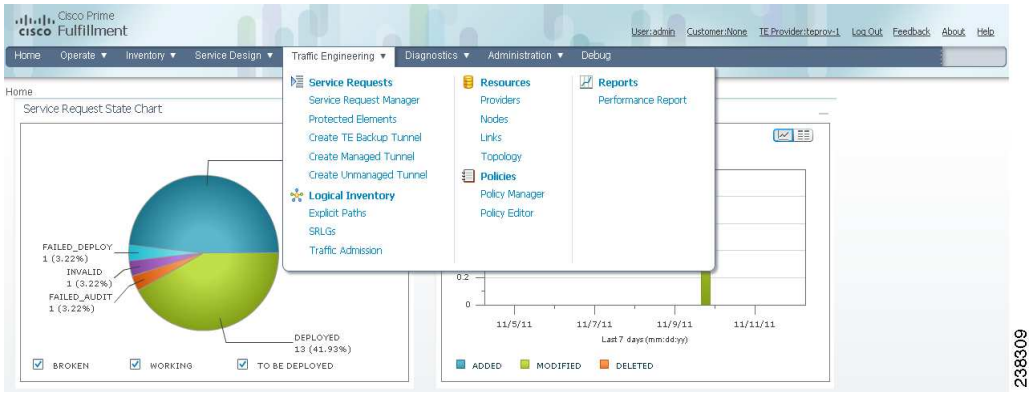
- **Resources**—Create and manage Customers, Providers, Resource Pools, and Route Targets. The following choices are explained in detail in Setting Up Resources, page 2-40 section of Chapter 2, "Before Setting Up Prime Fulfillment":
  - **Customers**—Create and manage customers.
  - **Providers**—Create and manage Providers.
  - **Resource Pools**—Create and manage pools for IP address, multicast address, route distinguisher, route target, site of origin, VC ID, and VLAN.
  - **CE Routing Communities**—Create and manage CE Routing Communities.
- **Policies**—Create and manage policies for licensed services.
- **Templates**—Create and manage templates and associated data (explained in detail in Chapter 9, "Managing Templates and Data Files").

Traffic Engineering

Traffic Engineering contains tools to create, deploy, and manage elements of Traffic Engineering Management. This is explained in detail in Chapter 7, "Managing MPLS Traffic Engineering Services."

From the Home window you receive upon logging in, click the **Traffic Engineering** tab and you receive a window as shown in Figure 1-11.

Figure 1-11 Traffic Engineering Selections

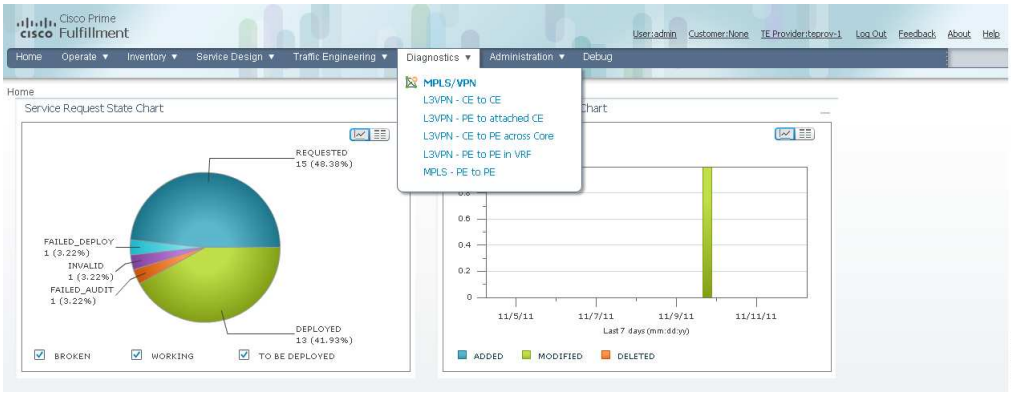


Diagnostics

Diagnostics contains automated troubleshooting and diagnostics for MPLS VPNs. This is explained in detail in *Part 6, Managing MPLS VPN Services*, from Chapter 11, "Performing Diagnostics".

From the Home window you receive upon logging in, click the **Diagnostic** tab and you receive a window as shown in Figure 1-12.

Figure 1-12 Diagnostic Selections

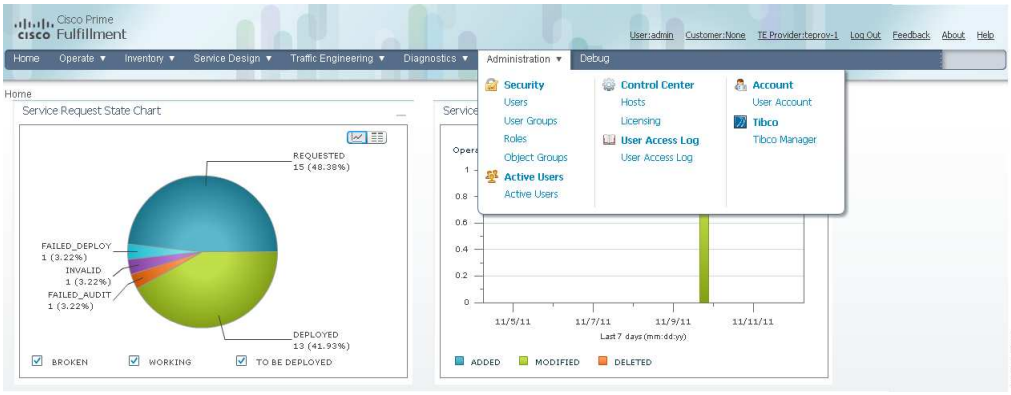


Administration

Administration contains tools to manage users, Prime Fulfillment configuration, servers, and licensing, to view users and the user access log, and to specify attributes for some messages.

From the Home window you receive upon logging in, click the **Administration** tab and you receive a window as shown in Figure 1-13.

Figure 1-13 Administration Selections



The selections are as follows:

- **Security**—Create and manage Users, User Groups, User Roles, and Object Groups. The following choices are explained in detail in Manage Security, page 14-9 section of Chapter 14, "Administration Tasks":
  - **Users**—Create and manage Users to also access Inventory Manager, Topology, and Northbound API.
  - **User Groups**—Create and manage User Groups. A Group is used to combine the privileges of all the roles contained within it.
  - **User Roles**—Create and manage User Roles, which define a set of permissions.
  - **Object Groups**—Create and manage a group of objects, such as devices, interfaces, and named physical circuits.
- **Control Center**—Manage Prime Fulfillment configuration, servers, and licensing. The following choices are explained in detail in Manage Control Center, page 14-2 section of Chapter 14, "Administration Tasks":
  - **Hosts**

Note If you want to do a **custom** install, this is only available through the Installation procedure explained in the *Cisco Prime Fulfillment Installation Guide 6.2*.

  - **Collection Zones**
  - **Licensing**
- **Active Users**—View users currently connected to Prime Fulfillment. Disconnect users (explained in detail in Manage Active Users and User Account, page 14-1 section of Chapter 14, "Administration Tasks").
- **User Access Log**—View the user access log (explained in detail in User Access Log, page 14-26 section of Chapter 14, "Administration Tasks").
- **Manage TIBCO Rendezvous**—Specify attributes for proper messaging among all Java™ Web Start distributed applications. This is explained in detail in Manage TIBCO Rendezvous, page 14-7 section of Chapter 14, "Administration Tasks".

CHAPTER 2

Before Setting Up Prime Fulfillment

This chapter explains how to set up the services. It contains the following sections:

Setting Up Devices and Device Groups

This section explains how to set up the physical services. It contains the following sections:

Devices

Every network element that Cisco Prime Fulfillment manages must be defined as a device in the system. An element is any device from which Prime Fulfillment can collect information. In most cases, devices are Cisco IOS routers that function as Provider Edge Routers (PEs) or Customer Edge Routers (CEs) in the MPLS VPN.

Note To provision services with Prime Fulfillment, you must have IPv4 connectivity.

This section describes how to configure SSH or SSHv2, set up SNMP, manually enable an RTR responder, and create, edit, delete, and configure various types of supported devices. This section includes the following topics:

Configuring SSH or SSHv2

Prime Fulfillment needs a mechanism to securely access and deploy configuration files on devices, which include routers and switches. And, to securely download a configlet and upload a configuration file from a device, Secure Shell (SSH) or SSH version 2(SSHv2) must be enabled.

The following sections describe:

Configuring SSH on Cisco IOS Routers Using a Domain Name

The procedure for configuring SSH on a Cisco IOS router is as follows:

	Command	Description
Step 1	`Router# configure terminal`	Enters global configuration mode.
Step 2	`Router(config)# ip domain-name <domain_name>`	Specifies the IP domain name.
Step 3	`Router(config)# username <username> password <password>`	Configures the user ID and password. Enter your Prime Fulfillment username and password. For example: `username admin password iscpwd`
Step 4	`Router(config)# crypto key generate rsa`	Generates keys for the SSH session.
Step 5	You will see the following prompt: `Choose the size of the key modulus in the range of 360 to 2048 for your general purpose keys. How many bits in the modulus (nnn):` Press **Enter** to accept the default number of bits.	Sets the number of bits.
Step 6	`Router(config)# line vty 0 4`	Enables SSH as part of the vty login transport.
Step 7	`Router(config-line)# login local`	The **login local** command indicates that the router stores the authentication information locally.
Step 8	`Router(config-line)# transport input telnet ssh`	Enables SSH transport.

	Command	Description
Step 9	`Router(config-line)# Ctrl+Z`	Returns to Privileged Exec mode.
Step 10	`Router# copy running startup`	Saves the configuration changes to nonvolatile random-access memory (NVRAM).

Configuring SSHv1 or SSHv2 on Cisco IOS Routers Using RSA Key Pairs

The procedure for configuring SSHv1 or SSHv2 on a Cisco IOS router is as follows.

	Command	Description
Step 1	`Router# enable`	Enables privileged EXEC mode. Enter your password, if prompted.
Step 2	`Router# configure terminal`	Enters global configuration mode.
Step 3	`Router(config)# ip ssh rsa keypair-name <keypair-name>`	Specifies which RSA keypair to use for SSH usage. Note: A Cisco IOS router can have many RSA key pairs.
Step 4	`Router(config)# crypto key generate rsa usage-keys label <key-label> modulus <modulus-size>`	Enables the SSH server for local and remote authentication on the router. For SSH Version 2, the modulus size must be at least 768 bits. Note: To delete the Rivest, Shamir, and Adelman (RSA) key-pair, use the **crypto key zeroize rsa** command. After you have deleted the RSA command, you automatically disable the SSH server.
Step 5	`Router(config)# ip ssh [timeout <seconds> | authentication-retries <integer>]`	Configures SSH control variables on your router.
Step 6	`Router(config)# ip ssh version [1 | 2]`	Specifies the version of SSH to be run on a router.

Configuring SSH or SSHv2 on Cisco IOS XR Routers

The procedure for configuring SSHv2 on a Cisco IOS XR router is as follows.

	Command	Description
Step 1	`RP/0/RP0/CPU0:router# configure`	Enters global configuration mode.
Step 2	`RP/0/RP0/CPU0:router(config)# hostname <hostname>`	Configures a hostname for your router.
Step 3	`RP/0/RP0/CPU0:router(config)# domain name <domain-name>`	`Defines a default domain name that the software uses to complete unqualified host names.`
Step 4	`RP/0/RP0/CPU0:router(config)# exit`	Exits global configuration mode, and returns the router to EXEC mode.
Step 5	`RP/0/RP0/CPU0:router(config)# crypto key generate rsa [usage keys | general-keys] [<keypair-label>]`	Generates an RSA key pair.

	Command	Description
Step 6	`RP/0/RP0/CPU0:router# crypto key generate dsa`	Enables the SSH server for local and remote authentication on the router. The recommended minimum modulus size is 1024 bits. Generates a DSA key pair. To delete the DSA key pair, use the **crypto key zeroize dsa** command. This command is used only for SSHv2.
Step 7	`RP/0/RP0/CPU0:router# configure`	Enters global configuration mode.
Step 8	`RP/0/RP0/CPU0:router# ssh timeout <seconds>`	(Optional) Configures the timeout value for user authentication to authentication, authorization, and accounting (AAA). If the user fails to authenticate itself to AAA within the configured time, the connection is aborted. If no value is configured, the default value of 30 is used for 30 seconds. The range is from 5 to 120.

Step 9	`RP/0/RP0/CPU0:router(config)# ssh server`
	or
	`RP/0/RP0/CPU0:router(config)# ssh server v2`
		Brings up an SSH server.
		To bring down an SSH server, use the **no ssh server** command.
		(Optional) Forces the SSH server to accept only SSHv2 clients if you configure the SSHv2 option by using the **ssh server v2** command. If you choose the **ssh server v2** command, only the SSH v2 client connections are accepted.

Step 10	`RP/0/RP0/CPU0:router(config)# end`
	or
	`RP/0/RP0/CPU0:router(config)# commit`
		Saves configuration changes.
		When you issue the **end** command, the system prompts you to commit changes: Uncommitted changes found, commit them before exiting (yes/no/cancel)? [cancel]
		Entering **yes** saves configuration changes to the running configuration file, exits the configuration session, and returns the router to EXEC mode.
		Entering **no** exits the configuration session and returns the router to EXEC mode without committing the configuration changes.
		Entering **cancel** leaves the router in the current configuration session without exiting or committing the configuration changes.
		Use the **commit** command to save the configuration changes to the running configuration file and remain within the configuration session.

	Command	Description
Step 11	`RP/0/RP0/CPU0:router# show ssh`	(Optional) Displays all of the incoming and outgoing SSHv1 and SSHv2 connections to the router.
Step 12	`RP/0/RP0/CPU0:router# show ssh session details`	(Optional) Displays a detailed report of the SSHv2 connections to and from the router.

Creating a Device

From the Create window, you can define different types of devices.

To create a device, follow these steps:

Step 1 Choose **Inventory > Physical Inventory > Devices**.

The Device List window appears.

Step 2 Click the **Create** button.

The Create options window appears.

The **Create** options include the following:

- **Catalyst Switch**—A Catalyst device running the Catalyst Operating System.
- **Cisco Device**—Any router that runs the Cisco IOS. This includes Catalyst devices running Cisco IOS.
- **Terminal Server**—A device that represents the workstation that can be used to provision edge routers.
- **Cisco Configuration Engine (IE2100)**—Any Cisco Intelligence Engine (IE) 2100 series network device.

Step 3 See the following sections for instructions on creating each type of device.

- Creating a Catalyst Switch, page 2-5
- Creating a Cisco Device, page 2-6
- Creating a Terminal Server, page 2-7
- Creating a Cisco Configuration Engine Server, page 2-12

Creating a Catalyst Switch

To create a Catalyst switch, follow these steps:

Step 1 Choose **Inventory > Physical Inventory > Devices**.

The Device List window appears.

Step 2 Click the **Create** button.

The Create options window appears.

Step 3 Select **Catalyst Switch**.

The Create Catalyst Device window appears.

See the following sections for descriptions of these attribute fields:

- General Attributes, page 2-7
- Login and Password Attributes, page 2-9
- Device and Configuration Access Information Attributes, page 2-9
- SNMP v1/v2c Attributes, page 2-10

Step 4 Enter the desired information for the Catalyst device you are creating.

Step 5 To access the Additional Properties section of the **Create Catalyst Device**, click **Show**.

The Additional Properties window appears.

See the following sections for descriptions of the Additional Properties attribute fields:

- SNMP v3 Attributes, page 2-10
- Terminal Server Options Attributes, page 2-10
- Device Platform Information Attributes, page 2-11

Step 6 Enter any desired Additional Properties information for the Catalyst device you are creating.

Step 7 Click **Save**.

The Devices window reappears with the new Catalyst device listed.

Creating a Cisco Device

To create a Cisco device, follow these steps:

Step 1 Choose **Inventory > Physical Inventory > Devices**.

The Device List window appears.

Step 2 Click the **Create** button.

The Create options window appears.

Step 3 Select **Cisco Device**.

The Create Cisco Device window appears.

See the following sections for descriptions of the fields:

- General Attributes, page 2-7
- Login and Password Attributes, page 2-9
- Device and Configuration Access Information Attributes, page 2-9
- SNMP v1/v2c Attributes, page 2-10

Step 4 Enter the desired information for the Cisco IOS device you are creating.

Step 5 To access the Additional Properties section of the **Create Cisco Device**, click **Show**.

The Additional Properties window appears.

See the following sections for descriptions of the Additional Properties fields:

- SNMP v3 Attributes, page 2-10
- Terminal Server Options Attributes, page 2-10
- Device Platform Information Attributes, page 2-11

Step 6 Enter any desired Additional Properties information for the Cisco IOS device you are creating.

Step 7 Click **Save**.

The Devices window reappears with the new Cisco IOS device listed.

Creating a Terminal Server

To create a Terminal Server device, follow these steps:

Step 1 Choose **Inventory > Physical Inventory > Devices**.

The Device List window appears.

Step 2 Click the **Create** button.

The Create options window appears.

Step 3 Select **Terminal Server**.

The Create Terminal Server window appears.

See the following sections for descriptions of the fields:

- General Attributes, page 2-7
- Login and Password Attributes, page 2-9
- Device and Configuration Access Information Attributes, page 2-9
- SNMP v1/v2c Attributes, page 2-10

Step 4 Enter the desired information for the Terminal Server you are creating.

Step 5 To access the Additional Properties section of the **Create Terminal Server**, click **Show**.

The Additional Properties window appears.

See the following sections for descriptions of the Additional Properties fields:

- SNMP v3 Attributes, page 2-10
- Terminal Server Options Attributes, page 2-10
- Device Platform Information Attributes, page 2-11

Step 6 Enter any desired Additional Properties information for the Terminal Server device you are creating.

Step 7 Click **Save**.

The Devices window reappears with the new Terminal Server device listed.

General Attributes

The General Attributes sections contains the following fields:

- **Device Host Name** (required)—Must begin with a letter, digit, or underscore followed by letters, digits, underscores, spaces, hyphens, or dots ending with a letter, digit, or underscore. This field must match the name configured on the target router device. Limited to 256 characters.
- **Device Domain Name** (optional)—Must begin with a letter, digit, or underscore followed by letters, digits, underscores, spaces, hyphens, or dots ending with a letter, digit, or underscore. The name must match the domain name on the target router device.

- **Description** (optional)—Limited to 80 characters. Can contain any pertinent information about the device such as the type of device, its location, or other information that might be helpful to service provider operators.
- **Collection Zone** (optional)—Drop-down list of all collection zones within the Prime Fulfillment. Choices include: None and all collection zones within the Prime Fulfillment. Default: None.
- **Management IP Address** —Valid IP address of the device that Prime Fulfillment uses to configure the target router device.
- **Element Management Key** —Valid IP address of the device that Prime Fulfillment.
- **Interfaces** (optional)—Click the **Edit** button to view, add, edit, and delete all interfaces associated with the device. See Table 2-1 for a description of the Interfaces fields.

Table 2-1 Create Catalyst Device Interfaces Fields

Field	Description	Additional
Interface Name	Name of this interface.	List can be sorted by this field. Limited to 80 characters.
IPV4 Address	IPv4 address associated with this interface.	
IPV6 Address	IPv6 address associated with this interface.	
Encapsulation	The Layer 2 Encapsulation for this device.	DEFAULT DOT1Q ETHERNET ISL FRAME_RELAY FRAME_RELAY_IETF HDLC PPP ATM AAL5SNAP AAL0 AAL5 AAL5MUX AAL5NLPID AAL2 ENCAP_QinQ GRE
Port Type		NONE ACCESS TRUNK ROUTED

Table 2-1 Create Catalyst Device Interfaces Fields (continued)

Field	Description	Additional
Description	Description of the device interface.	Description of the device interface.
IP Address Type	IP address type.	IP address type.

- **Associated Groups** (optional)—Click the **Edit** button to view, add, and remove all Device Group associations.

Login and Password Attributes

The Login and Password Information section contains the following fields:

- **Login User** (optional)—Not required by Prime Fulfillment. However, collection and upload/download will not function without the Login User and Login Password as Prime Fulfillment will not be able to access the device. Should match what is configured on the target router device. Limited to 80 characters.
- **Login Password** (optional)—Not required by Prime Fulfillment. However, collection and upload/download will not function without the Login User and Login Password, because Prime Fulfillment will not be able to access the device. Should match what is configured on the target router device. Limited to 80 characters.
- **Verify Login Password** (optional)—Must match the Login Password field. Limited to 80 characters.
- **Enable User** (optional)—Not required by Prime Fulfillment. However, collection and upload/download only function if the Login User has sufficient privileges to configure the router in EXEC mode. Should match what is configured on the target router device. Limited to 80 characters.
- **Enable Password** (optional)—Not required by Prime Fulfillment. However, collection and upload/download only function if the Login User has sufficient privileges to configure the router in EXEC mode. Should match what is configured on the target router device. Limited to 80 characters.
- **Verify Enable Password** (optional)—Must match the Enable Password field. Limited to 80 characters.

Device and Configuration Access Information Attributes

The Device and Configuration Access Information section contains the following fields:

- **Terminal Session Protocol** (optional)—Configures the method of communication between Prime Fulfillment and the device. Choices include: Telnet, Secure Shell (SSH), CNS, RSH, and SSH version 2 (SSHv2). In previous versions of Prime Fulfillment, this field was called the Transport field. Default: The default set in the DCPL properties.
- **Config Access Protocol** (optional)—Administers the access protocol for config upload and download. Choices include: Terminal, TFTP, FTP, and RCP. Default: The default set in the DCPL properties.
- **OS** (optional)—The choices are: IOS and IOS_XR. Applicable for Creating a Cisco Device and for Creating a Terminal Server.
- **SNMP Version** (optional)—Configures the version of SNMP to use when communicating with the device. Choices include: SNMP v1/v2c and SNMP v3. Default: The default set in the DCPL properties.

SNMP v1/v2c Attributes

The SNMP v1/v2c section contains the following fields:

- **Community String RO** (optional)—SNMP Read-Only Community String. Many tasks use SNMP to access the device. This field must match what is configured on the target router device. Limited to 80 characters.
- **Community String RW** (optional)—SNMP Read-Write Community String. Many tasks use SNMP to access the device. This field must match what is configured on the target router device. Limited to 80 characters.

SNMP v3 Attributes

The SNMP v3 section contains the following fields:

- **SNMP Security Level** (optional)—Choices include: Default (<*default_set_in_DCP*L>), Authentication/No Encryption, Authentication/Encryption, and No Authentication/No Encryption. Default: Default (<*default_set_in_DCP*L>). Note: When you change the DCPL property, the <*default_set_in_DCP*L> variable changes.
- **Authentication User Name** (optional)—User name configured on the specified device router. User must have permission to the object identification numbers (OIDs) specified in the security request (that is, write permission for a set request, and read permission for a get request). Should match what is configured on the target router device. Should be provisioned if the SNMP Security Level is Authentication/No Encryption or Authentication/Encryption. Limited to 80 characters.
- **Authentication Password** (optional)—Should be provisioned if the SNMP Security Level is Authentication/No Encryption or Authentication/Encryption. Should match what is configured on the target router device. Limited to 80 characters.
- **Verify Authentication Password** (optional)—Must match the Encryption Password field. Limited to 80 characters.
- **Authentication Algorithm** (optional)—Should be provisioned if the SNMP Security Level is Authentication/No Encryption or Authentication/Encryption. Choices include: None, MD5, and SHA. Default: None.
- **Encryption Password** (optional)—In previous versions of Prime Fulfillment, this field was called Privacy Password. Should match what is configured on the target router device. Should be provisioned if the SNMP Security Level is Authentication/Encryption. Limited to 80 characters.
- **Verify Encryption Password** (optional)—Must match the Encryption Password field. Limited to 80 characters.
- **Encryption Algorithm** (optional)—In previous versions of Prime Fulfillment, this field was called Privacy Protocol. Should be provisioned if the SNMP Security Level is Authentication/Encryption. Choices include: None and DES 56. Default: None.

Terminal Server Options Attributes

The Terminal Server Options section contains the following fields:

- **Terminal Server** (optional)—Choices include: None and the list of existing Terminal Server names. Default: None.
- **Port** (optional)—Disabled until a Terminal Server is selected. Range: 0-65535. Default: 0.

  The following fields are also available when you are creating a Cisco Device:

- **Fully Managed** (optional)—If the Fully Managed check box is checked, the device becomes a fully managed device. Prime Fulfillment performs additional management actions only for fully managed devices. These actions include e-mail notifications upon receipt of device configuration changes originated outside Prime Fulfillment and the scheduling of enforcement audit tasks upon detection of possible intrusion. Default: Not selected and therefore not selected.
- **Device State** (optional)—Choices include: ACTIVE and INACTIVE. ACTIVE indicates that the router has been plugged on the network and can be part of Prime Fulfillment tasks such as collect config and provisioning. INACTIVE indicates the router has not been plugged-in. Default: ACTIVE.
- **CNS Identification**—Required if the Device Event Identification field is set to CNS_ID. Only valid characters that Cisco IOS allows are alphanumeric characters and (.) (-) (_).
- **Device Event Identification** (optional)—Indicates whether the CNS Identification field contains a HOST_NAME or CNS_ID. Default: HOST_NAME.
- **Most Recent CNS event** (optional)—Choices include: None, CONNECT, and DISCONNECT. Changing from the default to None is not recommended. Note: The last connect or disconnect CNS TIBCO event received by Prime Fulfillment for each CNS-enabled IOS device is automatically recorded.
- **IE2100** (optional)—Disabled unless the Device State field is INACTIVE or the Terminal Session Protocol field is CNS. A valid IE2100 must be selected if the Terminal Session Protocol is CNS. Choices include: None and the list of existing IE2100 names. Default: None.
- **Cisco Configuration Engine Software Version** (optional)—Choices include: 1.3, 1.3.1, 1.3.2, 1.4, 1.5, 2.0, 3.0, and 3.5. This is the release version of Cisco Configuration Engine that manages the IOS device. Default: 1.4.
- **CNS Device Transport** (optional)—Choices include: HTTP and HTTPS. This field determines what will be the transport mechanism used by Prime Fulfillment to create, delete, or edit devices in the Cisco Configuration Engine repository. If HTTPS is used, the Cisco Configuration Engine must be running in secure mode. Default: HTTP.

Device Platform Information Attributes

The Device Platform Information section contains the following fields:

- **Platform** (optional)—Should match what is configured on the target router device. Limited to 80 characters.
- **Software Version** (optional)—Should match what is configured on the target router device. Limited to 80 characters.
- **Image Name** (optional)—Should match what is configured on the target router device. Limited to 80 characters.
- **Serial Number** (optional)—Should match what is configured on the target router device. Limited to 80 characters.
- **Device Owner's Email Address** (optional)—Used in the To: field when the Email button is selected from the device list. Limited to 80 characters and must be valid Email format.

Creating a Cisco Configuration Engine Server

Note To use the Cisco Configuration Engine server functionality on Prime Fulfillment, you must first set up the Cisco Configuration Engine server and the Prime Fulfillment workstation as explained in Appendix B, "Setting Up Cisco Configuration Engine with Prime Fulfillment" in the *Cisco Prime Fulfillment Installation Guide 6.2*. You must also create a Cisco IOS device to communicate with the Cisco Configuration Engine server. See Appendix A, "Creating a Cisco IOS Device Using the Cisco CNS Device Access Protocol". The Cisco configuration engine server is referred to as IE2100 throughout the Prime Fulfillment user interface. This is the model number of an appliance that is sued to run the configuration engine software.

To create a Cisco Configuration Engine server, follow these steps:

Step 1 Choose **Inventory > Physical Inventory > Devices**.

The Device List window appears.

Step 2 Click the **Create** button.

The Create options window appears.

Step 3 Select **Cisco Configuration Engine**.

The Create New Cisco Configuration Engine window appears.

The General section of the Create IE2100 Device window contains the following fields:

- **Device Host Name** (required)—Must begin with a letter, digit, or underscore followed by letters, digits, underscores, spaces, hyphens, or dots ending with a letter, digit, or underscore. This field must match the name configured on the target router device. Limited to 256 characters.
- **Device Domain Name** (optional)—Must begin with a letter, digit, or underscore followed by letters, digits, underscores, spaces, hyphens, or dots ending with a letter, digit, or underscore. The name must match the domain name on the target router device.
- **Description** (optional)—Limited to 80 characters. Can contain any pertinent information about the device such as the type of device, its location, or other information that might be helpful to service provider operators.
- **IPV4 Address** (optional)—Valid IPv4 address of the Cisco Configuration Engine server that Prime Fulfillment uses to configure the target router device.

Step 4 Enter the desired information for the Cisco Configuration Engine server you are creating.

Step 5 Click **Save**.

The Devices window reappears with the new Cisco Configuration Engine server listed.

Copying a Device

From the Copy window, you receive a copy of the chosen device and can name it and change values.

To access the Copy window, follow these steps:

Step 1 Choose **Inventory > Physical Inventory > Device.**

The Device List window appears.

Step 2 Select a single device to copy by checking the check box to the left of the Device Name.

Step 3 Click the **Copy** button. This button is only enabled if a device is selected.

A window appropriate to the type of device selected to copy appears. You receive an exact copy of the selected device but the Name, Management IP Address, all Interfaces, and VPNSM blades for a Catalyst Switch running Cisco IOS are blanked out and you must fill in the required information and save this new device. See the "Creating a Device" section on page 2-5 for specifics.

Editing a Device

From the Edit window, you can modify the fields that have been specified for a particular device.

To access the Edit window, follow these steps:

Step 1 Choose **Inventory > Physical Inventory > Devices**.

The Device List window appears.

Step 2 Select a single device to edit by checking the box to the left of the Device Name. You can also select a device to edit by clicking on the hyperlink of the device name.

Step 3 Click the **Edit** button. This button is only enabled if a device is selected.

The Edit window appropriate to the type of device selected appears. For example, if you selected a Cisco IOS device the Edit Cisco IOS Device window appears.

Step 4 Enter the changes you want to make to the selected device.

Step 5 Click **Save**.

The changes are saved and the Devices window reappears.

Deleting Devices

From the Delete window, you can remove selected devices from the database.

To access the Delete window, follow these steps:

Step 1 Choose **Inventory > Physical Inventory > Devices**.

The Device List window appears.

Step 2 Select one or more devices to delete by checking the check box(es) to the left of the Device Name(s).

Step 3 Click the **Delete** button. This button is enabled only if one or more devices are selected.

The Confirm Delete window appears.

Step 4 Click the **Delete** button to confirm that you want to delete the device(s) listed.

The Devices window reappears with the specified device(s) deleted.

Editing a Device Configuration

From the Config window, you can edit the configuration for a specified device.

To access the Config window, follow these steps:

Step 1 Choose **Inventory > Physical Inventory > Devices**.

The Device List window appears.

Step 2 Select a single device to modify by checking the check box to the left of the Device Name.

Step 3 Click the **Config** button.

The Device Configurations window for the selected device appears.

Step 4 Check the box to the left of the Date for the configuration that you want to modify and click the **Edit** button. This button is only enabled if a device is selected.

The Device Configuration window for the selected device appears.

Step 5 Enter the changes you want to make to the selected device configuration.

Step 6 Click **Save**.

The changes are saved and the Device Configurations window reappears.

Step 7 Click **OK** to return to the Devices window.

E-mailing a Device's Owner

From the E-mail window, you can send a device report via e-mail to the owners of specified devices.

To access the E-mail window, follow these steps:

Step 1 Choose **Inventory > Physical Inventory > Devices**.

The Device List window appears.

Step 2 Select the devices for which you want to send a device report by checking the check box(es) to the left of the Device Name(s).

Step 3 Click the **E-mail** button. This button is only enabled if one or more devices are selected.

The Send Mail to Device Owners window appears.

Step 4 Compose the e-mail that you want to send to the selected device owners.

Step 5 Click **Send**.

The e-mail is sent and the Devices window reappears.

Device Configuration Collection

We recommend that a Task Manager Collect Configuration task is used to add interface configuration to Devices in the Prime Fulfillment Repository. A Task Manager Collect Configuration task connects to the physical device in the network, collects the device information from the router (including interface configuration), and populates the Prime Fulfillment Repository with this information.

For details of how to add Device interface configuration using a Task Manager Collect Configuration task, see Task Manager, page 10-23.

Synchronizing the Prime Fulfillment Repository with Device Configuration

Note The accuracy of Diagnostics is dependent on up-to-date device information. We recommend that the device configuration is resynchronized with the physical devices after any configuration changes and at periodic intervals. This ensures that the device configuration held in the Prime Fulfillment inventory is consistent with the physical devices in the network.

We recommend that device configuration is kept up-to-date using a scheduled Task Manager task. Either Collect Configuration or Collect Configuration from File can be used. For details of how to create a scheduled Task Manager Collect Configuration task, see Task Manager, page 10-23. All PE and P routers in the MPLS network should have their configuration collected using a scheduled Task Manager Collect Configuration task. The Task Manager Collect Configuration task collects details of interface configuration and other device attributes. The interval at which Task Manager Collect Configuration tasks should be scheduled to run depends on the frequency of configuration changes to the network. We recommend running the Task Manager Collect Configuration task daily on each P and PE router.

Providers

This section describes how to create and manage providers. This section includes the following topics:

Creating a Provider

From the Create Provider window, you can create different providers.

To create a provider, follow these steps:

Step 1 Choose **Service Design > Resources > Providers**.

The Providers window appears.

Step 2 Click the **Create** button.

The Create Provider window appears.

The Create Provider window contains the following fields:

- **Name** (required)—Must begin with a letter. Can contain letters, numbers, and these punctuation characters: period, underscore, and dash. Limited to 80 characters.
- **BGP AS** (required)—Each BGP autonomous system is assigned a unique 16-bit number by the same central authority that assigns IP network numbers. Range: 1 to 65535.
- **Contact Information** (optional)—Any pertinent information about the provider that could be helpful to service provider operators. Limited to 256 characters.

Step 3 Enter the name, BGP AS, and any contact information for the Provider that you are creating.

Step 4 Click **Save**.

The Providers window reappears with the new provider listed.

Editing a Provider

From the Edit Provider window, you can modify the fields that have been specified for a particular provider.

To access the Edit Provider window, follow these steps:

Step 1 Choose **Service Design > Resources > Providers**.

The Providers window appears.

Step 2 Select a single provider to modify by checking the check box to the left of the Provider Name.

Step 3 Click the **Edit** button. This button is only enabled if a customer is selected.

The Edit Provider window appears.

Step 4 Enter the changes you want to make to the selected provider.

Step 5 Click **Save**.

The changes are saved and the Providers window reappears.

Deleting Providers

From the Delete window, you can remove selected providers from the database.

To access the Delete window, follow these steps:

Step 1 Choose **Service Design > Resources > Providers**.

The Providers window appears.

Step 2 Select provider(s) to delete by checking the check box to the left of the Provider Name.

Step 3 Click the **Delete** button. This button is enabled only if one or more Providers are selected.

The Confirm Delete window appears.

Step 4 Click the **Delete** button to confirm that you want to delete the provider(s) listed.

The Providers window reappears with the specified provider(s) deleted.

Provider Regions

A Provider Region is considered to be a group of provider edge routers (PEs) within a single BGP autonomous system. The primary objective for defining Provider Regions is to allow a provider to employ unique IP address pools in large Regions, such as Europe, Asia Pacific, and so forth.

This sections covers the following topics:

Creating a Provider Region

From the Create Provider Region window, you can create different PE regions.

To create a provider region, follow these steps:

Step 1 Choose **Inventory > Resources > Provider Regions**.

The Provider Regions window appears.

Step 2 Click the **Create** button.

The Create Provider Regions window appears.

Step 3 Enter the name and information for the Provider that you are creating. To enter the provider name follow these steps:

a. Click the **Select** button next to the Provider field.

A list of provider names appears.

b. Click the radio button next to provider name and then **Select**.

Step 4 Click **Cancel** if you do not want to save this information, and you will proceed to the previous window.

Otherwise, click **Save**. The changes are then saved and the Customer Site window reappears.

Editing a Provider Regions

From the Edit Provider Regions window, you can modify the fields that have been specified for a particular provider region.

To access the Edit Provider Regions window, follow these steps:

Step 1 Choose **Inventory > Resources > Provider Regions**.

The Provider Regions window appears.

Step 2 Select a single site name to modify by checking the check box to the left of the PE Region Name.

Step 3 Click the **Edit** button. This button is only enabled if a PE region name is selected.

The Edit Provider Region window appears.

Step 4 Enter the changes you want to make to the selected provider region.

Step 5 Click **Cancel** if you do not want to save this information, and you will proceed to the previous window.

Otherwise, click **Save**. The changes are then saved and the Provider Region window reappears.

Deleting Provider Regions

From the Delete window, you can remove selected provider regions from the database.

To access the Delete window, follow these steps:

Step 1 Choose **Inventory > Resources > Provider Regions**.

The Provider Regions window appears.

Step 2 Select one or more region to delete by checking the check box to the left of the PE Region Name.

Step 3 Click the **Delete** button. This button is enabled only if one or more PE region name are selected.

The Confirm Delete window appears.

Step 4 Click **Cancel** if you do not want to save this information, and you will proceed to the previous window.

Otherwise, click **Delete** to confirm that you want to delete the region name(s) listed. The Provider Regions window reappears with the specified PE region name(s) deleted.

Provider Devices

The PE Devices feature provides a list of provider edge routers (PEs) that have been associated with the region, either through the PE editor or Inventory Manager.

This section covers the following topics:

- Creating a Provider Devices, page 2-18
- Editing a Provider Devices, page 2-19
- Deleting Provider Devices, page 2-19

Creating a Provider Devices

From the Create Provider Device window, you can create different PE regions.

To create a provider region, follow these steps:

Step 1 Choose **Inventory > Resources > Provider Devices**.

The PE Devices window appears.

Step 2 Click the **Create** button.

The Create New Provider Devices window appears.

Step 3 To enter the Device Name follow these steps:

a. Click the **Select** button next to the Device Name field.

A list of Device Name window appears.

b. Click the radio button next to device name and then **Select**.

Step 4 To enter the PE Region Name follow these steps:

a. Click the **Select** button next to the PE Region Name field.

A list of Region Name window appears.

b. Click the radio button next to device name and then **Select**.

Step 5 Select the PE Role Type from drop-down list. The options are N-PE, U-PE, P, and PE-AGG.

Step 6 Check the check box next to the 6VPE.

Step 7 Click **Cancel** if you do not want to save this information, and you will proceed to the previous window.

Otherwise, click **Save**. The changes are then saved and the Provider Device window reappears.

Editing a Provider Devices

From the Edit Provider Devices window, you can modify the fields that have been specified for a particular provider region.

To access the Edit Provider Devices window, follow these steps:

Step 1 Choose **Inventory > Resources > Provider Devices**.

The PE Devices window appears.

Step 2 Select a single site name to modify by checking the check box to the left of the Device Name.

Step 3 Click the **Edit** button. This button is only enabled if a PE Device name is selected.

The Edit Provider Region window appears.

Step 4 Enter the changes you want to make to the selected PE device name.

Step 5 Click **Cancel** if you do not want to save this information, and you will proceed to the previous window.

Otherwise, click **Save**. The changes are then saved and the Provider Device window reappears.

Deleting Provider Devices

From the Delete window, you can remove selected provider device from the database.

To access the Delete window, follow these steps:

Step 1 Choose **Inventory > Resources > Provider Devices**.

The PE Devices window appears.

Step 2 Select one or more region to delete by checking the check box to the left of the Device Name.

Step 3 Click the **Delete** button. This button is enabled only if one or more PE Device name are selected.

The Confirm Delete window appears.

Step 4 Click **Cancel** if you do not want to save this information, and you will proceed to the previous window.

Otherwise, click **Delete** to confirm that you want to delete the provider device(s) listed. The Provider Devices window reappears with the specified provider device(s) deleted.

Using the Inventory Manager Window

To access the Inventory Manager, choose **Inventory > Physical Inventory > Inventory Manager**.

From the Inventory Manager window you can import devices or open a list of devices, providers, or customers.

This section covers the following topics:

Importing Devices

To import a device, it must be in an existing directory on the same server that is running Prime Fulfillment. After a device is imported into the Prime Fulfillment repository, you can assign it to a customer or provider, if desired.

To import devices with configuration files, follow these steps:

Step 1 Choose **Inventory > Physical Inventory > Inventory Manager**.

Step 2 Click the **Import Devices** button.

The Import Devices from Configuration Files window appears.

Step 3 Click the **Select** button.

The Select Device Configuration File window appears.

Step 4 At the **Select Device Configuration File** window, enter the directory on the Prime Fulfillment server where the configuration files reside, and the **Import Devices from Configuration Files** window appears.

Step 5 Select as many of the configuration files as you want to import by checking the box to the left of the Configuration File name.

Step 6 If you want to import devices from more than one directory, you can repeat Steps 3 through 6.

Step 7 Click **Import**.

The **General Attributes** window appears with the added information.

Step 8 Click **Save**.

Opening and Editing Devices

To open device configuration files to bulk edit, follow these steps:

Step 1 Choose **Inventory > Physical Inventory > Inventory Manager**.

Step 2 Click the **Open** button.

The **Open** drop-down list appears. The **Open** options include the following:

- **Devices**—Every network element that Prime Fulfillment manages.

Note To edit a PE, **Open Provider**, *not* **Open Devices**.

- **Provider**—PEs belonging to a specific provider.
- **Customer**—CEs belonging to a specific customer.

Step 3 Select **Devices**.

The Select Device window appears.

Step 4 Select a device to open by checking the check box to the left of the Device Name. You can select more than one device to open.

Step 5 Click the **Select** button.

The General Attributes window appears containing information on the selected devices.

Step 6 To view specific attributes click the **Attributes** button.

The Attributes options window appears.

Step 7 Select the type of attribute to display.

See the following sections for descriptions of these attribute fields.

- General Attributes, page 2-23
- Password Attributes, page 2-24
- SNMP Attributes, page 2-24
- CNS Attributes, page 2-25
- Platform Attributes, page 2-25
- Interfaces, page 2-26

Step 8 To bulk edit an attribute, do the following:

a. Check the one or more boxes to the left of the Device Name.

b. Check the check box above the attribute name column.

c. Click the **Edit** button.

Step 9 Enter the changes you want to make.

Step 10 Click **Save**.

The changes are saved.

Opening and Editing PEs

To open PE files to bulk edit, follow these steps:

Step 1 Choose **Inventory > Physical Inventory > Inventory Manager**.

Step 2 Click the **Open** button.

The **Open** drop-down list appears. The **Open** options include the following:

- **Devices**—Every network element that Prime Fulfillment manages.
- **Provider**—PEs belonging to a specific provider.
- **Customer**—CEs belonging to a specific customer.

Step 3 Select **Provider**.

The Select Provider window appears.

Step 4 Select a provider by clicking the radio button to the left of the Provider Name.

Step 5 Click the **Select** button.

The General Attributes Provider window appears showing the PEs assigned to the selected provider.

Step 6 To view specific attributes click the **Attributes** button.

The Attributes options window appears.

Step 7 Select the type of attribute to display.

See the following sections for descriptions of these attribute fields.

- General Attributes, page 2-23
- Password Attributes, page 2-24
- SNMP Attributes, page 2-24
- CNS Attributes, page 2-25
- Platform Attributes, page 2-25
- PE Attributes, page 2-27
- Interfaces, page 2-26

Step 8 To bulk edit an attribute, do the following:

a. Check the one or more boxes to the left of the Host or Device Name.

b. Check the check box above the attribute name column.

c. Click the **Edit** button.

Step 9 Enter the changes you want to make.

Step 10 Click **Save**.

The changes are saved.

Opening and Editing CEs

To open CE files to bulk edit, follow these steps:

Step 1 Choose **Inventory > Physical Inventory > Inventory Manager**.

Step 2 Click the **Open** button.

The **Open** drop-down list appears. The **Open** options include the following:

- **Devices**—Every network element that Prime Fulfillment manages.
- **Provider**—PEs belonging to a specific provider.
- **Customer**—CEs belonging to a specific customer.

Step 3 Select **Customer**.

The Select Customer window appears.

Step 4 Select a customer by clicking the radio button to the left of the Customer Name.

Step 5 Click the **Select** button.

The General Attributes Customer window appears showing the CEs assigned to the selected customer.

Step 6 To view specific attributes click the **Attributes** button.

The Attributes Options window appears.

Step 7 Select the type of attribute to display.

See the following sections for descriptions of these attribute fields.

- General Attributes, page 2-23
- Password Attributes, page 2-24
- SNMP Attributes, page 2-24
- CNS Attributes, page 2-25
- Platform Attributes, page 2-25
- CPE Attributes, page 2-27
- Interfaces, page 2-26

Step 8 To bulk edit an attribute, do the following:

a. Check the one or more boxes to the left of the Host or Device Name.

b. Check the check box above the attribute name column.

c. Click the **Edit** button.

Step 9 Enter the changes you want to make.

Step 10 Click **Save**.

The changes are saved.

General Attributes

The General Attributes Devices window contains the following:

- **Host** —Must begin with a letter, digit, or underscore followed by letters, digits, underscores, spaces, hyphens, or dots ending with a letter, digit, or underscore. This field is required and must match the name configured on the target router device. Limited to 256 characters.
- **Device Type**—The device type includes the following devices:
  - Cisco Router
  - Catalyst OS device
  - Terminal server
  - IE2100 (Cisco Configuration Engine server)
- **Description**—Can contain any pertinent information about the device, such as the type of device, its location, or other information that might be helpful to service provider operators. Limited to 80 characters.

- **Management IP Address**—Valid IP address of the device that Prime Fulfillment uses to configure the target router device. This IP address must be reachable from the Prime Fulfillment host.
- **Device Domain Name**—Must begin with a letter, digit, or underscore followed by letters, digits, underscores, spaces, hyphens, or dots ending with a letter, digit, or underscore. The name must match the domain name on the target router device.
- **Terminal Session Protocol**—Configures the method of communication between Prime Fulfillment and the device. Choices include: Telnet, Secure Shell (SSH), SSH version 2 (SSHv2), CNS, and RSH. Default: Telnet.
- **Config Access Protocol**—Administers the access protocol for config upload and download. Choices include: Terminal, TFTP, FTP, and RCP. Default: Terminal
- **Device Groups**—Lists the names of the Device Groups. You can add and modify Device Groups in this column.

Password Attributes

The Password Attributes Devices window contains the following:

- **Device Name**—Must begin with a letter, digit, or underscore followed by letters, digits, underscores, spaces, hyphens, or dots ending with a letter, digit, or underscore. This field is required and must match the name configured on the target router device. Limited to 256 characters.
- **Login User**—Not required by Prime Fulfillment. However, collection and upload/download will not function without the Login User and Login Password, as Prime Fulfillment will not be able to access the device. Should match what is configured on the target router device. Limited to 80 characters.
- **Login Password**—Displayed as stars (*). Not required by Prime Fulfillment. However, collection and upload/download will not function without the Login User and Login Password, as Prime Fulfillment will not be able to access the device. Should match what is configured on the target router device. Limited to 80 characters.
- **Enable User**—Not required by Prime Fulfillment. However, collection and upload/download only function if the Login User has sufficient privileges to configure the router in EXEC mode. Should match what is configured on the target router device. Limited to 80 characters.
- **Enable Password**—Displayed as stars (*). Not required by Prime Fulfillment. However, collection and upload/download only function if the Login User has sufficient privileges to configure the router in EXEC mode. Should match what is configured on the target router device. Limited to 80 characters.
- **Community String RO** —Many tasks use SNMP to access the device. This field must match what is configured on the target router device. Limited to 80 characters.
- **Community String RW**— Many tasks use SNMP to access the device. This field must match what is configured on the target router device. Limited to 80 characters.

SNMP Attributes

The SNMP Attributes Devices window contains the following:

- **Device Name**—Must begin with a letter, digit, or underscore followed by letters, digits, underscores, spaces, hyphens, or dots ending with a letter, digit, or underscore. This field is required and must match the name configured on the target router device. Limited to 256 characters.
- **SNMP Version**—Choices include: SNMP v1/v2c, and SNMP v3. The default value is determined by the setting in the DCPL property SnmpService\defaultSNMPVersion. (See Appendix B, "Property Settings" for more details.)

- **Security Level**—Choices include: No Authentication/No Encryption, Authentication/No Encryption, and Authentication/Encryption. Default: No Authentication/No Encryption.
- **Authentication User Name**—User name configured on the specified device router. User must have permission to the object identification numbers (OIDs) specified in the security request (that is, write permission for a set request, and read permission for a get request). Should match what is configured on the target router device. Should be provisioned if the SNMP Security Level is Authentication/No Encryption or Authentication/Encryption. Limited to 80 characters.
- **Authentication Password**—Displayed as stars (*). Should be provisioned if the SNMP Security Level is Authentication/No Encryption or Authentication/Encryption. Should match what is configured on the target router device. Limited to 80 characters.
- **Authentication Algorithm**—Should be provisioned if the SNMP Security Level is Authentication/No Encryption or Authentication/Encryption. Choices include: None, MD5, and SHA. Default: None.
- **Encryption Password**—Displayed as stars (*). In previous versions, this field was called Privacy Password. Should match what is configured on the target router device. Should be provisioned if the SNMP Security Level is Authentication/Encryption. Limited to 80 characters.
- **Encryption Algorithm**—In previous versions, this field was called Privacy Protocol. Should be provisioned if the SNMP Security Level is Authentication/Encryption. Choices include: None and DES 56. Default: None.

CNS Attributes

The CNS Attributes Devices window contains the following:

- **Device Name**—Must begin with a letter, digit, or underscore followed by letters, digits, underscores, spaces, hyphens, or dots ending with a letter, digit, or underscore. This field is required and must match the name configured on the target router device. Limited to 256 characters.
- **IE2100 Name**—Disabled unless the Device-State field is Inactive or the Terminal Session Protocol field is CNS. A valid Cisco Configuration Engine server must be selected if the Terminal Session Protocol is CNS. Choices include: None and the list of existing Cisco Configuration Engine server names. Default: None.
- **Device State**—Choices include: Active and Inactive. Active indicates that the router has been plugged on the network and can be part of Prime Fulfillment tasks such as collect config and provisioning. Inactive indicates the router has not been plugged-in. Default: Active.
- **Event Identification**—Indicates whether the CNS Identification field contains a HOST NAME or CNS ID. Default: HOST NAME.
- **CNS Identification**—Required if the Event Identification field is set to CNS ID. Can contain letters, numbers, and these punctuation characters: period, underscore, and dash.

Platform Attributes

The Platform Attributes Devices window contains the following:

- **Device Name**—Must begin with a letter, digit, or underscore followed by letters, digits, underscores, spaces, hyphens, or dots ending with a letter, digit, or underscore. This field is required and must match the name configured on the target router device. Limited to 256 characters.
- **Platform**—Should match what is configured on the target router device. Limited to 80 characters.
- **Software Version**—Should match what is configured on the target router device. Limited to 80 characters.

- **Image Name**—Should match what is configured on the target router device. Limited to 80 characters.
- **Serial Number**—Should match what is configured on the target router device. Limited to 80 characters.

Interfaces

The Interfaces Devices window contains the following:

- **Host** —Must begin with a letter, digit, or underscore followed by letters, digits, underscores, spaces, hyphens, or dots ending with a letter, digit, or underscore. This field is required and must match the name configured on the target router device. Limited to 256 characters.
- **Interface Name**—Must begin with a letter, digit, or underscore followed by letters, digits, underscores, spaces, hyphens, or dots ending with a letter, digit, or underscore. This field is required. Limited to 256 characters.
- **Interface Type**—Specifies the type of interface. It is a display-only field.
- **Interface Description**—Description of the interface. This field is display-only. Field is populated by importing a configuration file.
- **Interface IP Address**—IPv4 address associated with this interface.
- **Interface IPv6 Address**—IPv6 address associated with this interface.
- **Encapsulation**—The Layer 2 Encapsulation for this device. It is a display-only field. Possible values are:
  - DEFAULT
  - DOT1Q
  - ETHERNET
  - ISL
  - FRAME_RELAY
  - FRAME_RELAY_IETF
  - HDLC
  - PPP
  - ATM
  - AAL5SNAP
  - AAL0
  - AAL5
  - AAL5MUX
  - AAL5NLPID
  - AAL2
  - ENCAP_QinQ
  - GRE
- **Port Type**—Choices include: Access, Trunk, Routed, and None.

PE Attributes

The PE Attributes Provider window contains the following:

- **Device Name**—Must begin with a letter, digit, or underscore followed by letters, digits, underscores, spaces, hyphens, or dots ending with a letter, digit, or underscore. This field is required and must match the name configured on the target router device. Limited to 256 characters.
- **Provider**—Lists the names of providers. Must begin with a letter. Can contain letters, numbers, and these punctuation characters: period, underscore, and dash. Limited to 80 characters. You can sort the list by provider name.
- **Region**—Lists the names of regions. Must begin with a letter. Can contain letters, numbers, and these punctuation characters: period, underscore, and dash. Limited to 80 characters. You can sort the list by region name.
- **Role**—Choices include: N-PE, U-PE, P, PE_AGG.
- **Loopback Interface**—Loopback address is the IP address of any loopback interface on the device. You can select one of the loopback interfaces for this field and use the IP address on that loopback interface.
- **Managed**—Provisioned by Prime Fulfillment. Check the check box for yes. Default is no.

CPE Attributes

The CPE Attributes Customer window contains the following:

- **Device Name**—Must begin with a letter, digit, or underscore followed by letters, digits, underscores, spaces, hyphens, or dots ending with a letter, digit, or underscore. This field is required and must match the name configured on the target router device. Limited to 256 characters.
- **Customer**—Lists the names of customers. Must begin with a letter. Can contain letters, numbers, and these punctuation characters: period, underscore, and dash. Limited to 80 characters. You can sort the list by customer name.
- **Site**—Lists the names of sites. Must begin with a letter. Can contain letters, numbers, and these punctuation characters: period, underscore, and dash. Limited to 80 characters. You can sort the list by site name.
- **Management Type**—Choices include: Managed, Unmanaged, Managed - Management LAN, Unmanaged - Management LAN, Directly Connected, Directly Connected Management Host, Multi-VRF, and Unmanaged Multi-VRF.

Assigning Devices

To assign a device to a provider or customer, follow these steps:

Step 1 Choose **Inventory > Physical Inventory > Inventory Manager**.

Step 2 Click the **Open** button.

The **Open** drop-down list appears.

Step 3 Select **Devices**.

The Select Device window appears.

Step 4 Select a device to open by checking the box to the left of the Device Name. You can select more than one device to open.

Step 5 Click the **Select** button.

The General Attributes Devices window appears containing information on the selected devices.

Step 6 Click the **Assign CE/PE** button.

Step 7 Select **Customer** or **Provider**.

The corresponding **Select Customer** or **Select Provider** window appears.

Step 8 Select the customer or provider to which you want to assign the device by checking the box to the left of the Customer or Provider Name.

Step 9 Click the **Select** button.

If you assigned the device to a provider, the PE Attributes window appears. If you assigned the device to a customer, the CPE Attributes window appears.

Step 10 To save the assigned devices to the Prime Fulfillment repository, you must specify the Site in the CPE Attributes window or the Region in the PE Attributes window. Do the following:

a. Check the one or more boxes to the left of the Device Name.

b. Check the check box above the **Site** or **Region** column.

c. Click the **Edit** button. The **Edit Attributes** window appears.

d. Click **Select**. The **Select Site** or **Select Region** window appears.

e. Select a site or region by checking the box to the left of the Site Name or Region Name.

f. Click **Save**.

Step 11 You can choose to edit attributes as desired. Enter any changes you want to make.

Step 12 Click **Save**.

The PE or CPE is saved to the Prime Fulfillment repository.

Device Groups

Every network element that Cisco Prime Fulfillment manages must be defined as a device in the system. After you have defined your network elements as devices, you can organize the devices into groups for collection and management purposes.

This section describes how to create, edit, and delete device groups and e-mail device group owners. This section includes the following topics:

Creating a Device Group

From the Create Device Group window, you can create different device groups.

To create a device group, follow these steps:

Step 1 Choose **Inventory > Physical Inventory > Device Groups**.

The Device Groups window appears.

Step 2 Click the **Create** button.

The Create Device Group window appears.

Step 3 Enter the name and the description of the Device Group that you are creating.

Step 4 Click **Edit**.

The Select Group Members window appears.

Step 5 Select the devices that you want to be group members by checking the check box to the left of the device name.

Step 6 Click **OK**.

The Create Device Group window appears listing the selected devices.

Step 7 Click **Save**.

The Device Groups window reappears with the new device group listed.

Editing a Device Group

From the Edit Device Group window, you can modify the fields that have been specified for a particular device group.

To access the Edit Device Group window, follow these steps:

Step 1 Choose **Inventory > Physical Inventory > Device Groups**.

Step 2 Select a single device group to modify by checking the check box to the left of the Device Group Name.

Step 3 Click the **Edit** button. This button is only enabled if a device group is selected.

The Edit Device Group window appears.

Step 4 Enter the changes you want to make to the selected device group.

Step 5 Click **Save**.

The changes are saved and the Device Groups window reappears.

Deleting Device Groups

From the Delete window, you can remove selected device groups from the database.

To access the Delete window, follow these steps:

Step 1 Choose **Inventory > Physical Inventory > Device Groups.**

The Device Groups window appears.

Step 2 Select one or more device groups to delete by checking the check box(es) to the left of the Device Group Names.

Step 3 Click the **Delete** button. This button is enabled only if one or more device groups are selected.

The Confirm Delete window appears.

Step 4 Click the **Delete** button to confirm that you want to delete the device group(s) listed.

The Device Groups window reappears with the specified device group(s) deleted.

E-mailing a Device Group

From the E-mail window, you can send a device report via e-mail to the owners of specified device groups.

To access the E-mail window, follow these steps:

Step 1 Choose **Inventory > Physical Inventory > Device Groups**.

The Device Groups window appears.

Step 2 Select the device groups for which you want to send a device report by checking the check box to the left of the Device Group Name.

Step 3 Click the **E-mail** button. This button is only enabled if one or more device groups are selected.

The Send Mail to Device owners of selected groups window appears.

Step 4 Compose the e-mail that you want to send to the selected device group owners.

Step 5 Click **Send**.

The e-mail is sent and the Device Groups window reappears.

Ethernet Access Topology Information

This section covers the following topics:

Physical Rings

The Physical Rings displays the capability to create a two-node ring. You can create an NPC Ring with a minimum of two devices.

This section describes how you can create, edit, and delete Physical Rings. This section includes the following topics:

Creating Physical Rings

Rings with two devices has the option to add more devices to the same ring through add or edit option.

To create physical rings, follow these steps:

Step 1 Choose **Inventory > Logical Inventory > Physical Rings**.

The Physical Circuits window appears.

Step 2 Click the **Create** button.

The Create Ring window appears. A ring has a minimum of two physical links that form a ring.

Note At any time, if you click **Cancel**, everything you have chosen disappears.

Step 3 Start with the first line, which represents the first physical link.

Step 4 In the **Source Device** column, click **Select source device** and a Select Source Device - CPE/PE window appears.

Note The CPE you choose *must* be a Multi-VRF CE.

Step 5 Click a radio button next to the device to be the source device for this physical link and then click **Select**. The Create Ring window reappears with the chosen **Source Device**.

Note When choosing the **Source Device** for a physical link, this same choice is made for the **Destination Device** for the previous physical link (or the last physical link if you are choosing for the first physical link). For a selected device, do not select the same interface for the source and destination interface.

Step 6 In the **Source Interface** column in this new version of the new Create Ring window, click **Select source interface** and a Select Source Interface window appears with a list of interfaces.

Step 7 Click a radio button next to the interface to be the source interface for this physical link and then click **Select**. The Create Ring window reappears with the chosen **Source Interface**.

Step 8 In the **Destination Device** column in this new version of the Create Rind window, click **Select destination device** and a Select Source Device — CPE/PE window appears.

Step 9 Click a radio button next to the device to be the destination device for this physical link and then click **Select**.

The Create Ring window reappears with the chosen **Destination Device**.

Note When choosing the **Destination Device** for the a physical link, this same choice is made for the next **Source Device**. Do not choose the same Interface for these devices. The minimum number of devices that can participate in a ring is two.

Step 10 In the **Destination Interface** column in this new version of the Create Rind window, click **Select destination interface** and the Select Source Interface window appears with a list of interfaces.

Step 11 Click a radio button next to the interface to be the destination interface for this NPC and then click **Select**

The Create Ring window reappears with the chosen **Destination Interface**.

Step 12 Repeat Step 4 for the middle physical links and Step 4 to Step 7 for the last physical link.

Step 13 If you want to insert an extra physical link in the ring, check the check box for the line that represents the physical link you want the new physical link to follow and click **Insert**. Implement Step 4 to fill in the remaining entries in this new physical link.

Step 14 If you want to delete a physical link in the ring but a minimum of three physical links will remain, check the check box for the line that represents the physical link you want to delete and click **Delete**.

Step 15 If you want to establish additional cross links between non-adjacent devices in this ring, you can click **Edit Cross Links** in the Create Ring window, and you then view a new Create Ring window with no entry. Click the **Add** button and you can choose from the devices already in your ring. The result is a new entry in the Create Ring window with this device as the **Source Device**. Establish the Destination Device and Source and Destination Interfaces as you did when creating the ring. The choices of devices and interfaces is limited to those already established in your ring.

Note To **Edit Cross Links**, a minimum of four devices is needed to form this ring.

Step 16 Click **Cancel** if you do not want to save this information, and you will proceed to the previous window.

Otherwise, when you have completed setting up your ring click **Save**. The new ring is added in Physical Rings window, and a green check for Succeeded appears. The new ring is identified by the source device-source interface.

Step 17 To create a ring with more than three physical links, check the check box for the link in the Create Ring window to which you want to insert and the **Insert** button is then enabled. Proceed in adding links as explained in this section.

Editing Physical Rings

To edit physical rings, follow these steps:

Note If the specified Physical Ring is participating in any of the Named Physical Circuits, then you can not edit the ring. An error message appears containing IDs of the NPCs that contain the NPC Ring.

Step 1 Choose **Inventory > Logical Inventory > Physical Rings** and a window appears.

Step 2 Check the check box next to the line that represents an NPC ring and then click **Edit.**

The Create Ring window appears with all the data for this ring. Proceed as in the "Creating Physical Rings" section on page 2-30 to make any changes you want.

Step 3 When you have the ring as you want it, click **Save**. The Physical Rings window appears with the appropriate name (source device-source interface) and a green check for Succeeded appears.

Deleting Physical Rings

Rings with more than two nodes has the option to transition to a two device ring by removal of the device from the ring topology through edit or delete option.

To delete physical rings, follow these steps:

Note If the specified NPC Ring is participating in any of the Named Physical Circuits, then you can not delete the ring. An error message appears containing IDs of the NPCs that contain the NPC Ring.

Step 1 Choose **Inventory > Logical Inventory > Physical Rings** and a window appears.

Step 2 Check the check box(es) next to the line(s) that represent(s) NPC ring(s) that you want to delete and then click Delete.

Step 3 Click **Cancel** if you change your mind about deleting the chosen ring(s) or click **Delete** to actually delete the ring.

The Physical Rings window appears with the remaining ring names and a green check for Succeeded appears.

Named Physical Circuits

Named physical circuits (NPCs) are named circuits that describe a physical connection between a CPE or U-PE and an N-PE. The intermediate nodes of the NPCs can either be CPE or PE. They can be connected in a circular fashion forming a ring of devices, which is represented by an entity known as NPC Rings. NPC Rings represent the circular topology between devices (CPE or PE) to the Named Physical Circuits. To create an NPC, you must specify how the source CPE/U-PE and the destination N-PE are connected and specify the intermediate nodes.

The connectivity of the NPCs is defined by specifying a set of devices serving as physical links; each device has two interfaces that are part of the NPC connections. The Incoming Interface defines the interface from the CE direction. The Outgoing Interface defines the interface toward the PE direction.

You can also add (meaning after the chosen device) or insert (meaning before the chosen device) an NPC Ring in the link.

Keep in mind the following when you are creating an NPC:

- In the Prime Fulfillment software, the device you select can be any node in the link. The Prime Fulfillment software only shows the appropriate devices. The first device *must* be a CPE or U-PE and the last device *must* be an N-PE.
- NPCs should be created before the MPLS multi-device, VPLS, or L2VPN service request is created with cpe1 and pe1. So when you create the SR, you would select the policy, cpe1, pe1, and the NPC that defines the link between cpe1 and pe1.

This section describes how you can create and delete NPCs and create, edit, and delete NPC Rings. This section includes the following topics:

- Creating a Named Physical Circuit, page 2-33
- Deleting Named Physical Circuits, page 2-35

Creating a Named Physical Circuit

To add an NPC physical link, follow these steps:

Step 1 Choose **Inventory > Logical Inventory > Named Physical Circuit**.

The Named Physical Circuit window appears.

Step 2 Click the **Create** button.

The Create a Named Physical Circuits window appears.

Each line represents a physical link and each physical link contains the following attributes:

- **Device**

- **Incoming Interface**
- **Outgoing Interface**
- **Ring** (optional)

Note Before adding a ring in an NPC, create a ring and save it in the repository, as explained in the "Creating Physical Rings" section on page 2-30.

Note An NPC must have at least one link defined. The link must have two devices, an Incoming Interface, and an Outgoing Interface.

Step 3 Click **Add Device** or **Insert Device.**

The Select Device window appears.

Step 4 Be sure that the drop-down list in **Show** is **CPE or PE**.

Step 5 Click a radio button next to a device and then click **Select**. The Create a Named Physical Circuits window reappears with the chosen **Device**.

Step 6 If you want to add a device to your NPC as the last item or after the item checked in the check box, click the **Add Device** button in the Create a Named Physical Circuit window and then add device and interface information as explained in the previous steps. If you want to insert a device to your NPC as the first item or before the item checked in the check box, click the **Insert Device** button in the Create a Named Physical Circuit window and then add device and interface information as explained in the previous steps.

Step 7 In the **Outgoing Interface** column in this new version of the Create a Named Physical Circuit window, click **Select outgoing interface** and a window appears with a list of interfaces.

Step 8 Click a radio button next to the interface to be the source interface for this NPC and then click **Select**. The Create a Named Physical Circuit window reappears with the chosen **Interface**.

Step 9 In the **Incoming Interface** column in this new version of the Create a Named Physical Circuit window, click **Select incoming interface** and a window appears with a list of interfaces.

Step 10 Click a radio button next to the interface to be the incoming interface for this NPC and then click **Select**. The Create a Named Physical Circuit window reappears with the chosen **Incoming Interface**.

Step 11 If you created an NPC ring that you want to insert or add into this NPC, as explained in the "Creating Physical Rings" section on page 2-30, you can click **Insert Ring** or **Add Ring** and the ring appears at the beginning or before the item checked in the check box for **Insert Ring** or the ring appears at the end or after the item checked in the check box for **Add Ring**.

Note When inserting a ring, select the source device of the ring that connects to a source device or an NPC and the destination device of the ring that connects to the destination device of the NPC.

If you have not created an NPC ring that you want to insert into this NPC, proceed to Step 14.

Step 12 Click a radio button next to the ring you choose and then click **Select**. The Create a Named Physical Circuit window reappears with the chosen **Ring**.

Step 13 Select the missing devices and interfaces as explained in the "Creating Physical Rings" section on page 2-30.

Step 14 Click **Cancel** if you do not want to save this information, and you will proceed to the previous window.

Otherwise, click **Save**. The Create a Named Physical Circuit window reappears with the new NPC listed.

Deleting Named Physical Circuits

To delete NPC(s), follow these steps:

Step 1 Choose **Inventory > Logical Inventory > Named Physical Circuits.**

The Named Physical Circuits window appears.

Step 2 Select one or more NPCs to delete by checking the check box(es) on the left.

Step 3 Click the **Delete** button.

The Delete NPC window appears.

Note If the specified NPC is being used by any of the Service Requests, you will not be allowed to delete it. An error message appears explaining this.

Step 4 Click the **Delete** button to confirm that you want to delete the NPCs listed.

The Named Physical Circuits window reappears with the specified NPCs deleted.

Managing Customer Premise Devices

This section includes the following topics:

Customers

A customer site is a set of IP systems with mutual IP connectivity between them without the use of a VPN. Each customer site belongs to exactly one customer. A customer site can contain one or more (for load balancing) edge device routers. This section describes how to create, edit, and delete customers.

This section covers the following topics:

Creating a Customer

From the Create Customer window, you can create different customers.

To create a customer, follow these steps:

Step 1 Choose **Service Design > Resources > Customers**.

The Customers window appears.

Step 2 Click the **Create** button.

The Create Customer window appears.

Step 3 Enter the name and information for the Customer that you are creating. Check the **Site of Origin Enabled** check box if you want this enabled.

Step 4 Click **Cancel** if you do not want to save this information, and you will proceed to the previous window.

Otherwise, click **Save**. The changes are then saved and the Customers window reappears.

Editing a Customer

From the Edit Customer window, you can modify the fields that have been specified for a particular customer.

To access the Edit Customer window, follow these steps:

Step 1 Choose **Service Design > Resources > Customers**.

The Customers window appears.

Step 2 Select a single customer to modify by checking the check box to the left of the Customer Name.

Step 3 Click the **Edit** button. This button is only enabled if a customer is selected.

The Edit Customer window appears.

Step 4 Enter the changes you want to make to the selected customer.

Step 5 Click **Cancel** if you do not want to save this information, and you will proceed to the previous window.

Otherwise, click **Save**. The changes are then saved and the Customers window reappears.

Deleting Customers

From the Delete window, you can remove selected customers from the database.

To access the Delete window, follow these steps:

Step 1 Choose **Service Design > Resources > Customers.**

The Customers window appears.

Step 2 Select one or more customers to delete by checking the check box to the left of the Customer Name.

Step 3 Click the **Delete** button. This button is enabled only if one or more customers are selected.

The Confirm Delete window appears.

Step 4 Click **Cancel** if you do not want to save this information, and you will proceed to the previous window.

Otherwise, click **Delete** to confirm that you want to delete the customer(s) listed. The Customers window reappears with the specified customer(s) deleted.

Customer Sites

The Customer Sites window feature is used to create, edit, and delete customer sites.

This section covers the following topics:

Creating a Customer Site

From the Create Customer Sites window, you can create different customer sites.

To create a customer sites, follow these steps:

Step 1 Choose **Inventory > Resources > Customer Sites**.

The Customer Sites window appears.

Step 2 Click the **Create** button.

The Create New Customer Sites window appears.

Step 3 Enter the name and information for the Customer that you are creating. To enter the customer name follow these steps:

a. Click the **Select** button next to the Customer field.

A list of customer names appears.

b. Click the radio button next to customer name and then **Select**.

Step 4 Enter the Site Information.

Step 5 Click **Cancel** if you do not want to save this information, and you will proceed to the previous window.

Otherwise, click **Save**. The changes are then saved and the Customer Site window reappears.

Editing a Customer Site

From the Edit Customer Sites window, you can modify the fields that have been specified for a particular customer sites.

To access the Edit Customer Sites window, follow these steps:

Step 1 Choose **Inventory > Resources > Customer Sites**.

The Customer Sites window appears.

Step 2 Select a single site name to modify by checking the check box to the left of the Site Name.

Step 3 Click the **Edit** button. This button is only enabled if a customer is selected.

The Edit Customer window appears.

Step 4 Enter the changes you want to make to the selected customer site.

Step 5 Click **Cancel** if you do not want to save this information, and you will proceed to the previous window.

Otherwise, click **Save**. The changes are then saved and the Customer Site window reappears.

Deleting Customer Sites

From the Delete window, you can remove selected customer sites from the database.

To access the Delete window, follow these steps:

Step 1 Choose **Inventory > Resources > Customer Sites.**

The Customer Sites window appears.

Step 2 Select one or more customer sites to delete by checking the check box to the left of the Site Name.

Step 3 Click the **Delete** button. This button is enabled only if one or more customer sites are selected.

The Confirm Delete window appears.

Step 4 Click **Cancel** if you do not want to delete this information, and you will proceed to the previous window.

Otherwise, click **Delete** to confirm that you want to delete the customer site(s) listed. The Customer Sites window reappears with the specified customer site(s) deleted.

Customer Devices

The CPE feature provides a list of CPEs that have been associated with a site through the CPE editor or Inventory Manager.

This section covers the following topics:

- Create CPE Device, page 2-39
- Edit CPE Device, page 2-40
- Delete CPE Device, page 2-40

Choose **Inventory > Resources > Customer Devices,** the CPE Devices window appears.

The CPE Devices window contains the following:

- **Device Name**—Lists the names of devices. The first character must be a letter. Can contain letters, numbers, and these punctuation characters: period, underscore, and dash. Limit: 80 characters. You can sort the list by device name.
- **Customer Name**—Lists the names of customer. The first character must be a letter. Can contain letters, numbers, and these punctuation characters: period, underscore, and dash. Limit: 80 characters. You can sort the list by customer name.
- **Site Name**—Lists the names of sites. The first character must be a letter. Can contain letters, numbers, and these punctuation characters: period, underscore, and dash. Limit: 80 characters. You can sort the list by site name.
- **Management Type—**When associating a CE with a customer site, you can select Managed or Unmanaged. Other choices are available (see below), but they should not be confused with this primary choice.
  - Managed—A managed CE can be provisioned directly by the provider using Prime Fulfillment. The CE must be reachable from an Prime Fulfillment server.

- Unmanaged —An unmanaged CE cannot be provisioned directly by the provider. If Unmanaged is selected, the provider can use Prime Fulfillment to generate a configuration, and then send the configuration to the customer for placement on the CE.
- Managed - Management LAN —A managed Management LAN or Management CE (MCE) is configured like a managed CE router, but it resides in the provider space. Normally, an MCE acts as the network operations center (NOC) gateway router.
- Unmanaged - Management LAN —An unmanaged Management LAN or MCE is configured like an unmanaged CE router, but it resides in the provider space. Normally, an MCE acts as the network operations center (NOC) gateway router.
- Directly Connected —In most cases, the CE is connected to a PE router. In this case, the CE is connected to a workstation or other device.
- Directly Connected Management Host —In most cases, the CE is connected to a PE router. In this case, the CE is connected to a workstation or other device, on which Prime Fulfillment resides.
- Multi-VRF —A multi-VRF CE (MVRFCE) is owned by the customer, but resides in the provider space. It is used to offload traffic from the PE.
- Unmanaged Multi-VRF—An unmanaged multi-VRF CE is provisioned like an unmanaged CE (configurations are not uploaded or downloaded to the device by the provider). It is owned by the customer and resides in the provider space.

Create CPE Device

From the Create Customer Devices window, you can create different CPE devices.

To create a CPE device, follow these steps:

Step 1 Choose **Inventory > Resources > Customer Devices.**

The Customer Devices window appears.

Step 2 Click **Create** to create new CPE devices. Enabled only if no customer site is selected.

The Create New CPE Device window appears.

Step 3 Click **Select** for the required **Device Name** and **Site Name**.

For each, you receive a list of the devices and sites, respectively, from which you can choose one in each window and then click **Select**. Click **Cancel** if you do not want to save this information, and you will proceed to the previous window.

Note The Customer Name is displayed only if the customer site is created.

Step 4 The drop-down window for **Management Type** allows you choose the management type of the CPE device you are creating.

Step 5 Click **Cancel** if you do not want to save this information, and you will proceed to the previous window.

Otherwise, click **Save**. The changes are saved and the CPE Device window reappears.

Edit CPE Device

From the Edit Customer Device window, you can modify the fields that have specified for a particular CPE device.

To edit a CPE device, follow these steps:

Step 1 Choose **Inventory > Resources > Customer Devices**.

The Customer Devices window appears.

Step 2 Select a single device name to modify by checking the check box to the left of the Device Name.

Step 3 Click the **Edit** button. This button is only enabled if a device name is selected.

The Edit Customer window appears.

Step 4 Enter the changes you want to make to the selected CPE device.

Step 5 Click **Cancel** if you do not want to save this information, and you will proceed to the previous window.

Otherwise, click **Save**. The changes are then saved and the Customer Device window reappears.

Delete CPE Device

From the Delete window, you can remove selected customer device from the database.

To access the Delete window, follow these steps:

Step 1 Choose **Inventory > Resources > Customer Devices.**

The Customer Devices window appears.

Step 2 Select one or more device name to delete by checking the check box to the left of the Device Name.

Step 3 Click the **Delete** button. This button is enabled only if one or more device names are selected.

The Confirm Delete window appears.

Step 4 Click **Cancel** if you do not want to save this information, and you will proceed to the previous window.

Otherwise, click **Delete** to confirm that you want to delete the device name(s) listed. The Customer Devices window reappears with the specified device name(s) deleted.

Setting Up Resources

This section explains how to set up the resources. It contains the following sections:

Access Domains

To access the Access Domains window: Choose **Inventory > Resources > Access Domains.**

From the Access Domains window, you can create, edit, or delete access domains.

This sections covers the following topics:

Creating Access Domains

From the Create Access Domains window, you can create different access domain.

To create a access domain, follow these steps:

Step 1 Choose **Inventory > Resources > Access Domains**.

The Access Domains window appears.

Step 2 Click the **Create** button.

The Create New Access Domains window appears.

Step 3 Enter the access domain name. This is a required field.

Step 4 To enter the Provider follow these steps (this is a required field):

a. Click the **Select** button next to the Provider field.

A list of Provider Name window appears.

b. Click the radio button next to provider name and then **Select**.

Step 5 Enter the PEs information (required field). This information about the PE will be helpful to the access domain operators. Limited to 256 characters.

Step 6 Enter the Reserved VLAN information (this is optional).

Step 7 Click **Cancel** if you do not want to save this information, and you will proceed to the previous window.

Otherwise, click **Save**. The changes are then saved and the Access Domains window reappears.

Editing Access Domains

From the Edit Access Domains window, you can modify the fields that have been specified for a particular provider region.

To access the Edit Access Domains window, follow these steps:

Step 1 Choose **Inventory > Resources > Access Domains**.

The Access Domains window appears

Step 2 Select a single access domain to modify by checking the check box to the left of the Access Domains Name.

Step 3 Click the **Edit** button. This button is only enabled if a access domain name is selected.

The Edit Access Domains window appears.

Step 4 Enter the changes you want to make to the selected access domain.

Step 5 Click **Cancel** if you do not want to save this information, and you will proceed to the previous window.

Otherwise, click **Save**. The changes are then saved and the Access Domains window reappears.

Deleting Access Domains

From the Delete window, you can remove selected access domain from the database.

To access the Delete window, follow these steps:

Step 1 Choose **Inventory > Resources > Access Domains**.

The Access Domains window appears

Step 2 Select one or more access domain to delete by checking the check box to the left of the Access domain Names.

Step 3 Click the **Delete** button. This button is enabled only if one or more access domains are selected.

The Confirm Delete window appears.

Step 4 Click **Cancel** if you do not want to save this information, and you will proceed to the previous window.

Otherwise, click **Delete** to confirm that you want to delete the access domain(s) listed. The Access Domains window reappears with the specified access domain(s) deleted.

Interface Access Domains

To access the Interface Access Domains window: Choose **Inventory > Resources > Interface Access Domains.**

From the Access Domains window, you can create, edit, or delete access domains.

This sections covers the following topics:

Note Outer VLAN ID resource pools can be created once the Interface Access Domains is created.

Creating Interface Access Domains

From the Create Interface Access Domains window, you can create different interface access domains.

To create an interface access domain, follow these steps:

Step 1 Choose **Inventory > Resources > Interface Access Domains**.

The Interface Access Domains window appears.

Step 2 Click the **Create** button.

The Create New Interface Access Domains window appears.

Step 3 Enter the interface access domain name. This is a required field.

Step 4 To enter the Provider follow these steps (this is a required field):

a. Click the **Select** button next to the Provider field.

A list of Provider Name window appears.

b. Click the radio button next to provider name and then **Select**.

Step 5 Select the PE device (required field) from the list of Provider devices available for the selected Provider.

Step 6 Select the Interfaces (required field) from the interface pop-up window. Interface pop-up window displays all available EVC supported physical ports from the selected NPE device.

Note Single interface or group multiple interfaces can be selected based on the requirements.

Step 7 Click **Cancel** if you do not want to save this information, and you will proceed to the previous window.

Otherwise, click **Save**. The changes are then saved and the Interface Access Domains window reappears.

Editing Interface Access Domains

From the Edit Interface Access Domains window, you can modify the fields that have been specified for a particular provider region.

To access the Edit Interface Access Domains window, follow these steps:

Step 1 Choose **Inventory > Resources > Interface Access Domains**.

The Interface Access Domains window appears

Step 2 Select a single interface access domain to modify by checking the check box to the left of the Interface Access Domains Name.

Step 3 Click the **Edit** button. This button is only enabled if an interface access domain name is selected.

The Edit Access Domains window appears.

Step 4 Enter the changes you want to make to the selected interface access domain.

Step 5 Click **Cancel** if you do not want to save this information, and you will proceed to the previous window.

Otherwise, click **Save**. The changes are then saved and the Interface Access Domains window reappears.

Deleting Interface Access Domains

From the Delete window, you can remove selected access domain from the database.

To access the Delete window, follow these steps:

Step 1 Choose **Inventory > Resources > Interface Access Domains**.

The Interface Access Domains window appears.

Step 2 Select one or more access domain to delete by checking the check box to the left of the Interface Access Domain Names.

Step 3 Click the **Delete** button. This button is enabled only if one or more access domains are selected.

The Confirm Delete window appears.

Step 4 Click **Cancel** if you do not want to save this information, and you will proceed to the previous window.

Otherwise, click **Delete** to confirm that you want to delete the access domain(s) listed. The Interface Access Domains window reappears with the specified access domain(s) deleted.

Resource Pools

Cisco IP Solution Center enables multiple pools to be defined and used during operations. The following resource pools are available:

- IP address pool: The IP address pool can be defined and assigned to regions or VPNs. This feature gives the service operator the flexibility to manage the allocation of all IP addresses in the network.
- Multicast pool: The Multicast pool is used for Multicast MPLS VPNs.
- Route Target (RT) pool: A route target is the MPLS mechanism that informs PEs as to which routes should be inserted into the appropriate VRFs. Every VPN route is tagged with one or more route targets when it is exported from a VRF and offered to other VRFs. The route target can be considered a VPN identifier in MPLS VPN architecture. RTs are a 64-bit number.
- Route Distinguisher (RD) pool: The IP subnets advertised by the CE routers to the PE routers are augmented with a 64-bit prefix called a route distinguisher (RD) to make them unique. The resulting 96-bit addresses are then exchanged between the PEs, using a special address family of Multiprotocol BGP (referred to as MP-BGP). The RD pool is a pool of 64-bit RD values that Cisco IP Solution Center uses to make sure the IP addresses in the network are unique.
- Site of origin pool: The pool of values for the site-of-origin (SOO) attribute. The site-of-origin attribute prevents routing loops when a site is multihomed to the MPLS VPN backbone. This is achieved by identifying the site from which the route was learned, based on its SOO value, so that it is not readvertised back to that site from a PE in the MPLS VPN network.
- VC ID pool: VC ID pools are defined with a starting value and a size of the VC ID pool. (VC ID is a 32-bit unique identifier that identifies a circuit/port.) A given VC ID pool is not attached to any Inventory object. During the deployment of an Ethernet Service (EWS, ERS for example), VC ID is auto-allocated from the VC ID pool.
- VLAN ID pool: VLAN ID pools are defined with a starting value and a size of the VLAN pool. A given VLAN ID pool can be attached to an Access Domain. During the deployment an Ethernet Service (EWS, ERS for example), VLAN ID can be auto-allocated from the Access Domain's VLAN pools. This gives the Service Provider a tighter control of VLAN ID allocation.

All these resources, that are made available to the service provider, enable the automation of service deployment.

This section describes how you can create and manage pools for various types of resources. This section includes the following topics:

- Creating an IP Address Pool, page 2-45
- Creating a Multicast Pool, page 2-46

Creating an IP Address Pool

Prime Fulfillment uses IP address pools to automatically assign IP addresses to PEs and CEs. Each Region has an IP address pool to use for IP numbered addresses (/30 pools) and a separate IP address pool for IP unnumbered addresses (/32 loopback address pools).

Within a VPN or extranet, all IP addresses must be unique. Customer IP addresses must not overlap with the provider's IP addresses. Overlapping IP addresses are only possible when two devices cannot see each other—that is, when they are in isolated VPNs.

From the Create IP Address Pool window, you can create IP address pools.

To create an IP address pool, follow these steps:

Step 1 Choose **Service Design > Resources > Resource Pools**.

The Resource Pools window appears.

Step 2 Select **IPV4 Address** from the **Pool Type** in the upper left of the Resource Pools window.

Step 3 Click the **Create** button.

The Create New IP Address Resource Pool window appears.

The Create New IP Address Resource Pool window contains the following fields:

- **IP Address Pool** (required)—Text field in the format a.b.c.d/mask, for example 172.0.0.0/8.
- **Pool Mask (bits)** (required)—Choices include: **30** and **32**

  where:

  30 is used for IP numbered address pools (/30)

  32 is used for IP unnumbered loopback address pools (/32).
- **Pool Association** (required)—Choices include: **Region**, **VPN**, and **Customer** from the drop-down list. Then you can click the **Select** button to receive all selections for the choice you made in the drop-down list. From this new window, make your selection and click **Select**.

Note If you choose **VPN**, an additional optional field appears, **Pool Name Suffix**. This field allows the creation of multiple address pools within the same VPN. If you are creating this address pool for DMVPN usage, the recommendation is to use this field to specify a suffix.

- **Pool Name Suffix** (optional)—Suffixes are used to make a pool name unique. You can append this IP Address Pool to an existing pool by selecting a previously defined suffix, or click **New** to create a new pool.

Step 4 Enter the required information for the IP address pool you are creating.

Step 5 Click **Save**.

The Resource Pools window reappears with the new IP address pool listed.

Creating a Multicast Pool

From the Create Multicast Pool window, you can create multicast pools. These pools are global and are not associated with any provider or customer.

To create a multicast pool, follow these steps:

Step 1 Choose **Service Design > Resources > Resource Pools**.

The Resource Pools window appears.

Step 2 Select **Multicast** from the **Pool Type** in the upper left of the Resource Pools window.

Step 3 Click the **Create** button.

The Create New Multicast Resource Pool window appears.

The Create New Multicast Resource Pool window contains the following fields:

- **Multicast Address** (required)—Text field in the format **a.b.c.d/mask**, for example 239.0.0.0/8. Range: 224.0.1.0/8 to 239.255.255.255/32.
- **Use for default MDT** (optional)—This is a check box. Default: selected.
- **Use for Data MDT** (optional)—This is a check box. The *data MDT* contains a range of multicast group addresses and a bandwidth threshold. Thus, whenever a CE behind a multicast-VRF exceeds that bandwidth threshold while sending multicast traffic, the PE sets up a new data MDT for the multicast traffic from that source. The PE informs the other PEs about this data MDT and, if they have receivers for the corresponding group, the other PEs join this data MDT. Default: selected.

Step 4 Enter the required information for the multicast pool you are creating.

Step 5 Click **Save**.

The Resource Pools window reappears with the new multicast pool listed.

Creating a Route Distinguisher and Route Target Pool

MPLS-based VPNs employ Border Gateway Protocol (BGP) to communicate between PEs to facilitate customer routes. This is made possible through extensions to BGP that carry addresses other than IPv4 addresses. A notable extension is called the route distinguisher (RD).

The purpose of the route distinguisher (RD) is to make the prefix value unique across the network backbone. Prefixes should use the same RD if they are associated with the same set of route targets (RTs) and anything else that is used to select routing policy. The community of interest association is based on the route target (RT) extended community attributes distributed with the Network Layer Reachability Information (NLRI). The RD value must be a globally unique value to avoid conflict with other prefixes.

The MPLS label is part of a BGP routing update. The routing update also carries the addressing and reachability information. When the RD is unique across the MPLS VPN network, proper connectivity is established even if different customers use non-unique IP addresses.

For the RD, every CE that has the same overall role should use a VRF with the same name, same RD, and same RT values. The RDs and RTs are only for route exchange between the PEs running BGP. That is, for the PEs to do MPLS VPN work, they have to exchange routing information with more fields than usual for IPv4 routes; that extra information includes (but is not limited to) the RDs and RTs.

From the Create Route Distinguisher Pool window, you can create route distinguisher pools.

Create a Route Distinguisher Pool

To create a route distinguisher pool, follow these steps:

Step 1 Choose **Service Design > Resources > Resource Pools**.

The Resource Pools window appears.

Step 2 Select **Route Distinguisher** from the **Pool Type** in the upper left of the Resource Pools window.

Step 3 Click the **Create** button.

The Create New Route Distinguisher Resource Pool window appears.

The Create New Route Distinguisher Resource Pool window contains the following fields:

- **RD Pool Start** (required)—Range: 0 to 2147483646.
- **RD Pool Size** (required)—Range: 1 to 2147483647.
- **Provider** (required)

Step 4 Enter the **RD Pool Start** and **Size** information for the route distinguisher pool you are creating.

Step 5 Click the **Select** button.

The Provider for new Resource Pool window appears.

Step 6 Select one of the providers listed and click **Select**.

Step 7 Click **Save**.

The Resource Pools window reappears with the new route distinguisher pool listed.

Create a Route Target Pool

To create a Route Target Pool, follow these steps:

Step 1 Choose **Service Design > Resources > Resource Pools**.

The Resource Pools window appears.

Step 2 Select **Route Target** from the **Pool Type** in the upper left of the Resource Pools window.

Step 3 Click the **Create** button.

The Create New Route Target Resource Pool window appears.

The Create New Route Target Resource Pool window contains the following fields:

- **RT Pool Start** (required)—Range: 0 to 2147483646.
- **RT Pool Size** (required)—Range: 1 to 2147483647.
- **Provider** (required)

Step 4 Enter the **RT Pool Start** and **Size** information for the route target pool you are creating.

Step 5 Click the **Select** button.

The Provider for new Resource Pool window appears.

Step 6 Select one of the providers listed and click **Select**.

Step 7 Click **Save**.

The Resource Pools window reappears with the new route target pool listed.

Creating a Site of Origin Pool

In MPLS VPN, CE sites use private/public AS numbers and when one AS number is used for each VPN, all sites belonging to the same VPN share the same private/public AS number. The default BGP behavior is to drop any prefix if its own AS number is already in the AS path. As a result, a customer site does not learn prefixes of a remote site in this situation. AS-OVERRIDE must be configured (if there are hub sites involved, ALLOWAS-IN must be configured) to allow those prefixes to be sent by PE routers but a routing loop can occur.

For example, CE1 and CE2 belong to the same customer VPN and have the same AS number 65001. The AS path between two customer sites is 65001 - 1234 - 65001 and prefixes cannot be exchanged between customer sites because AS 65001 is already in the path. To solve this problem, AS-OVERRIDE options are configured on PE routers; but it introduces a routing loop into the network without using extended community site of origin attributes.

Site of origin is a concept in MPLS VPN architecture that prevents routing loops in sites that are multi-homed to the MPLS VPN backbone and in sites using AS-OVERRIDE in conjunction. Site of origin is a type of BGP extended community attribute used to identify a prefix that originated from a site so that the re-advertisement of that prefix back to the site can be prevented. This attribute uniquely identifies the site from which the PE router learned the route. Site of origin is tagged at PE in peering with BGP neighbors using an inbound route-map and works in conjunction with BGP CE-PE routing protocol.

Site of origin must be unique per customer site per VPN/customer (when these sites are multi-homed). Therefore, the same value of site of origin must be used on PE routers connected to the same CE router or to the same customer site.

Note Each time a customer site is created, Prime Fulfillment generates a unique site of origin value from the selected site of origin provider pool if Site of Origin is enabled. This site of origin value must be unique per customer site per customer/VPN.

From the Create Site of Origin Pool window, you can create site of origin pools.

To create a site of origin pool, follow these steps:

Step 1 Choose **Service Design > Resources > Resource Pools**.

The Resource Pools window appears.

Step 2 Select **Site of Origin** from the **Pool Type** in the upper left of the Resource Pools window.

Step 3 Click the **Create** button.

The Create New Site of Origin Resource Pool window appears.

The Create New Site of Origin Resource Pool window contains the following fields:

- **SOO Pool Start** (required)—Range: 0 to 2147483646.
- **SOO Pool Size** (required)—Range: 1 to 2147483647.
- **Provider** (required)

Step 4 Enter the **SOO Pool Start** and **Size** information for the site of origin pool you are creating.

Step 5 Click the **Select** button.

The Provider for new Resource Pool window appears.

Step 6 Select one of the providers listed and click **Select**.

Step 7 Click **Save**.

The Site of Origin pools window reappears with the new route target pool listed.

Creating a VC ID Pool

From the Create VC ID Pool window, you can create VC ID pools. These pools are global and are not associated with any provider or customer

To create a VC ID pool, follow these steps:

Step 1 Choose **Service Design > Resources > Resource Pools**.

The Resource Pools window appears.

Step 2 Select **VC ID** from the **Pool Type** in the upper left of the Resource Pools window.

Step 3 Click the **Create** button.

The Create New VC ID Resource Pool window appears.

The Create New VC ID Resource Pool window contains the following fields:

- **VC Pool Start** (required)—Range: 1 to 2147483646.
- **VC Pool Size** (required)—Range: 1 to 2147483647.

Step 4 Enter the required information for the site of origin pool you are creating.

Step 5 Click **Save**.

The VC ID Pools window reappears with the new VC ID pool listed.

Creating a VLAN Pool

From the Create VLAN Pool window, you can create VLAN pools.

To create a VLAN pool, follow these steps:

Step 1 Choose **Service Design > Resources > Resource Pools**.

The Resource Pools window appears.

Step 2 Select **VLAN** from the **Pool Type** in the upper left of the Resource Pools window.

Step 3 Click the **Create** button.

The Create New VLAN Resource Pool window appears.

The Create New VLAN Resource Pool window contains the following fields:

- **VLAN Pool Start** (required)— Range: 1 to 4094.
- **VLAN Pool Size** (required)—Range: 1 to 4094.
- **Access Domain** (required)

Step 4 Enter the **VLAN Pool Start** and **Size** information for the VLAN pool you are creating.

Step 5 Click the **Select** button.

The Access Domain for new VLAN Pool window appears.

Step 6 Select one of the access domains listed and click **Select**.

Step 7 Click **Save**.

The VLAN Pools window reappears with the new VLAN pool listed.

Creating an EVC Outer VLAN Pool

From the Create EVC OUTER VLAN Pool window, you can create EVC OUTER VLAN pools.

To create an OUTER VLAN pool, follow these steps:

Step 1 Choose **Service Design > Resources > Resource Pools**.

The Resource Pools window appears.

Step 2 Select **EVC OUTER VLAN** from the **Pool Type** in the upper left of the Resource Pools window.

Step 3 Click the **Create** button.

The Create New OUTER VLAN Resource Pool window appears.

The Create New OUTER VLAN Resource Pool window contains the following fields:

- **OUTER VLAN Pool Start** (required)— Range: 1 to 4094.
- **OUTER VLAN Pool Size** (required)—Range: 1 to 4094.
- **Interface Access Domain** (required)

Step 4 Enter the **OUTER VLAN Pool Start** and **Size** information for the OUTER VLAN pool you are creating.

Step 5 Click the **Select** button.

The Interface Access Domain for new OUTER VLAN Pool window appears.

Step 6 Select one of the interface access domains listed and click **Select**.

Step 7 Click **Save**.

The OUTER VLAN Pools window reappears with the new OUTER VLAN pool listed.

Deleting Resource Pools

From the Resource Pool window, you can delete specific resource pools.

To delete resource pools, follow these steps:

Step 1 Choose **Service Design > Resources > Resource Pools**.

The Resource Pools window appears.

Step 2 Select a pool type from the **Pool Type** in the upper left of the Resource Pools window.

Step 3 Select one or more resource pools to delete by checking the check box(es) to the left of the resource pool(s).

Step 4 Click the **Delete** button.

A Confirm Delete window appears.

Step 5 Click the new **Delete** button to confirm that you want to delete the resource pool(s) listed.

The Resource Pools window reappears with the specified pool(s) deleted.

Route Targets

A VPN can be organized into subsets called *Route Targets*. A Route Target describes how the CEs in a VPN communicate with each other. Thus, Route Targets describe the logical topology of the VPN. Cisco Prime Fulfillment can be employed to form a variety of VPN topologies between CEs by building hub and spoke or full mesh CE routing communities. Route Targets are building blocks that allow you to form complex VPN topologies and CE connectivity.

The most common types of VPNs are *hub-and-spoke* and *full mesh*.

- A hub-and-spoke Route Target is one in which one or a few CEs act as hubs, and all spoke CEs talk only to or through the hubs, never directly to each other.
- A full mesh Route Target is one in which every CE connects to every other CE.

These two basic types of VPNs—full mesh and hub and spoke—can be represented with a single Route Target. Whenever you create a VPN, the Prime Fulfillment software creates one default Route Target for you. This means that until you need advanced customer layout methods, you will not need to define new Route Targets. Up to that point, you can think of a Route Target as standing for the VPN itself—they are one and the same. If, for any reason, you must override the software's choice of route target values, you can do so only at the time you create a Route Target in the Prime Fulfillment software.

To build very complex topologies, it is necessary to break down the required connectivity between CEs into groups, where each group is either fully meshed, or has a hub and spoke pattern. (Note that a CE can be in more than one group at a time, if each group has one of the two basic patterns.) Each subgroup in the VPN wants its own Route Target. Any CE that is only in one group just joins the corresponding Route Target (as a spoke if necessary). If a CE is in more than one group, then you can use the Advanced Setup choice during provisioning to add the CE to all the relevant groups in one service request. Given this information, the provisioning software does the rest, assigning route target values and VRF tables to arrange exactly the connectivity the customer requires. You can use the Topology tool to double-check the Route Target memberships and resultant VPN connectedness.

Prime Fulfillment supports multiple CEs per site and multiple sites connected to the same PE. Each Route Target has unique route targets (RT), route distinguisher (RD), and VPN Routing and Forwarding instance (VRF) naming. After provisioning a Route Target, it is a good idea to run the audit reports to verify the Route Target deployment and view the topologies created by the service requests. The product supports linking two or more CE routing communities in the same VPN.

This section describes how you can create and manage CE routing communities. This section includes the following topics:

- Creating Route Targets, page 2-52
- Deleting Route Targets, page 2-53

Creating Route Targets

When you create a VPN, the Prime Fulfillment software creates one default Route Target for you. But if your network topology and configuration require customized Route Target definitions, you can define Route Targets customized for your network.

Tip Customized Route Targets should be defined only in consultation with the VPN network administrator. To build complex topologies, it is necessary to break down the required connectivity between CEs into groups, where each group is either fully meshed or has a hub-and-spoke pattern. A CE can be in more than one group at a time, as long as each group has one of the two basic configuration patterns.

Each subgroup in the VPN wants its own Route Target. Any CE that is only in one group just joins the corresponding Route Target (as a spoke if necessary). If a CE is in more than one group, then you can use the Advanced Setup choice during provisioning to add the CE to all the relevant groups in one service request. Given this information, Cisco IP Solution Center does the rest, assigning route target values and VRF tables to arrange the precise connectivity the customer requires.

To create a CE routing community, follow these steps:

Step 1 Choose **Service Design > Resources > Route Targets**.

The Route Targets window appears.

Step 2 Click **Create**.

The Create CE Routing Community window appears.

Step 3 Complete the Route Target fields as required for the CE Routing Community:

a. **Provider Name** (required)—To specify the service provider associated with this Route Target, click **Select**.

   The Select Provider window appears.

b. From this new window, choose the name of the service provider, then click **Select**.

c. **Name** (required)—Enter the name of the Route Target.

d. **Route Target Type**—Specify the Route Target type: Hub and Spoke or Fully Meshed.

e. **Auto-Pick Route Target Values**—Choose to either let Cisco IP Solution Center automatically set the route target (RT) values or set the RT values manually.

   By default, the **Auto-pick route target values** check box is checked. If you uncheck the check box, you can enter the Route Target values manually.

Caution If you choose to bypass the **Auto-pick route target values** option and set the route target (RT) values manually, note that the RT values cannot be edited after they have been defined in the Prime Fulfillment software.

Step 4 When you have finished entering the information in the Create CE Routing Community window, click **Save**.

After creating the Route Target, you can add it to the VPN.

Deleting Route Targets

From the CE Routing Community window, you can delete specific Route Targets.

To delete Route Target(s), follow these steps:

Step 1 Choose **Service Design > Resources > Route Targets.**

The Route Targets window appears.

Step 2 Select Route Target(s) to delete by checking the check box(es) to the left of the Route Target name.

Step 3 Click the **Delete** button.

The Confirm Delete window appears.

Step 4 Click **OK** to confirm that you want to delete the Route Target(s) listed.

The Route Targets window reappears with the specified Route Target(s) deleted.

Setting Up Logical Inventory

VPNs

At its simplest, a virtual private network (VPN) is a collection of sites that share the same routing table. A VPN is also a framework that provides private IP networking over a public infrastructure such as the Internet. In Cisco IP Solution Center: MPLS VPN Management, a VPN is a set of customer sites that are configured to communicate through a VPN service. A VPN is defined by a set of administrative policies.

A VPN is a network in which two sites can communicate over the provider's network in a private manner; that is, no site outside the VPN can intercept their packets or inject new packets. The provider network is configured such that only one VPN's packets can be transmitted through that VPN—that is, no data can come in or out of the VPN unless it is specifically configured to allow it. There is a physical connection from the provider edge network to the customer edge network, so authentication in the conventional sense is not required.

This section describes how you can create and manage pools for various types of resources. This section includes the following topics:

- Creating a VPN, page 2-53
- Deleting VPNs, page 2-56

Creating a VPN

To create a VPN, follow these steps:

Step 1 Choose **Inventory > Logical Inventory > VPN**.

The VPNs window appears.

Step 2 Click **Create**.

The Create VPN window appears.

Step 3 Complete the fields as required for the VPN:

a. **Name** (required)—Enter the name of the VPN, any name of your choice.

b. **Customer** (required)—To select the customer associated with this VPN, choose **Select**.

c. From the list of customers, select the appropriate customer, then click **Select**.

d. If you want MPLS attributes, complete the fields in the MPLS Attributes section of the window. For VPLS, skip to step **w.**

e. **Create Default Route Targets** (optional)—To create a default Route Targets, check the **Create Default Route Targets** check box and select a provider.

f. **Enable Unique Route Distinguisher**—The BGP Multipath Load Sharing for Both eBGP and iBGP in an MPLS VPN feature is enabled only under the IPv4 VRF address family configuration mode. When enabled, this feature can perform load balancing on eBGP and/or iBGP paths that are imported into the VRF.

g. **Enable IPv4 Multicast** —To enable multicast IPv4 VPN routing, check the **Enable IPv4 Multicast** check box.

An IP address that starts with the binary prefix *1110* is identified as a *multicast group address*. There can be more than one sender and receiver at any time for a given multicast group address. The senders send their data by setting the group address as the destination IP address. It is the responsibility of the network to deliver this data to all the receivers in the network who are listening to that group address.

Note Before you can create a VPN with multicast enabled, you must define one or more multicast resource pools.

h. **Enable IPv6 Multicast** —To enable multicast IPv6 VPN routing, check the **Enable IPv6 Multicast** check box.

An IP address that starts with the binary prefix *1110* is identified as a *multicast group address*. There can be more than one sender and receiver at any time for a given multicast group address. The senders send their data by setting the group address as the destination IP address. It is the responsibility of the network to deliver this data to all the receivers in the network who are listening to that group address.

Note Before you can create a VPN with multicast enabled, you must define one or more multicast resource pools.

i. **Enable Auto Pick MDT Addresses** (optional)—Check this check box to use **Default MDT Address** and **Default MDT Subnet** values from a multicast resource pool.

j. **Default MDT Address**—If **Enable Auto Pick MDT Addresses** is set on, **Default MDT Address** is required.

k. **Data MDT Subnet** (optional)—If **Enable Auto Pick MDT Addresses** is not checked (set on), you can provide the **Default MDT Subnet**.

l. **Data MDT Size** (optional)—If **Enable Multicast** is set on, **Data MDT Size** is required. From the drop-down list, select the data MDT size.

MDT refers to a *multicast distribution tree* (MDT). The MDT defined here carries multicast traffic from customer sites associated with the multicast domain.

m. **Data MDT Threshold** (optional)—If **Enable Multicast** is set on, **Data MDT Threshold** is required. Enter the bandwidth threshold for the data multicast distribution tree.

   The *data MDT* contains a range of multicast group addresses and a bandwidth threshold. Thus, whenever a CE behind a multicast-VRF exceeds that bandwidth threshold while sending multicast traffic, the PE sets up a new data MDT for the multicast traffic from that source. The PE informs the other PEs about this data MDT and, if they have receivers for the corresponding group, the other PEs join this data MDT.

n. **Default PIM Mode** (optional)—For Default Protocol Independent Multicast (PIM) mode, click the drop-down list and choose **SPARSE_MODE** or **SPARSE_DENSE_MODE**. For IOS XR devices, no configlet is generated for either mode.

o. **Enable PIM SSM** (optional)—Check this check box for PIM Source Specific Multicast (SSM).

p. **SSM List Name** (optional)—Choose **DEFAULT** from the drop-down list and you create the following CLI: **ip pim vpn** <*vpnName*> **ssm default**. No configlet is generated for IOS XR devices, because they are using the standard SSM range 232.0.0.0/8. Choose **RANGE** from the drop-down list to associate an access-list number or a named access-list with the SSM configuration. This creates the following CLI: **ip pim vpn** <*vpnName*> **ssm range {ACL#!named-ACL-name}**.

q. **Multicast Route Limit** (optional)—Enter a valid value of 1 to 2147483647. For IOS XR devices, no configlet is generated.

r. **Enable Auto RP Listener** (optional)—Check this check box to enable the Rendezvous Point (RP) listener function. By default, this feature is running on IOS XR devices and no configlet is generated for this attribute.

s. **Configure Static-RP** (optional)—To configure Static RPs, check the associated check box. The Edit option for **PIM Static-RPs** then goes active.

t. **PIM Static-RPs**—To edit or add PIM Static-RPs, click **Edit**. The Edit PIM Static RPs window appears. Then click **OK.**

u. **Route Targets** (optional)—If **Enable Multicast** is set on, **Route Targets** is required. If you do not choose to enable the default Route Target, you can select a customized Route Target that you have already created in Prime Fulfillment. From the Route Targets pane, click **Select**.

   The Select Route Targets window appears.

v. Check the check box for the Route Target you want used for this service policy, then click **Select**.

   You return to the Create VPN window, where the new Route Target selection is displayed, along with its hub route target (HRT) and spoke route target (SRT) values.

w. If you want VPLS attributes, the optional fields for that are in x. to aa.

x. **Enable VPLS** (optional)—Check this check box to enable VPLS.

y. **VPLS VPN ID** (optional)—Enter an integer in the range of 1 to 2147483646.

z. **Service Type** (optional)—Click the drop-down list and choose from **ERS** (Ethernet Relay Service) or **EWS** (Ethernet Wire Service).

aa. **Topology** (optional)—Choose the VPLS topology from the drop-down list: **Full Mesh** (each CE has direct connections to every other CE) or **Hub and Spoke** (only the Hub CE has connection to each Spoke CE and the Spoke CEs do not have direct connection to each other).

Step 4 When you are satisfied with the settings for this VPN, click **Save**.

You have successfully created a VPN, as shown in the **Status** display in the lower left corner of the VPNs window.

Deleting VPNs

From the VPNs window, you can delete specific VPNs.

Note Only VPNs not associated with MPLS service requests can be deleted.

To delete VPN(s), follow these steps:

Step 1 Choose **Inventory > Logical Inventory > VPN**.

The VPNs window appears.

Step 2 Select VPN(s) to delete by checking the check box(es) to the left of the VPN name.

Step 3 Click the **Delete** button.

The Confirm Delete window appears.

Step 4 Click **OK** to confirm that you want to delete the VPN(s) listed.

The VPNs window reappears with the specified VPN(s) deleted.

CHAPTER 3

Managing L2VPN and Carrier Ethernet Services

This chapter describes how to use Prime Fulfillment policies and service requests to manage various L2VPN and Carrier Ethernet services. It contains the following sections:

Getting Started with L2VPN Services

This section provides a road map to help you get started using the L2VPN component in Cisco Prime Fulfillment 6.2. It contains the following sections:

Overview

Before you can use the L2VPN component to provision Layer 2 services, you must complete several installation and configuration steps, as outlined in this section. In addition, you should be familiar with basic concepts for Prime Fulfillment and L2VPN services. The following subsections provide a summary of the key tasks you must accomplish to be able to provision L2VPN, VPLS and EVC services using Prime Fulfillment. You can use the information in this section as a checklist. Where appropriate, references to other sections in this guide or to other guides in the Prime Fulfillment documentation set are provided. See the referenced documentation for more detailed information. After the basic installation and configuration steps are completed for both Prime Fulfillment and the L2VPN component, see the subsequent sections to create and provision L2VPN, VPLS and EVC services.

Installing Prime Fulfillment and Configuring the Network

Before you can use the L2VPN module in Prime Fulfillment to provision L2VPN or VPLS services, you must first install Prime Fulfillment and do the basic network configuration required to support Prime Fulfillment. Details on these steps are provided in Chapter 2, "Before Setting Up Prime Fulfillment." See that chapter for information about Prime Fulfillment installation and general network configuration requirements.

Note To use the L2VPN component within Prime Fulfillment, you must purchase and activate the L2VPN license.

Configuring the Network to Support Layer 2 Services

In addition to basic network configuration required for Prime Fulfillment, you must perform the following network configuration steps to support Layer 2 services. Information on doing these steps is not provided in the Prime Fulfillment documentation. See the documentation for your devices for information on how to perform these steps.

1. Enable MPLS on the core-facing interfaces of the N-PE devices attached to the provider core.
2. Set up /32 loopback addresses on N-PE devices. These loopback addresses should be the termination of the LDP connection(s).
3. Set all Layer 2 devices (switches) to VTP transparent mode. This ensures that none of the switches will operate as VLAN servers and will prevent VLAN information from automatically propagating through the network.

Setting Up Basic Prime Fulfillment Services

After the basic network configuration tasks are completed to support Prime Fulfillment and L2 services, you use Prime Fulfillment to define elements in the Prime Fulfillment repository, such as providers and regions, customers and sites, devices, VLAN and VC pools, NPCs, and other resources that are necessary to provision L2 services. Detailed steps to perform general Prime Fulfillment tasks are covered in Chapter 2, "Before Setting Up Prime Fulfillment." You can also find a summary of some important Prime Fulfillment set up tasks in Setting Up the Prime Fulfillment Services, page 3-5. The information below is a checklist of basic Prime Fulfillment services you must set up before provisioning L2 services.

Setting Up Providers, Customers, and Devices

Perform the following steps to set up providers, customers, and devices in the Prime Fulfillment repository. These are global resources that can be used by all Prime Fulfillment services.

1. **Set up service providers and regions.** The region is important because a single provider could have multiple networks. The region is used as a further level of differentiation to allow for such circumstances. To create a provider and a region, see Setting Up Resources, page 2-40. See also Defining a Service Provider and Its Regions, page 3-8.
2. **Set up customers and customer sites.** A customer is a requestor of a VPN service from an ISP. Each customer can own many customer sites. Each customer site belongs to one and only one Customer and can own many CEs. For detailed steps to create customers and sites, see Setting Up Resources, page 2-40. See also Defining Customers and Their Sites, page 3-8.
3. **Import or add raw devices.** Every network element that Prime Fulfillment manages must be defined as a device in the Prime Fulfillment repository. An element is any device from which Prime Fulfillment can collect information. In most cases, devices are Cisco IOS routers and switches. You can set up devices in Prime Fulfillment manually, through autodiscovery, or through importing device configuration files. For detailed steps for importing, adding, and collecting configurations for devices, see Appendix G, "Inventory - Discovery." See also Using Autodiscovery for L2 Services, page 3-167.
4. **Assign devices roles as PE or CE.** After devices are created in Prime Fulfillment, you must define them as customer (CE) or provider (PE) devices. You do this by editing the device attributes on individual devices or in batch editing through the Prime Fulfillment inventory manager. To set device attributes, see Setting Up Devices and Device Groups, page 2-1.

Setting Up the N-PE Loopback Address

Within Prime Fulfillment, you must set the loopback address on the N-PE device(s). For details about this procedure, see Setting Up the N-PE Loopback Address, page 3-3.

Setting Up Prime Fulfillment Resources for L2VPN and VPLS Services

Some Prime Fulfillment resources, such as access domains, VLAN pools, and VC pools are set up to support Prime Fulfillment L2VPN and VPLS services only. To set up these resources, perform the following steps.

1. **Create access domain(s).** For L2VPN and VPLS, you create an access domain if you provision an Ethernet-based service and want Prime Fulfillment to automatically assign a VLAN for the link from the VLAN pool. For each Layer 2 access domain, you need a corresponding access domain object in Prime Fulfillment. During creation, you select all the N-PE devices that are associated with this domain. Later, one VLAN pool can be created for an access domain. For detailed steps to create access domains, see Setting Up Resources, page 2-40. See also Creating Access Domains, page 3-8.
2. **Create VLAN pool(s).** A VLAN pool is created for each access domain. For L2VPN and VPLS, you create a VLAN pool so that Prime Fulfillment can assign a VLAN to the links. VLAN ID pools are defined with a starting value and a size. For detailed steps to create VLAN pools, see Setting Up Resources, page 2-40. See also Creating VLAN Pools, page 3-9.
3. **Create VC pool(s).**VC ID pools are defined with a starting value and a size of the VC ID pool. A given VC ID pool is not attached to any inventory object (a provider or customer). Create one VC ID pool per network. For detailed steps to create VC pools, see Setting Up Resources, page 2-40. See also Creating a VC ID Pool, page 3-10.

Setting Up NPCs

Before creating an L2VPN or VPLS service request, you must predefine the physical links between CEs and PEs or between U-PEs and N-PEs. The Named Physical Circuit (NPC) represents a link going through a group of physical ports. Thus, more than one logical link can be provisioned on the same NPC. Therefore, the NPC is defined once but used by several L2VPN or VPLS service requests. For detailed steps to create NPCs, see Setting Up Logical Inventory, page 2-53. See also Creating Named Physical Circuits, page 3-11.

Setting Up VPNs

You must define VPNs before provisioning L2VPN or VPLS services. In L2VPN, one VPN can be shared by different service types. In VPLS, one VPN is required for each VPLS instance. To define VPNs, see Setting Up Logical Inventory, page 2-53. See also Defining VPNs, page 3-8.

Working with EVC, L2VPN, and VPLS Policies and Service Requests

After you have set up providers, customers, devices, and resources in Prime Fulfillment, you are ready to create EVC, L2VPN, or VPLS policies, provision service requests (SRs), and deploy the services. After the service requests are deployed you can monitor, audit and run reports on them. All of these tasks are covered in this guide. To accomplish these tasks, perform the following steps.

1. **Review overview information about L2 services concepts.** See the chapter “Prime Fulfillment Layer 2 VPN Concepts” in the *Cisco Prime Fulfillment Theory of Operations Guide 6.2*.
2. **Set up an EVC, L2VPN, or VPLS policy.** See the appropriate section, depending on the type of policy you want to create:
   - Creating an EVC Ethernet Policy, page 3-18
   - Creating an EVC ATM-Ethernet Interworking Policy, page 3-54
   - Creating an L2VPN Policy, page 3-90
   - Creating a VPLS Policy, page 3-130
3. **Provision the EVC, L2VPN, or VPLS service request.** See the appropriate section, depending on the type service request you want to provision:
   - Managing an EVC Ethernet Service Request, page 3-34
   - Managing an EVC ATM-Ethernet Interworking Service Request, page 3-69
   - Managing an L2VPN Service Request, page 3-118
   - Managing a VPLS Service Request, page 3-158
4. **Deploy the service request.** See Deploying, Monitoring, and Auditing Service Requests, page 3-166.
5. **Check the status of deployed services.** You can use one or more of the following methods:
   - Monitor service requests. See Deploying, Monitoring, and Auditing Service Requests, page 3-166.
   - Audit service requests. See Deploying, Monitoring, and Auditing Service Requests, page 3-166.
   - Run L2 and VPLS reports. See Generating L2 and VPLS Reports, page 10-32.

A Note on Terminology Conventions

The Prime Fulfillment GUI and this chapter of the user guide use specific naming conventions for Ethernet services. These align closely with the early MEF conventions. This is expected to be updated in future releases of to conform with current MEF conventions. For reference, the equivalent terms used by the MEF forum are summarized in Table 3-1.

See the chapter "Prime Fulfillment Layer 2 VPN Concepts," in the *Cisco Prime Fulfillment Theory of Operations Guide 6.2*, for more information on terminology conventions and how these align with underlying network technologies.

Table 3-1 Ethernet Service Terminology Mappings

Term Used in GUI and This User Guide	Current MEF Equivalent Term
L2VPN over MPLS Core	
Ethernet Wire Service (EWS)	Ethernet Private Line (EPL)
Ethernet Relay Service (ERS)	Ethernet Virtual Private Line (EVPL)
ATM over MPLS (ATMoMPLS)	—
Frame Relay over MPLS (FRoMPLS)	—
VPLS Over MPLS Core	
Ethernet Wire Service (EWS) or Ethernet Multipoint Service (EMS)	Ethernet Private LAN (EP-LAN)
Ethernet Relay Service (ERS) or Ethernet Relay Multipoint Service (ERMS)	Ethernet Virtual Private LAN (EVP-LAN)
VPLS over Ethernet Core	
Ethernet Wire Service (EWS)	Ethernet Private LAN (EP-LAN)
Ethernet Relay Service (ERS)	Ethernet Virtual Private LAN (EVP-LAN)

Setting Up the Prime Fulfillment Services

To create L2VPN, VPLS, and EVC policies and service requests, you must first define the service-related elements, such as target devices, VPNs, and network links. Normally, you create these elements once.

This section contains the basic steps to set up the Cisco Prime Fulfillment 6.2 resources for L2VPN services. It contains the following sections:

- Creating a VC ID Pool, page 3-10
- Creating Named Physical Circuits, page 3-11
- Creating and Modifying Pseudowire Classes, page 3-14
- Defining L2VPN Group Names for IOS XR Devices, page 3-18

Note This section presents high-level information on Prime Fulfillment services that are relevant to L2VPN. For more detailed information on setting up these and other basic Prime Fulfillment services, see Chapter 2, "Before Setting Up Prime Fulfillment."

Creating Target Devices and Assigning Roles (N-PE or U-PE)

Every network element that Prime Fulfillment manages must be defined as a device in the system. An element is any device from which Prime Fulfillment can collect information. In most cases, devices are Cisco IOS routers that function as N-PE, U-PE, or P. For detailed steps to create devices, see Setting Up Devices and Device Groups, page 2-1.

Configuring Device Settings to Support Prime Fulfillment

Two device settings must be configured to support the use of Prime Fulfillment in the network:

- Switches in the network must be operating in VTP transparent mode.
- Loopback addresses must be set on N-PE devices.

Note These are the two minimum device settings required for Prime Fulfillment to function properly in the network. You must, of course, perform other device configuration steps for the proper functioning of the devices in the network.

Configuring Switches in VTP Transparent Mode

For security reasons, Prime Fulfillment requires VTPs to be configured in transparent mode on all the switches involved in ERS or EWS services before provisioning L2VPN service requests. To set the VTP mode, enter the following Cisco IOS commands:

```
Switch# configure terminal
Switch(config)# vtp mode transparent
```

Enter the following Cisco IOS command to verify that the VTP mode has changed to transparent:

```
Switch# Show vtp status
```

Setting the Loopback Addresses on N-PE Devices

The loopback address for the N-PE has to be properly configured for an Any Transport over MPLS (AToMPLS) connection. The IP address specified in the loopback interface must be reachable from the remote pairing PE. The label distribution protocol (LDP) tunnels are established between the two loopback interfaces of the PE pair. To set the PE loopback address, perform the following steps.

Step 1 Choose **Inventory > Resources > Provider Devices**.

The Provider Devices window appears.

Step 2 Choose a specific PE device and click the **Edit** button.

The Edit Provider Device window appears.

To prevent a wrong loopback address being entered into the system, the Loopback IP Address field on the GUI is read-only.

Step 3 Choose the loopback address by clicking the **Select** button (in the Loopback IP Address attribute).

The Select Device Interface window appears.

Step 4 Choose one of the loopback addresses listed in the Interface Name column.

This step ensures that you choose only a valid loopback address defined on the device.

Step 5 To further narrow the search, you can check the **LDPTermination Only** check box and click the **Select** button.

This limits the list to the LDP-terminating loopback interface(s).

Setting Up Devices for IOS XR Support

L2VPN in Cisco Prime Fulfillment 6.2, supports devices running Cisco's IOS XR software. IOS XR, a new member of the Cisco IOS family, is a unique self-healing and self-defending operating system designed for always-on operation while scaling system capacity up to 92Tbps. In L2VPN, IOS XR is only supported on Cisco XR12000 and CRS-1 series routers functioning as network provider edge (N-PE) devices.

In L2VPN, the following E-line services are supported for IOS XR:

- Point-to-point ERS with or without a CE.
- Point-to-point EWS with or without a CE.

The following L2VPN features are not supported for IOS XR:

- Standard UNI port on an N-PE running IOS XR. (The attribute **Standard UNI Port** in the Link Attributes window is disabled when the UNI is on an N-PE device running IOS XR.)
- SVI interfaces on N-PEs running IOS XR. (The attribute **N-PE Pseudo-wire On SVI** in the Link Attributes window is disabled for IOS XR devices.)
- Pseudowire tunnel selection. (The attribute **PW Tunnel Selection** in the Link Attributes window is disabled for IOS XR devices.)
- EWS UNI (dot1q tunnel or Q-in-Q) on an N-PE running IOS XR.
- Frame Relay/ATM and VPLS services.

To enable IOS XR support in L2VPN, perform the following steps.

Step 1 Set the DCPL property Provisioning\Service\l2vpn\platform\CISCO_ROUTER\IosXRConfigType to XML.

Possible values are CLI, CLI_XML, and XML (the default).

Step 2 Create the device in Prime Fulfillment as an IOS XR device, as follows:

a. Create the Cisco device by choosing **Inventory > Devices > Create Cisco Devcie**.

b. Choose **Cisco Device** in the drop-down list.

The Create Cisco Router window appears.

c. Set the **OS** attribute, located under Device and Configuration Access Information, to IOS_XR.

Note For additional information on setting DCPL properties and creating Cisco devices, see Appendix B, "Property Settings."

Step 3 Create and deploy L2VPN service requests, following the procedures in this guide.

Sample configlets for IOS XR devices are provided in Sample Configlets, page 3-176.

Defining a Service Provider and Its Regions

You must define the service provider administrative domain before provisioning L2VPN. The provider administrative domain is the administrative domain of an ISP with one BGP autonomous system (AS) number. The network owned by the provider administrative domain is called the backbone network. If an ISP has two AS numbers, you must define it as two provider administrative domains. Each provider administrative domain can own many region objects.

For detailed steps to define the provider administrative domain, see Setting Up Resources, page 2-40.

Defining Customers and Their Sites

You must define customers and their sites before provisioning L2VPN. A customer is a requestor of a VPN service from an ISP. Each customer can own many customer sites. Each customer site belongs to one and only one Customer and can own many CPEs. For detailed steps to create customers, see Setting Up Resources, page 2-40.

Defining VPNs

You must define VPNs before provisioning L2VPN or VPLS. In L2VPN, one VPN can be shared by different service types. In VPLS, one VPN is required for each VPLS instance. For detailed steps to create VPNs, see Setting Up Logical Inventory, page 2-53.

Note The VPN in L2VPN is only a name used to group all the L2VPN links. It has no intrinsic meaning as it does for MPLS VPN.

Creating Access Domains

For L2VPN and VPLS, you create an Access Domain if you provision an Ethernet-based service and want Prime Fulfillment to automatically assign a VLAN for the link from the VLAN pool.

For each Layer 2 access domain, you need a corresponding Access Domain object in Prime Fulfillment. During creation, you select all the N-PE devices that are associated with this domain. Later, one VLAN pool can be created for an Access Domain. This is how N-PEs are automatically assigned a VLAN.

Before you begin, be sure that you:

- Know the name of the access domain that you want to create.
- Have created a service provider to associate with the new access domain.
- Have created a provider region associated with your provider and PE devices.
- Have created PE devices to associate with the new access domain.
- Know the starting value and size of each VLAN to associate with the new access domain.
- Know which VLAN will serve as the management VLAN.

For detailed steps on creating Access Domains, see Setting Up Resources, page 2-40.

Creating VLAN Pools

For L2VPN and VPLS, you create a VLAN pool so that Prime Fulfillment can assign a VLAN to the links. VLAN ID pools are defined with a starting value and a size of the VLAN pool. A VLAN pool can be attached to an access domain. During the deployment of an Ethernet service, VLAN IDs can be autoallocated from the access domain's pre-existing VLAN pools. When you deploy a new service, Prime Fulfillment changes the status of the VLAN pool from Available to Allocated. Autoallocation gives the service provider tighter control of VLAN ID allocation.

You can also allocate VLAN IDs manually.

Note When you are setting a manual VLAN ID on a Prime Fulfillment service, Prime Fulfillment warns you if the VLAN ID is outside the valid range of the defined VLAN pool. If so, Prime Fulfillment does not include the manually defined VLAN ID in the VLAN pool. We recommend that you preset the range of the VLAN pool to include the range of any VLAN IDs that you manually assign.

Create one VLAN pool per access domain. Within that VLAN pool, you can define multiple ranges.

Before you begin, be sure that you:

- Know each VLAN pool start number.
- Know each VLAN pool size.
- Have created an access domain for the VLAN pool.
- Know the name of the access domain to which each VLAN pool will be allocated.

To have Prime Fulfillment automatically assign a VLAN to the links, perform the following steps.

Step 1 Choose **Service Design > Resources > Resource Pools**.

The Resource Pools window appears.

Step 2 Choose **VLAN** from the **Pool Type** drop-down list.

Step 3 Click **Create**.

The Create New VLAN Resource Pool window appears.

Step 4 Enter a VLAN Pool Start number.

Step 5 Enter a VLAN Pool Size number.

Step 6 If the correct access domain is not showing in the Access Domain field, click **Select** to the right of Access Domain field.

The Select Access Domain dialog box appears.

If the correct access domain is showing, continue with Step 9.

a. Choose an Access Domain Name by clicking the button in the Select column to the left of that Access Domain.

b. Click **Select**. The updated Create New VLAN Resource Pool window appears.

Step 7 Click **Save**.

The updated VLAN Resource Pool window appears.

Note The pool name is created automatically, using a combination of the provider name and the access domain name.

Note The Status field reads "Allocated" if you already filled in the Reserved VLANs information when you created the access domain. If you did not fill in the Reserved VLANs information when you created the access domain, the Status field reads "Available." To allocate a VLAN pool, you must fill in the corresponding VLAN information by editing the access domain. (See Creating Access Domains, page 3-8.) The VLAN pool status automatically sets to "Allocated" on the Resource Pools window when you save your work.

Step 8 Repeat this procedure for each range you want to define within the VLAN.

Creating Outer VLAN Pools

An outer VLAN pool is used in conjunction with the AutoPick Outer VLAN attribute in EVC Ethernet and EVC ATM-Etherner policies. For instructions on how to set up outer VLAN pools, see the section Resource Pools, page 2-44.

Creating a VC ID Pool

VC ID pools are defined with a starting value and a size of the VC ID pool. A given VC ID pool is not attached to any inventory object (a provider or customer). During deployment of an L2VPN or VPLS service, the VC ID can be autoallocated from the same VC ID pool or you can set it manually.

Note When you are setting a manual VC ID on a Prime Fulfillment service, Prime Fulfillment warns you if the VC ID is outside the valid range of the defined VC ID pool. If so, Prime Fulfillment does not include the manually defined VC ID in the VC ID pool. We recommend that you preset the range of the VC ID pool to include the range of any VC IDs that you manually assign.

Create one VC ID pool per network.

In a VPLS instance, all N-PE routers use the same VC ID for establishing emulated Virtual Circuits (VCs). The VC-ID is also called the VPN ID in the context of the VPLS VPN. (Multiple attachment circuits must be joined by the provider core in a VPLS instance. The provider core must simulate a virtual bridge that connects the multiple attachment circuits. To simulate this virtual bridge, all N-PE routers participating in a VPLS instance form emulated VCs among them.)

Note VC ID is a 32-bit unique identifier that identifies a circuit/port.

Before you begin, be sure that you have the following information for each VC ID pool you must create:

- The VC Pool start number
- The VC Pool size

For all L2VPN and VPLS services, perform the following steps.

Step 1 Choose **Service Design > Resources > Resource Pools**.

The Resource Pools window appears.

Step 2 Choose **VC ID** from the **Pool Type** drop-down list.

Because this pool is a global pool, it is not associated with any other object.

Step 3 Click **Create**.

The Create New VC ID Resource Pool window appears.

Step 4 Enter a VC pool start number.

Step 5 Enter a VC pool size number.

Step 6 Click **Save**.

The updated Resource Pools window appears.

Creating Named Physical Circuits

Before creating an L2VPN or VPLS service request, you must predefine the physical links between CEs and PEs. The Named Physical Circuit (NPC) represents a link going through a group of physical ports. Thus, more than one logical link can be provisioned on the same NPC; therefore, the NPC is defined once but used during several L2VPN or VPLS service request creations.

There are two ways to create the NPC links:

- Through an NPC GUI editor. For details on how to do this, see Creating NPCs Through the NPC GUI Editor, page 3-12.
- Through the autodiscovery process. For details on how to do this, see Creating NPC Links Through the Autodiscovery Process, page 3-14.

An NPC definition must observe the following creation rules:

- An NPC must begin with a CE or an up-link of the device where UNI resides or a Ring.
- An NPC must end with an N-PE or a ring that ends in an N-PE.

If you are inserting NPC information for a link between a CE and UNI, you enter the information as:

- Source Device is the CE device.

- Source Interface is the CE port connecting to UNI.
- Destination Device is the UNI box.
- Destination interface is the UNI port.

If you are inserting NPC information for a CE not present case, you enter the information as:

- Source Device is the UNI box.
- Source Interface is the UP-LINK port, not the UNI port, on the UNI box connecting to the N-PE or another U-PE or PE-AGG.
- Destination Device is the U-PE, PE-AGG, or N-PE.
- Destination Interface is the DOWN-LINK port connecting to the N-PE or another U-PE or PE-AGG.

If you have a single N-PE and no CE (no U-PE and no CE), you do not have to create an NPC since there is no physical link that needs to be presented.

If an NPC involves two or more links (three or more devices), for example, it connects ence11, enpe1, and enpe12, you can construct this NPC as follows:

- Build the link that connects two ends: mlce1 and mlpe4.
- Insert a device (enpe12) to the link you just made.

Creating NPCs Through the NPC GUI Editor

To create NPCs through the NPC GUI editor, perform the following steps.

Step 1 Choose **Inventory > Logical Inventory > Named Physical Circuits**.

The Named Physical Circuits window appears.

To create a new NPC, you choose a CE as the beginning of the link and a N-PE as the end. If more than two devices are in a link, you can add or insert more devices (or a ring) to the NPC.

Note The new device or ring added is always placed after the device selected, while a new device or ring inserted is placed before the device selected.

Each line on the Point-to-Point Editor represents a physical link. Each physical link has five attributes:

- **Source Device**
- **Source Interface**
- **Destination Device** (must be an N-PE)
- **Destination Interface**
- **Ring**

Note Before adding or inserting a ring in an NPC, you must create a ring and save it in the repository. To obtain information on creating NPC rings, see Setting Up Logical Inventory, page 2-53.

Source Device is the beginning of the link and **Destination Device** is the end of the link.

Step 2 Click **Create**.

The Create Named Physical Circuits window appears.

Step 3 Click **Add Device**.

The Select a Device window appears.

Step 4 Choose a CE as the beginning of the link.

Step 5 Click **Select**.

The device appears in the Create a Named Physical Circuits window.

Step 6 To insert another device or a ring, click **Insert Device** or **Insert Ring**.

To add another device or ring to the NPC, click **Add Device** or **Add Ring**. For this example, click **Add Device** to add the N-PE.

Step 7 Choose a PE as the destination device.

Step 8 Click **Select**.

The device appears.

Step 9 In the Outgoing Interface column, click **Select outgoing interface**.

A list of interfaces defined for the device appears.

Step 10 Choose an interface from the list and click **Select.**

Step 11 Click **Save**.

The Create Named Physical Circuits window now displays the NPC that you created.

Creating a Ring-Only NPC

To create an NPC that contains only a ring without specifying a CE, perform the following steps.

Step 1 Choose **Inventory > Logical Inventory > Named Physical Circuits**.

Step 2 Click **Create**.

The Create Named Physical Circuits window appears.

Step 3 Click **Add Ring**.

The Select NPC Ring window appears.

Step 4 Choose a ring and click **Select**. The ring appears.

Step 5 Click the **Select device** link to select the beginning of the ring.

A window appears showing a list of devices.

Step 6 Choose the device that is the beginning of the ring and click **Select**.

Step 7 Click the **Select device** link to choose the end of the ring.

Step 8 Choose the device that is the end of the ring and click **Select**.

Note The device that is the end of the ring in a ring-only NPC must be an N-PE.

Step 9 The Named Physical Circuits window appears showing the Ring-Only NPC.

Step 10 Click **Save** to save the NPC to the repository.

Terminating an Access Ring on Two N-PEs

Prime Fulfillment supports device-level redundancy in the service topology to provide a failover in case one access link should drop. This is accomplished through a special use of an NPC ring that allows an access link to terminate at two different N-PE devices. The N-PEs in the ring are connected by a logical link using loopback interfaces on the N-PEs. The redundant link starts from a U-PE device and may, optionally, include PE-AGG devices.

For details on how to implement this in Prime Fulfillment, see Appendix E, "Terminating an Access Ring on Two N-PEs."

Creating NPC Links Through the Autodiscovery Process

With autodiscovery, the existing connectivity of network devices can be automatically retrieved and stored in the Prime Fulfillment database. NPCs are further abstracted from the discovered connectivity.

For detailed steps to create NPCs using autodiscovery, see Setting Up Logical Inventory, page 2-53.

Creating and Modifying Pseudowire Classes

The pseudowire class feature provides you with the capability to configure various attributes associated with a pseudowire that is deployed as part of an L2VPN service request on IOS XR-capable devices.

Note The pseudowire class feature is supported for IOS XR 3.6.1 and higher.

The pseudowire class feature supports configuration of the encapsulation, transport mode, fallback options, and selection of a traffic engineering tunnel down which the pseudowire can be directed. For tunnel selection, you can select the tunnel using the Prime Fulfillment Traffic Engineering Management (TEM) application, if it is being used. Otherwise, you can specify the identifier of a tunnel that is already provisioned within the network. For IOS XR-capable devices, the pseudowire class is a separately defined object in the Prime Fulfillment repository, which can be attached to an L2VPN service policy or service request. The pseudowire class feature is only available for use in L2VPN ERS, EWS and ATM policies and service requests.

This section describes how to create and modify pseudowire classes. For information on how the pseudowire class is associated to a L2VPN policy and used within a service request, see Creating an L2VPN Policy, page 3-90. and Managing an L2VPN Service Request, page 3-118.

Creating a Pseudowire Class

To create a pseudowire class, perform the following steps.

Step 1 Choose **Inventory > Logical Inventory > Pseudowire Class**.

The Pseudowire Class window appears.

Step 2 Click the **Create** button.

The Create Pseudowire Class window appears.

Step 3 In the **Name** field, enter a valid PseudoWireClass name.

The pseudowire class name is used for provisioning **pw-class** commands on the IOS XR device. The name should not exceed 32 characters and should not contain spaces.

Step 4 In the **Description** field, enter a meaningful description of less than 128 characters.

This field is optional.

Step 5 Choose the **MPLS** encapsulation type from the **Encapsulation** drop-down list.

Note Currently, the only encapsulation type supported is MPLS.

Step 6 Choose the transport mode from the **TransportMode** drop-down list. The choices are:

- **NONE** (default)
- **Vlan**
- **Ethernet**

Note If you want to set the TransportMode to Vlan, we recommend you do this via a pseudowire class, if supported by the version of IOS XR being used. If pseudowire class is not supported in a particular version of IOS XR, then you must set the TransportMode using a Dynamic Component Properties Library (DCPL) property, as explained in the section Configuring the Transport Mode When Pseudowire Classes are Not Supported, page 3-17.

Step 7 Choose the protocol from the **Protocol** drop-down list. The choices are:

- **NONE** (default)
- **LDP—**Configures LDP as the signaling protocol for this pseudowire class.

Step 8 To configures sequencing on receive or transmit, choose a selection from the **Sequencing** drop-down list. The choices are:

- **NONE** (default)
- **BOTH—**Configures sequencing on receive and transmit.
- **TRANSMIT—**Configures sequencing on transmit.
- **RECEIVE—**Configures sequencing on receive.

Step 9 Enter a **Tunnel ID** of a TE tunnel that has already been provisioned by Prime Fulfillment or that has been manually provisioned on the device.

This value is optional. You can also select a TE tunnel that has already been provisioned by Prime Fulfillment, as covered in the next step.

Step 10 Click **Select TE Tunnel** if you want to select a TE tunnel that has been previously provisioned by Prime Fulfillment.

The Select TE Tunnel pop-up window appears. Choose a TE tunnel and click **Select**. This populates the TE Tunnel field with the ID of the selected TE tunnel.

Note After a TE tunnel is associated to a pseudowire class or provisioned in a service request, you will receive an error message if you try to delete the TE tunnel using the Traffic Engineering Management (TEM) application. TE tunnels associated with a pseudowire class or service request cannot be deleted.

Step 11 Check the **Disable Fallback** check box to disable the fallback option for the pseudowire tunnel.

Choose this option based on your version of IOS XR. It is required for IOS XR 3.6.1 and optional for IOS XR 3.7 and above.

Modifying a Pseudowire Class

This section describes how to modify (edit) an existing pseudowire class and how the editing operation might impact L2VPN service requests.

To modify a pseudowire class, perform the following steps.

Step 1 Choose **Inventory > Logical Inventory > Pseudowire Class**.

The Pseudowire Class window appears.

Step 2 Select the pseudowire class object you want to modify, and click **Edit**.

The PseudoWire Class Edit window appears.

Step 3 Make the desired changes and click **Save**.

Note The Name field is not editable if the pseudowire class is associated with any service requests.

If the pseudowire class being modified is associated with any L2VPN service requests, the Affected Jobs window appears, which displays a list of affected service requests

Note A list of affected service requests only appears if the Transport Mode, Tunnel ID, or Disable Fallback values are changed in the pseudowire class being modified.

Step 4 Click **Save** to update service requests associated with the modified pseudowire class.

The impacted service requests are moved to the Requested state.

Step 5 Click **Save and Deploy** to update and deploy service requests associated with the modified pseudowire class.

Deployment tasks are created for the impacted service requests that were previously in the Deployed state.

Step 6 Click **Cancel** to discard changes made to the modified pseudowire class.

In this case, no change of state occurs for any service requests associated with the pseudowire class.

Deleting a Pseudowire Class

To delete a Pseudowire class, follow these steps:

Note A PseudoWire Class that is in use with a service request or policy cannot be deleted.

Step 1 Choose **Inventory > Logical Inventory > PseudoWire Class**.

The Pseudowire Classes window appears.

Step 2 Check the check box(es) next to the pseudowire class(es) you want to delete.

Step 3 Click the **Delete** button and a window appears with the selected pseudowire class name.

Step 4 Click the **Delete** button to confirm that you want to delete the specified pseudowire class(es).

Step 5 Click **Cancel** if you want to return without deleting the selected pseudowire class(es).

Configuring the Transport Mode When Pseudowire Classes are Not Supported

This section describes how to configure the pseudowire transport mode to be of type Vlan for versions of IOS XR that do not support pseudowire classes. This is done through setting a Dynamic Component Properties Library (DCPL) property. See the usage notes following the steps for additional information.

Perform the following steps.

Step 1 In Prime Fulfillment, navigate to **Administration > Control Center > Hosts**.

Step 2 Check a check box for a specific host and click the **Config** button.

Step 3 Navigate to the DCPL property **Services\Common\pseudoWireVlanMode**.

Step 4 Set the property to **true**.

Step 5 Click **Set Property**.

Prime Fulfillment then generates VLAN transport mode configuration for the pseudowire.

Usage notes:

- To set the transport mode to Vlan, it is recommended that you do this via a pseudowire class, if supported by the version of IOS XR being used. If the pseudowire class feature is not supported, then the transport mode must be set using a DCPL property, as explained in the steps of this section
- The DCPL property pseudoWireVlanMode only sets the default value for PseudoWireClass TransportMode as Vlan if the DCPL property is set to true. Users can always over ride it.
- The DCPL property pseudoWireVlanMode acts in a dual way:
  - It sets a default value for PseudoWireClass TransportMode to Vlan.
  - In the absence of a pseudowire class, it generates a deprecated command **transport-mode vlan**. The **transport-mode vlan** command is a deprecated command in IOS XR 3.6 and later. Thus, when a pseudowire class is selected for an IOS XR device and the DCPL property is also set to true, the **transport-mode vlan** command is not generated. Pseudowire class and the **transport-mode vlan** command do not co-exist. If a pseudowire class is present, it takes precedence over the deprecated **transport-mode vlan** command.
- The value of the DCPL property pseudoWireVlanMode should not be changed during the life of a service request.

Defining L2VPN Group Names for IOS XR Devices

This section describes how to specify the available L2VPN group names for policies and service requests for IOS XR devices. The choices appear in a drop-down list of the L2VPN Group Name attribute in policies and service requests. The name chosen is used for provisioning the L2VPN group name on IOS XR devices. The choices are defined through setting a Dynamic Component Properties Library (DCPL) property.

Perform the following steps.

Step 1 In Prime Fulfillment, navigate to **Administration > Control Center > Hosts**.

Step 2 Check a check box for a specific host and click the **Config** button.

Step 3 Navigate to the DCPL property **Services\Common\l2vpnGroupNameOptions**.

Step 4 Enter a comma-separated list of L2VPN group names in the **New Value** field.

Step 5 Click **Set Property**.

Creating an EVC Ethernet Policy

This section contains an overview of EVC support in Cisco Prime Fulfillment 6.2, as well as the basic steps to create an EVC Ethernet policy. It contains the following subsections:

- Defining the EVC Ethernet Policy, page 3-18
- Setting the Service Options, page 3-20
- Setting the EVC Attributes, page 3-22
- Setting the Interface Attributes, page 3-28
- Enabling Template Association, page 3-33

For information on creating EVC Ethernet service requests, see Managing an EVC Ethernet Service Request, page 3-34.

Note For a general overview of EVC support in Prime Fulfillment, see the chapter “Layer 2 Concepts” in the *Cisco Prime Fulfillment Theory of Operations Guide 6.2*.

Note For Ethernet (E-Line and E-LAN) services, use of the EVC policy and service request is recommended. If you are provisioning services using the EVC syntax, or plan to do so in the future, use the EVC service. Existing services that have been provisioned using the L2VPN and VPLS service policy types are still supported and can be maintained with those service types. For ATM and FRoMPLS services, use the L2VPN service policy, as before.

Defining the EVC Ethernet Policy

You must define an EVC Ethernet policy before you can provision a service. A policy can be shared by one or more service requests that have similar service requirements.

A policy is a template of most of the parameters needed to define an EVC service request. After you define it, an EVC policy can be used by all the EVC service requests that share a common set of characteristics. You create a new EVC policy whenever you create a new type of service or a service with different parameters. EVC policy creation is normally performed by experienced network engineers.

An Editable check box in for an attribute in the policy gives the network operator the option of making a field editable. If the value is set to editable, the service request creator can change the value(s) of the particular policy attribute. If the value is *not* set to editable, the service request creator cannot change the attribute.

You can also associate Prime Fulfillment templates and data files with a policy. See Chapter 9, "Managing Templates and Data Files" for more about using templates and data files in policies.

It is also possible to create user-defined attributes within a policy (and service requests based on the policy). For background information on how to use the additional information feature, see Appendix H, "Adding Additional Information to Services."

To define an EVC Ethernet policy, you start by setting the service type attributes. To do this, perform the following steps.

Step 1 Choose **Service Design > Policies > Policy Manager**.

The Policy Manager window appears.

Step 2 Click **Create**.

The Policy Editor window appears.

Step 3 Choose **EVC** from the Policy Type drop-down list.

The Policy Editor window appears.

Step 4 Enter a **Policy Name** for the EVC policy.

Step 5 Choose the **Policy Owner** for the EVC policy.

There are three types of EVC policy ownership:

- Customer ownership
- Provider ownership
- Global ownership—Any service operator can make use of this policy.

This ownership has relevance when the Prime Fulfillment Role-Based Access Control (RBAC) comes into play. For example, an EVC policy that is customer-owned can only be seen by operators who are allowed to work on this customer-owned policy. Similarly, operators who are allowed to work on a provider's network can view, use, and deploy a particular provider-owned policy.

Step 6 Click **Select** to choose the owner of the EVC policy.

The policy owner was established when you created customers or providers during Prime Fulfillment setup. If the ownership is global, the Select function does not appear.

Step 7 Choose the **Policy Type**.

The choices are:

- **ETHERNET**
- **ATM-Ethernet Interworking**

Note This section describes creating the ETHERNET policy type. For information on using the EVC ATM-Ethernet Interworking policy type, see Creating an EVC ATM-Ethernet Interworking Policy, page 3-54.

Step 8 Click **Next**.

The Service Options window appears.

Step 9 Continue with the steps contained in the next section, Setting the Service Options, page 3-20.

Setting the Service Options

This section describes how to set the service options for the EVC Ethernet policy,

To set the EVC service options, perform the following steps.

Step 1 Check the **CE Directly Connected to EVC** check box if the CEs are directly connected to the N-PE.

This check box is not checked by default.

Note The **Editable** check box gives you the option of making a field editable. If you check the **Editable** check box, the service operator who is using this EVC policy can modify the editable parameter during EVC service request creation.

Usage notes:

- If the check box is checked, a service request created using this policy can have only directly connected links. No Ethernet access nodes will be involved.
- If the check box is unchecked, a service request created using this policy might or might not have Ethernet access nodes in the links.
- When a CE is directly connected to the N-PE, NPCs are not applicable to the link while creating service requests.
- When a CE is not directly connected to the N-PE, NPCs are used during service request creation, as per standard Prime Fulfillment behavior. There is no change in NPC implementation to support EVC functionality.

Step 2 Check the **All Links Terminate on EVC** check box if all links need to be configured with EVC features.

This check box is not check by default. Usage notes:

- If the check box is checked, a service request created using such policy will have all links using the EVC feature.
- If the check box is unchecked, zero or more links can use the EVC feature. This ensures that existing platforms can still be used in one or more links while delivering the services. This allows the possibility of a link with EVC support being added in the future.

Note If the check box is unchecked, in the service request creation process the user must indicate whether or not the created link is EVC or non-EVC.

- If no links are expected to use the EVC feature even in the future (for example, if the provider is not planning to upgrade to the EVC infrastructure for the service that is being created), existing Prime Fulfillment policy types (L2VPN or VPLS) can be used instead of EVC.

Step 3 Choose an **MPLS Core Connectivity Type** from the drop-down list.

Note The core option supports MPLS only. There is no L2TPv3 support for this service.

The choices are:

- **PSEUDOWIRE**—Choose this option to allow connectivity between two N-PEs across the MPLS core. This option does not limit the service to point-to-point (E-Line). This is because even with the PSEUDOWIRE option selected, there can still be multiple CEs connected to a bridge domain on one or both sides of the pseudowire.
- **LOCAL**—Choose this option for local connect cases in which there is no connectivity required across the MPLS core.

  Local connect supports the following scenarios:

  – All interfaces on the N-PE are EVC-capable and using the EVC infrastructure. This is configured by associating all of the customer traffic on these interfaces to a bridge domain. This consumes a VLAN ID on the N-PE (equal to the bridge domain ID).

  – Some interfaces on the N-PE are EVC-capable, while others are switch-port-based. In such cases, all of the customer traffic on the interfaces that are configured with the EVC infrastructure are associated to a bridge domain. The traffic on the non-EVC interfaces (and all the access nodes/interfaces beyond this N-PE) are configured with the Service Provider VLAN ID, where the Service Provider VLAN ID is the same as the bridge domain ID for the EVC-based services.

  – Only two interfaces on the N-PE are involved, and both are based on EVC-capable line cards. In the first case, the operator might choose not to configure the bridge domain option. In this case, the **connect** command that is used for the local connects are used, and the global VLAN is conserved on the device. If the operator chooses to configure with the bridge domain option, both interfaces are associated to a bridge domain ID, so that additional local links can be added to the service in future. This consumes a VLAN ID (bridge domain ID) on the N-PE.

- **VPLS**—Choose this option to allow connectivity between multiple N-PEs across the MPLS core.

  There is no limit on the number of N-PEs across the MPLS core within a service request. However, many service requests can refer to the same customer-associated VPN.

Note Attributes available in subsequent windows of the policy workflow dynamically change based on the choice made for the MPLS Core Connectivity Type (PSEUDOWIRE, LOCAL, or VPLS). For completeness, all attributes available for the different core types are documented in the following steps. Attributes apply to all core types, unless otherwise noted.

Note Also, some attributes are supported only on IOS or IOS XR platforms. Attributes apply to both platforms, unless otherwise noted. All platform-specific attributes are visible in the policy workflow windows. Later, when a service request is created based on the policy (and specific devices are associated with the service request), platform-specific attributes are filtered from service request windows, depending on the device type (IOS or IOS XR).

Step 4 Check the **Configure With Bridge Domain** check box to determine bridge domain characteristics.

The behavior of the Configure With Bridge-Domain option works in tandem with the choice you selected in the MPLS Core Connectivity Type option, as follows.

- **PSEUDOWIRE** as the MPLS Core Connectivity Type. There are two cases:

  A. With EVC:

  - If **Configure With Bridge Domain** is checked, the policy configures pseudowires under SVIs associated to the bridge domain.
  - If **Configure With Bridge Domain** is unchecked, the policy will configure pseudowires directly under the service instance. This conserves the global VLAN.

  B. Without EVC:

  - If **Configure With Bridge Domain** is checked, the policy configures pseudowires as in L2VPN services (with SVIs).
  - If **Configure With Bridge Domain** is unchecked, the policy configures pseudowires directly under subinterfaces.

  Only pseudowires can be either configured directly under service instance of the corresponding EVC-capable interface or under SVIs associated to the bridge domain.
- **LOCAL** as the MPLS Core Connectivity Type:
  - If **Configure With Bridge Domain** is checked, the policy allows either point-to-point or multipoint local connect services.
  - If **Configure With Bridge Domain** is unchecked, Prime Fulfillment allows only point-to-point local connects without bridge domain.
- **VPLS—Configure With Bridge Domain** is checked by default and non-editable.

Step 5 Click **Next**.

The EVC Attributes window appears.

Step 6 Continue with the steps contained in the next section, Setting the EVC Attributes, page 3-22.

Setting the EVC Attributes

This section describes how to set the EVC attributes for the EVC Ethernet policy.

EVC attributes are organized under the following categories:.

- Service Attributes
- VLAN Match Criteria
- VLAN Rewrite Criteria

The following sections describe how to set the options under each category.

Setting the Service Attributes

To set the EVC service attributes, perform the following steps.

Step 1 Check the **AutoPick Service Instance ID** check box to specify that the service instance ID will be autogenerated and allocated to the link during service request creation.

If the check box is unchecked, while setting the Prime Fulfillment link attributes during service request creation, Prime Fulfillment will prompt the operator to specify the service instance ID.

Usage notes:

- The service instance ID represents an Ethernet Flow Point (EFP) on an interface in the EVC infrastructure. The service instance ID is locally significant to the interface. This ID has to be unique only at the interface level. The ID must be a value from 1 to 8000.
- There are no resource pools available in Prime Fulfillment from which to allocate the service instance IDs.
- It is the responsibility of the operator creating the service request to maintain the uniqueness of the ID at the interface level.

Step 2 Check the **AutoPick Service Instance Name** check box to have Prime Fulfillment autogenerate a service instance name when you create a service request based on the policy. The autogenerated value is in the following pattern: *CustomerName_ServiceRequestJobID*.

If the check box is unchecked, then you can enter a value during service request creation.

Step 3 Check the **Enable PseudoWire Redundancy** check box to enable pseudowire redundancy (alternative termination device) under certain conditions.

Usage notes:

- Enable Pseudo Wire Redundancy is only available if the MPLS Core Connectivity Type was set as PSEUDOWIRE in the Service Options window (see Setting the Service Options, page 3-20).
- See Appendix E, “Terminating an Access Ring on Two N-PEs” and, specifically, the section Using N-PE Redundancy in FlexUNI/EVC Service Requests, page E-3, for notes on how this option can be used.

Step 4 Check the **AutoPick VC ID** check box to have Prime Fulfillment autopick the VC ID during service request creation.

If this check box is unchecked, the operator will be prompted to specify a VC ID during service request creation.

Usage notes:

- This attribute is available only if MPLS Core Connectivity of Type was set as PSEUDOWIRE or VPLS in the Service Options window (see Setting the Service Options, page 3-20).
- When AutoPick VC ID is checked, Prime Fulfillment allocates a VC ID for pseudowires from the Prime Fulfillment-managed VC ID resource pool.
- If MPLS Core Connectivity of Type is VPLS, Prime Fulfillment allocates the VPLS VPN ID from the Prime Fulfillment-managed VC ID resource pool.

Step 5 Check the **AutoPick VFI Name** check box to have Prime Fulfillment autopick the virtual forwarding instance (VFI) name during service request creation.

If this check box is unchecked, the operator will be prompted to specify a VFI name during service request creation.

Note The AutoPick VFI Name attribute is only applicable if the MPLS Core Connectivity Type is set as VPLS. For other core types (PSEUDOWIRE and LOCAL), this attribute will not be displayed.

Step 6 Check the **AutoPick Bridge Domain/VLAN ID** check box to have Prime Fulfillment autopick the VLAN ID for the service request during service request creation.

If this check box is unchecked, the operator will be prompted to specify a VLAN ID during service request creation.

Usage notes:

- AutoPick Bridge Domain/VLAN ID consumes a global VLAN ID on the device.
- The bridge domain/VLAN ID is picked from the existing Prime Fulfillment VLAN pool. Once the VLAN ID is assigned in the service request, Prime Fulfillment makes the VLAN ID unavailable for subsequent service requests.
- In the case of manual VLAN ID allocation, Prime Fulfillment does not manage the VLAN ID if the ID lies outside the range of an Prime Fulfillment-managed VLAN pool. In this case, the operator must ensure the uniqueness of the ID in the Ethernet access domain. If an operator specifies a VLAN ID that is within the range of an Prime Fulfillment-managed VLAN pool and the VLAN ID is already in use in the access domain, Prime Fulfillment displays an error message indicating that the VLAN ID is in use.

Note on Access VLAN IDs

An access VLAN ID is of local significance to the EVC-capable ports. It should not be confused with the global VLANs. This can be visualized as a partitioning of the Ethernet access network beyond the EVC ports into several subEthernet access domains (one each for an EVC-capable port).

However, all the service interfaces on the Ethernet access nodes beyond the EVC ports will have this very same VLAN ID for a link. This ID must be manually specified by the operator when setting the link attributes during service request creation. The operator must ensure the uniqueness of the ID across the EVC-demarcated Ethernet access domain.

These VLAN IDs are not managed by Prime Fulfillment by means of locally-significant VLAN pools. But once a VLAN ID is assigned for a link in the service request, Prime Fulfillment makes the VLAN unavailable for subsequent service requests within the Ethernet access domain demarcated by the EVC. Likewise, if a manually-specified VLAN is already in use in the access domain delimited by the EVC, Prime Fulfillment will display an error message indicating that the new VLAN ID being specified is already in use on the NPC. The operator will be prompted to specify a different VLAN ID, which will be provisioned on the L2 access nodes.

Step 7 Check the **AutoPick Bridge Group Name** check box to have Prime Fulfillment autopick the group name for the service request during service request creation.

If this check box is unchecked, the operator will be prompted to specify a group name during service request creation. If the check box is checked, the group name will default to the customer name.

Note This attribute is applicable only for supported IOS XR devices.

Step 8 Check the **AutoPick Bridge Domain Name** check box to have Prime Fulfillment autopick the domain name for the service request during service request creation.

Usage notes:

- If this check box is unchecked, the operator will be prompted to specify a domain name during service request creation.
- If the check box is checked, the domain name will default to the following format:
  - For pseudowire and local connect core types: *ISC-Job-Job_ID*, where *Job_ID* is the service request job ID.

– For VPLS core type: *ISC-VPN_Name-VPN_ID*, where *VPN_Name* is the name of the VPLS VPN being used, and *VPN_ID* is the VPN ID used in the service request.

Note This attribute is applicable only for supported IOS XR devices.

Step 9 Continue with the steps contained in the next section, Setting the VLAN Matching Criteria Attributes, page 3-25.

Setting the VLAN Matching Criteria Attributes

Prior to the introduction of the EVC capability, service providers could either deploy service-multiplexed services (ERS/ERMS or EVPL/EVCS) or service-bundled services on a single port. Both could not be supported simultaneously due to the limitations in the infrastructure, which only allowed matching the outer-most VLAN tag.

One of the key benefits of EVC support in Prime Fulfillment is to provide a flexible means to examine the VLAN tags (up to two levels) of the incoming frames and associate them to appropriate Ethernet Flow Points (EFPs). This allows service providers to deploy simultaneously both the service-multiplexed and service-bundled services on a single port.

To set the EVC VLAN matching criteria attributes, perform the following steps.

Step 1 Check the **Both Tags** check box to enable service requests created with the policy to match both the inner and outer VLAN tags of the incoming frames.

If you do not check this check box, service requests created with the policy will match only the outer VLAN tag of the incoming frames.

Checking the Both Tags attribute causes the Inner VLAN Ranges attribute (covered in the next steps) to appear in the EVC Attribute window.

Step 2 Check the **Inner VLAN Ranges** check box to enable the range of inner VLAN tags to be specified during service request creation.

If the check box is unchecked, the range of inner VLAN tags are not allowed. In this case, the operator must specify discrete VLAN IDs during service request creation.

Step 3 Check the **Outer VLAN Ranges** check box to enable the range of outer VLAN tags to be specified during service request creation.

If the check box is unchecked, the range of outer VLAN tags are not allowed. In this case, the operator must specify discrete VLAN IDs during service request creation.

Step 4 Check the **AutoPick Outer VLAN** check box to have Prime Fulfillment autopick the outer VLAN ID from a previously created outer VLAN ID resource pool during service request creation.

If this check box is unchecked, the operator will be prompted to specify an outer VLAN ID during service request creation.

Note Use of the AutoPick Outer VLAN attribute requires that two elements have already been set up in Prime Fulfillment. One is an Interface Access Domain, which is a logical element that groups the physical ports of an N-PE device. The other is an EVC Outer VLAN resource pool, which is used by the Interface Access Domain. For instructions on how to set up these elements, see the sections Setting Up Resources, page 2-40, and Resource Pools, page 2-44.

Usage notes:

- AutoPick Outer VLAN can be used for interfaces that support EVC functionality.
- AutoPick Outer VLAN consumes a VLAN ID on the interface that supports EVC.
- The bridge domain VLAN ID is picked from the existing Prime Fulfillment VLAN pool.

Step 5 Continue with the steps contained in the next section, Setting the VLAN Rewrite Criteria Attributes, page 3-26.

Setting the VLAN Rewrite Criteria Attributes

Together with VLAN matching criteria, VLAN rewrite makes the EVC infrastructure very powerful and flexible. The following VLAN rewrite options are supported:

- Pop one or two tags.
- Push one or two tags.
- Translation (1:1, 2:1, 1:2, 2:2).

Be aware of the following considerations when setting the VLAN rewrite criteria attributes:

- Only one kind of rewrite can be done on every CE-facing EVC link.
- All VLAN rewrites are done using the **symmetric** keyword on the ingress traffic (for example, **rewrite ingress tag pop 2 symmetric**).
- For any service instance, only one type of rewrite option (pop, push, or translate) is allowed per instance. For example, if pop out is enabled, push inner, push outer, translate inner, and translate outer are not available.

To set the EVC VLAN rewrite criteria attributes, perform the following steps.

Step 1 Check the **Pop Outer** check box to pop the outer VLAN ID tag of the incoming frames that fulfill the match criteria.

If this check box is unchecked, the outer tag of the incoming traffic is not popped.

Step 2 Check the **Pop Inner** check box to pop the inner VLAN ID tag of the incoming frames that fulfill the match-criteria.

If this check box is unchecked, the inner tag is not popped. Note that, if Pop Inner is checked, Pop Outer is automatically checked.

Step 3 Check the **Push Outer** check box to impose an outer VLAN ID tag onto the incoming frames that fulfill the match criteria.

If this check box is unchecked, no outer tag is imposed on the incoming frames.

Usage notes:

- If Push Outer is checked, all service requests created with the policy push a dot1q outer tag on the incoming frames matching the match criteria. When creating the link during service creation, the operator can specify an outer tag with a value from 1 to 4096.
- This attribute is available regardless of the number of tags used in the match criteria. Whether the incoming traffic is double tagged or single tagged, if Push Outer is enabled, all corresponding service requests push an outer tag. All subsequent nodes consider only the outer-most two tags (if EVC-capable) or just one tag (not EVC-capable) and treat the inner-most tags transparently as payload.

- This VLAN ID is not derived from Prime Fulfillment-managed VLAN ID pools.

Step 4 Check the **Push Inner** check box to impose an inner VLAN ID tag onto the incoming frames that fulfill the match criteria.

This operation pushes both an inner and an outer tag onto the incoming packet, not just an inner tag. If this check box is unchecked, no inner tag is imposed on the incoming frames.

Usage notes:

- If Push Inner is checked, all service requests created with the policy push a dot1q inner tag on the incoming frames matching the match criteria. When creating the link during service creation, the operator can specify an inner tag with a value from 1 to 4096.
- If Push Inner is checked, Push Outer is automatically checked.
- This attribute is available regardless of the number of tags used in the match criteria. Regardless of whether the incoming traffic is double tagged or single tagged, if Push Inner is enabled, all corresponding service requests push an inner tag. All subsequent nodes consider only the outer-most two tags (if EVC-capable) or just one tag (not EVC-capable) and treat the inner-most tags transparently as payload.
- This VLAN ID is not derived from Prime Fulfillment-managed VLAN ID pools.

Step 5 Check the **Translate Outer** check box to allow the operator to specify a target outer VLAN ID during service request creation.

The outer tag of all the incoming frames that fulfill the match criteria are translated to this ID. If the check box is unchecked, no outer tag translation is performed. See Table 3-2.

Step 6 Check the **Translate Inner** check box to allow the operator to specify a target inner VLAN ID during service request creation.

The inner tag of all the incoming frames that fulfill the match criteria are translated to this ID. If the check box is unchecked, no inner tag translation is performed. See Table 3-2.

Note Table 3-2 summarizes the realization of different VLAN translations available in the EVC infrastructure. The second and third columns (Match Outer Tag and Match Inner Tag) refer to policy settings. The last two columns (Translate Outer Tag and Translate Inner Tag) indicate the VLAN translation that occurs on the incoming frames.

Table 3-2 VLAN Translation Summary Table

Type	Match Outer Tag	Match Inner Tag	Translate Outer Tag	Translate Inner Tag	Push Outer Tag
1:1	True	N/A	Yes	No	N/A
1:2	True	N/A	N/A	N/A	Yes
2:1	True	True	Yes	No	N/A
2:2	True	True	Yes	Yes	N/A

Step 7 Click **Next**.

The Interface Attribute window appears.

Step 8 Continue with the steps contained in the next section, Setting the Interface Attributes, page 3-28.

Setting the Interface Attributes

This step of creating the EVC Ethernet policy involves setting the interface attributes in the Interface Attribute window. The attributes you can configure in this window are grouped under the following categories:

- UNI Information
- VLAN
- Pseudowire
- ACL
- Security
- UNI Storm Control
- Protocol

In some cases, checking an attribute causes additional attributes to appear in the GUI. This is covered in the steps that follow.

Note If the CE is directly connected to an N-PE, only speed, duplex, UNI shutdown, and other generic options are presented. In this case, port security, storm control, L2 protocol tunneling, and other advanced features are not supported due to the current platform limitations. If these features are needed for a service, the service provider must deploy Layer 2 Ethernet access nodes beyond the EVC to support these requirements.

Note Attributes available in the Interface Attributes window dynamically change based on the choice made for the MPLS Core Connectivity Type (PSEUDOWIRE, LOCAL, or VPLS) in the Service Options window (see Setting the Service Options, page 3-20). For completeness, all attributes available for the different core types are documented in the following steps. Attributes apply to all core types, unless otherwise noted.

To set the EVC interface attributes, perform the following steps.

Step 1 Check the **Standard UNI Port** check box to enable port security.

This is the default. When you uncheck the check box, the port is treated as an uplink with no security features, and the window dynamically changes to eliminate items related to port security.

Step 2 Check the **UNI Shutdown** check box if you want to leave the UNI port shut during service activation, for example, when the service provider wants to deploy a service in the network but wants to activate it at a later time.

Step 3 Check the **Keep Alive** check box to configure keepalives on the UNI port.

By default, this check box is unchecked, which causes the command **no keepalive** to be provisioned on the UNI port. This prevents a CPE from sending keepalive packets to the U-PE, for security purposes. This attribute is editable, in order to support modification on a per-service request basis.

Step 4 Enter a **Link Media** (optional) of None, auto-select, rj45, or sfp.

Step 5 Enter a **Link Speed** (optional) of None, 10, 100, 1000, Auto, or nonegotiate.

Step 6 Enter a **Link Duplex** (optional) of None, Full, Half, or Auto.

Step 7 Choose an **Encapsulation** type.

The choices are:

- **DOT1QTRUNK**—Configures the UNI as a trunk with 802.1q encapsulation. If the UNI belongs to a directly connected and EVC link, this setting signifies that the incoming frames are 802.1q encapsulated and that they match the VLAN ID configured for the link. This specific topology does not involve a trunk UNI as such.
- **DOT1QTUNNEL**—Configures the UNI as an 802.1q tunnel (also known as a dot1q tunnel or Q-in-Q) port.
- **ACCESS**—Configures the UNI as an access port.

Step 8 Specify the type of **VLAN Translation** for this policy by clicking the appropriate radio button.

The choices are:

- **No**—No VLAN translation is performed. (This is the default.)
- **1:1**—1:1 VLAN translation. Translates an incoming customer VLAN to another.
- **2:1**—2:1 VLAN translation. Converts both inner and outer VLANs to a single VLAN.
- **1:2**—1:2 VLAN translation. Pushes one more provider VLAN.
- **2:2**—2:2 VLAN translation. Translates both inner and outer VLANs to two other VLANs.

Note For more details on how VLAN translation is supported in EVC Ethernet services, see the coverage of the VLAN Translation attribute in Managing an EVC Ethernet Service Request, page 3-34.

Step 9 Check the **Use PseudoWireClass** check box to enable the selection of a pseudowire class.

This attribute is unchecked by default.

Usage notes:

- The pseudowire class name is used for provisioning pw-class commands on IOS XR devices. See Creating and Modifying Pseudowire Classes, page 3-14 for additional information on pseudowire class support for IOS XR devices.
- If **Use PseudoWireClass** is checked, an additional attribute, **PseudoWireClass**, appears in the GUI. Click the **Select** button of PseudoWireClass attribute to choose a pseudowire class previously created in Prime Fulfillment.
- The Use PseudoWireClass attribute is only available if the MPLS core connectivity type was set as PSEUDOWIRE in the Service Options window (see Setting the Service Options, page 3-20).
- Use PseudoWireClass is only applicable for IOS XR devices.

Step 10 For **L2VPN Group Name** choose one of the following from the drop-down list:

- **ISC**
- **VPNSC**

Usage notes:

- This attribute is used for provisioning the L2VPN group name on IOS XR devices.

Note The choices in the drop-down list are derived from a configurable DCPL property. For information about how to define the L2VPN Group Name choices available in the drop-down list, see Defining L2VPN Group Names for IOS XR Devices, page 3-18.

- The L2VPN Group Name attribute is not available if the MPLS core connectivity type was set as VPLS in the Service Options window (see Setting the Service Options, page 3-20).
- L2VPN Group Name is only applicable for IOS XR devices.

Step 11 Enter an **E-Line Name** to specify the point-to-point (p2p) E-line name.

Usage notes:

- If no value is specified for the **E-Line Name** in either the policy or the service request based on the policy, Prime Fulfillment autogenerates a default name as follows:
  - For PSEUDOWIRE core connectivity type, the format is:

    DeviceName--VC_ID
  - For LOCAL core connectivity type, the format is:

    DeviceName--0--VLAN_ID

  If the default name is more than 32 characters, the device names are truncated.
- The E-Line Name attribute is not available if the MPLS core connectivity type was set as VPLS in the Service Options window (see Setting the Service Options, page 3-20).
- E-Line Name is only applicable for IOS XR devices.

Step 12 Check the **PW Tunnel Selection** check box if you want to be able to manually select the Traffic Engineering (TE) tunnel for the pseudowire connecting point-to-point N-PEs.

This attribute is unchecked by default.

Subsequently, when you create a service request based on this policy, you must specify the TE tunnel ID in a field provided. Prime Fulfillment uses the tunnel information to create and provision a pseudowire class that describes the pseudowire connection between two N-PEs. This pseudowire class can be shared by more than one pseudowire, as long as the pseudowires share the same tunnel ID and remote loopback address. You are responsible to ensure that the tunnel interface and associated ID are configured. During service request creation when you specify the tunnel ID number, Prime Fulfillment does not check the validity of the value. That is, Prime Fulfillment does not verify the existence of the tunnel.

Step 13 Check the **N-PE Pseudo-wire on SVI** check box to have Prime Fulfillment generate forwarding commands under SVIs (switch virtual interfaces).

By default, this check box is not checked. In this case, Prime Fulfillment generates forwarding commands under the service instance.

For an EVC link, the attribute N-PE Pseudo-wire on SVI is dependent on the value of the attribute Configure with Bridge Domain (this is available in the policy workflow in the EVC Policy Editor - Service Options window). N-PE Pseudo-wire on SVI, if enabled, will be reflected only when Configure with Bridge Domain is set to true. Otherwise, the service request will not be created with xconnect under SVI, even if N-PE Pseudo-wire on SVI is enabled.

Usage notes:

- Prime Fulfillment supports a hybrid configuration for EVC service requests. In a hybrid configuration, the forwarding commands (such as xconnect) for one side of an attachment circuit can be configured under a service instance, and the xconnect configuration for the other side of the attachment circuit can be configured under a switch virtual interface (SVI).
- For examples of these cases, see configlet examples EVC (Pseudowire Core Connectivity, Bridge Domain, Pseudowire on SVI), page 3-212 and EVC (Pseudowire Core Connectivity, no Bridge Domain, no Pseudowire on SVI), page 3-213.
- N-PE Pseudo-wire on SVI is applicable for all connectivity types (PSEUDOWIRE, VPLS, and LOCAL), but a hybrid SVI configuration is possible only for pseudowire connectivity.

- When MPLS Core Connectivity Type is set as VPLS, the N-PE Pseudo-wire on SVI attribute is always enabled in the policy and service request.
- When MPLS Core Connectivity Type is set as LOCAL connectivity type, the N-PE Pseudo-wire on SVI attribute is always disabled in the policy and service request.
- The N-PE Pseudo-wire on SVI attribute is not supported for IOS XR devices. Only subinterfaces are supported on ASR 9000 devices; service instance is not supported. All the xconnect commands are configured on L2 subinterfaces.
- Table 3-3 shows various use cases for hybrid configuration for EVC service requests.

Table 3-3 Use Cases for Hybrid Configuration for EVC Service Requests

Use Bridge Domain	EVC	N-PE Pseudowire on SVI	CLIs Generated
True	True	True	• xconnect under VLAN interface. • Service instance under main interface.
True	True	False	• xconnect under service instance. • Service instance under main interface.
False	True	N/A	• xconnect under service instance. • Service instance under main interface.
True	False	True	xconnect under VLAN interface.
True	False	False	xconnect under subinterface.
False	False	False	xconnect under subinterface.

Step 14 Check the **Use Existing ACL Name** check box if you want to assign your own named access list to the port.

By default, this check box is not checked and Prime Fulfillment automatically assigns a MAC-based ACL on the customer facing UNI port, based on values you enter in **UNI MAC addresses** (below).

Step 15 Enter a **Port-Based ACL Name** (if you checked the **Use Existing ACL Name** check box, as mentioned in the previous step).

Note Prime Fulfillment does not create this ACL automatically. The ACL must already exist on the device, or be added as part of a template, before the service request is deployed. Otherwise, deployment will fail.

Step 16 Enter one or more Ethernet MAC addresses in **UNI MAC addresses**.

This selection is present only if you uncheck the **Use Existing ACL Name** check box. Click the **Edit** button to bring up a pop-up window in which you enter MAC addresses to be allowed or denied on the port. You can also specify a range of addresses by setting a base MAC address and a filtered MAC address.

Step 17 Check the **UNI Port Security** check box if you to want to provision port security-related CLIs to the UNI port by controlling the MAC addresses that are allowed to go through the interface.

a. For **Maximum Number of MAC address**, enter the number of MAC addresses allowed for port security.

b. For **Aging,** enter the length of time the MAC address can stay on the port security table.

c. For **Violation Action**, choose what action will occur when a port security violation is detected:

- **PROTECT**—Drops packets with unknown source addresses until a sufficient number of secure MAC addresses are removed to drop below the maximum value.
- **RESTRICT**—Drops packets with unknown source addresses until a sufficient number of secure MAC addresses are removed to drop below the maximum value and causes the Security Violation counter to increment.
- **SHUTDOWN**—Puts the interface into the error-disabled state immediately and sends an SNMP trap notification.

d. In the **Secure MAC Addresses** field, enter one or more Ethernet MAC addresses.

Step 18 Check the **Enable Storm Control** check box to help prevent the UNI port from being disrupted by a broadcast, multicast, or unicast storm.

Enter a threshold value for each type of traffic. The value, which can be specified to two significant digits, represents the percentage of the total available bandwidth of the port. If the threshold of a traffic type is reached, further traffic of that type is suppressed until the incoming traffic falls below the threshold level.

Step 19 Check the **Protocol Tunnelling** check box if you want to define the Layer 2 Bridge Protocol Data Unit (BPDU) frames that can be tunneled over the core to the other end.

For each protocol that you choose, enter the shutdown threshold and drop threshold for that protocol:

a. **Enable cdp**—Enable Layer 2 tunnelling on Cisco Discover Protocol (CDP).

b. **cdp shutdown threshold—**Enter the number of packets per second to be received before the interface is shut down.

c. **cdp drop threshold**—Enter the number of packets per second to be received at which point the interface will start dropping CDP packets.

d. **Enable vtp**—Enable Layer 2 tunnelling on VLAN Trunk Protocol (VTP).

e. **vtp shutdown threshold**—Enter the number of packets per second to be received before the interface is shut down.

f. **vtp drop threshold**—Enter the number of packets per second to be received at which point the interface will start dropping VTP packets.

g. **Enable stp—**Enable Layer 2 tunnelling on Spanning Tree Protocol (STP).

h. **stp shutdown threshold—**Enter the number of packets per second to be received before the interface is shut down.

i. **stp drop threshold**—Enter the number of packets per second to be received at which point the interface will start dropping STP packets.

j. **Recovery Interval**—Enter the amount of time, in seconds, to wait before recovering a UNI port.

Step 20 Enter the **MTU Size** in bytes.

The maximum transmission unit (MTU) size is configurable and optional. The default size is 9216, and the range is 1500 to 9216. Prime Fulfillment does not perform an integrity check for this customized value. If a service request goes to the Failed Deploy state because this size is not accepted, you must adjust the size until the Service Request is deployed.

In Cisco Prime Fulfillment 1.0, different platforms support different ranges.

- For the 3750 and 3550 platforms, the MTU range is 1500 to 1546.

- For the Cisco 7600 Ethernet port, the MTU size is always 9216. Even with the same platform and same IOS release, different line cards support the MTU differently. For example, older line cards only take an MTU size of 9216 and newer cards support 1500 to 9216. However, Prime Fulfillment uses 9216 in both cases.
- For the Cisco 7600 SVI (interface VLAN), the MTU size is 1500 to 9216.

Step 21 If you would like to enable template association for this policy, click the **Next** button.

See the section Enabling Template Association, page 3-33 for information about this feature.

Note An additional window appears the policy workflow before the Template Association window. This window allows you to create user-defined attributes within the policy (and service requests based on the policy). For background information on how to use the additional information feature, see Appendix H, "Adding Additional Information to Services." If you are not using this feature, click **Next** to proceed to the Template Association window, or else click **Finish** to save the policy.

Step 22 To save the EVC policy, click **Finish**.

To create a service request based on an EVC policy, see Managing an EVC Ethernet Service Request, page 3-34.

Enabling Template Association

The Prime Fulfillment template feature gives you a means to download free-format CLIs to a device. If you enable templates, you can create templates and data files to download commands that are not currently supported by Prime Fulfillment.

Step 1 To enable template association for the policy, click the **Next** button in the Interface Attribute window (before clicking **Finish**).

The Template Association window appears. In this window, you can enable template support and, optionally, associate templates and data files with the policy. For instructions about associating templates with policies and how to use the features in this window, see Using Templates with Service Requests, page 9-24.

Step 2 When you have completed setting up templates and data files for the policy, click **Finish** in the Template Association window to close it and return to the Policy Editor window.

Step 3 To save the EVC policy, click **Finish**.

To create a service request based on an EVC policy, see Managing an EVC Ethernet Service Request, page 3-34.

Managing an EVC Ethernet Service Request

This section provides information on how to provision an EVC Ethernet service request. It contains the following subsections:

Introducing EVC Service Requests

An EVC Ethernet service request allows you to configure interfaces on an N-PE to support the EVC features described in Creating an EVC Ethernet Policy, page 3-18. To create an EVC service request, an EVC service policy must already be defined. Based on the predefined EVC policy, an operator creates an EVC service request and deploys the service. One or more templates can also be associated to the N-PE as part of the service request.

Creating an EVC Ethernet service request involves the following steps:

- Choose an existing EVC Ethernet policy.
- Choose a VPN.

Note When working with VPN objects in the context of EVC Ethernet policies and service requests, only the VPN name and customer attributes are relevant. Other VPN attributes related to MPLS and VPLS are ignored.

- Specify a bridge domain configuration (if applicable).
- Specify a service request description.
- Specify automatic or manual allocation of the VC ID or VPLS VPN ID.
- Add direct connect links (if applicable).
- Add links with L2 access nodes (if applicable).
- Choose the N-PE and UNI interface for links.
- For links with L2 access nodes, choose a Named Physical Circuit (NPC) if more than one NPC exists from the N-PE or the UNI interface.
- Edit the link attributes.
- Modify the service request.
- Save the service request.

For sample configlets for EVC Ethernet scenarios, see Sample Configlets, page 3-176.

Creating an EVC Service Request

To create an EVC Ethernet service request, perform the following steps.

Step 1 Choose **Operate > Service Requests > Service Request Manager**.

The Service Request Manager window appears.

Step 2 Click **Create**.

The Service Request Editor window appears.

Step 3 From the Policy drop-down list, choose an EVC policy from the policies previously created (see Creating an EVC Ethernet Policy, page 3-18).

The EVS Service Request editor window appears.

The new service request inherits all the properties of the chosen EVC policy, such as all the editable and non-editable features and pre-set parameters.

Step 4 Continue with the steps contained in the next section, Setting the Service Request Details, page 3-35.

Setting the Service Request Details

After you have selected the EVC Ethernet policy to be used as the basis of the service request, the EVC Service Request Editor window appears. It is divided into three main sections:

- Link Page
- Direct Connect Links (no NPCs)
- Links with L2 Access Nodes (involves NPCs)

This window enables you to specify options for the service request, as well as configure directly connected links and links with L2 access nodes. The options displayed in first section of the window change, depending on the MPLS Core Connectivity Type that was specified in the policy (pseudowire, VPLS, or local). For clarity, each of these scenarios is presented in a separate section below, to highlight the different window configurations and behavior of the displayed options.

Proceed to the appropriate section, as determined by the MPLS Core Connectivity Type for the policy:

- Pseudowire Core Connectivity, page 3-35
- VPLS Core Connectivity, page 3-37
- Local Core Connectivity, page 3-39

Instructions for setting up direct connect links and links with L2 access nodes are presented in later sections.

Pseudowire Core Connectivity

This sections covers the case in which the MPLS Core Connectivity Type for the EVC Ethernet policy is PSEUDOWIRE.

To set the attributes in the first section of the Link Page window. perform the following steps.

Note The **Job ID** and **SR ID** fields are read-only. When the service request is being created for the first time, the fields display a value of NEW. When an existing service request is being modified, the values of the fields indicate the respective IDs that the Prime Fulfillment database holds within the editing flow of the service request.

Note The **Policy** field is read-only. It displays the name of the policy on which the service request is based. Clicking on the read-only policy name displays a list of all the attribute values set within the policy.

Step 1 Click **Select VPN** to choose a VPN for use with this service request.

The Select VPN window appears with the VPNs defined in the system.

Note The same VPN can be used by service requests with LOCAL and PSEUDOWIRE core types. If a VPN for a service request is used with VPLS core type, the same VPN cannot be used for service requests with LOCAL or PSEUDOWIRE core type.

Step 2 Choose a **VPN Name** in the Select column.

Step 3 Click **Select**.

The EVC Service Request Editor window appears with the VPN name displayed.

Step 4 Check the **AutoPick VC ID** check box if you want Prime Fulfillment to choose a VC ID.

If you do not check this check box, you will be prompted to provide the ID in the VC ID field, as covered in the next step.

When AutoPick VC ID is checked, Prime Fulfillment allocates a VC ID for pseudowires from the Prime Fulfillment-managed VC ID resource pool. In this case, the text field for the VC ID option is non-editable.

Step 5 If AutoPick VC ID was unchecked, enter a VC ID in the **VC ID** field.

Usage notes:

- The VC ID value must be an integer value corresponding to a VC ID.
- When a VC ID is manually allocated, Prime Fulfillment verifies the VC ID to see if it lies within Prime Fulfillment's VC ID pool. If the VC ID is in the pool but not allocated, the VC ID is allocated to the service request. If the VC ID is in the pool and is already in use, Prime Fulfillment prompts you to allocate a different VC ID. If the VC ID lies outside of the Prime Fulfillment VC ID pool, Prime Fulfillment does not perform any verification about whether or not the VC ID allocated. The operator must ensure the VC ID is available.
- The VC ID can be entered only while creating a service. If you are editing the service request, the VC ID field is not editable.

Step 6 Check the **Enable PseudoWire Redundancy** check box to enable pseudowire redundancy (alternative termination device) under certain conditions.

See Appendix E, "Terminating an Access Ring on Two N-PEs" and, specifically, the section Using N-PE Redundancy in FlexUNI/EVC Service Requests, page E-3, for notes on how this option can be used.

Step 7 If the AutoPick VC ID attribute was unchecked, enter a VC ID for the backup pseudowire in the **Backup PW VC ID** field.

See the usage notes for the AutoPick VC ID attribute in Step 7, above. The backup VC ID behaves the same as the VC ID of the primary pseudowire.

Step 8 Check the **Configure Bridge Domain** check box to determine bridge domain characteristics.

The behavior of the Configure Bridge Domain option works in tandem with the choice you selected in the MPLS Core Connectivity Type option in the EVC policy, which in this case is pseudowire core connectivity. There are two cases:

- With EVC:
  - If **Configure With Bridge Domain** is checked, the policy will configure pseudowires under SVIs associated to the bridge domain.
  - If **Configure With Bridge Domain** is unchecked, the policy will configure pseudowires directly under the service instance. This will conserve the global VLAN.
- Without EVC:
  - If **Configure With Bridge Domain** is checked, the policy will configure pseudowires under SVIs.
  - If **Configure With Bridge Domain** is unchecked, the policy will configure pseudowires directly under subinterfaces.

  Pseudowires can be configured either directly under service instance of the corresponding EVC-capable interface or under SVIs associated to the bridge domain.

Step 9 Check the **Use Split Horizon** check box to enable split horizon with bridge domain.

Usage notes:

- The Use Split Horizon attribute is disabled by default.
- The Use Split Horizon attribute can be used only when the Configure Bridge Domain check box is checked (enabled).
- When Use Split Horizon is enabled, the **bridge domain** command in the CLI will be generated with split horizon. When it is disabled, the **bridge domain** command will be generated without split horizon.

Step 10 Click the “Click here” link of the **Description** attribute to enter a description label for the service request.

This is useful for searching the Prime Fulfillment database for the particular service request.

A dialogue appears in which you can enter a description.

Step 11 To set up direct connect links, see the section Setting Direct Connect Links, page 3-41.

Step 12 To set up links with L2 access nodes, see the section Setting Links with L2 Access Nodes, page 3-52.

VPLS Core Connectivity

This section covers the case in which the MPLS Core Connectivity Type for the EVC Ethernet policy is VPLS.

To set the attributes in the first section of the Link Page window, perform the following steps.

Step 1 The **Job ID** and **SR ID** fields are read-only.

When the service request is being created for the first time, the fields display a value of NEW. When an existing service request is being modified, the values of the fields indicate the respective IDs that the Prime Fulfillment database holds within the editing flow of the service request.

Step 2 The **Policy** field is read-only. It displays the name of the policy on which the service request is based.

Step 3 Click **Select VPN** to choose a VPN for use with this service request.

The Select VPN window appears with the VPNs defined in the system.

Note The same VPN can be used by service requests with LOCAL and PSEUDOWIRE core types. If a VPN for a service request is used with VPLS core type, the same VPN cannot be used for service requests with LOCAL or PSEUDOWIRE core type.

Note If the same VPN is used among multiple service requests, all having VPLS core type, then all these service requests participate in the same VPLS service.

Step 4 Choose a **VPN Name** in the Select column.

Step 5 Click **Select**.

The EVC Service Request Editor window appears with the VPN name displayed.

Step 6 Check the **AutoPick VPLS VPN ID** check box if you want Prime Fulfillment to choose a VPLS VPN ID.

If you do not check this check box, you will be prompted to provide the VPN ID in the VPLS VPN ID field, as covered in the next step.

- When AutoPick VPLS VPN ID is checked, Prime Fulfillment allocates a VPLS VPN ID from the Prime Fulfillment-managed VC ID resource pool. In this case, the text field for the VPLS VPN ID option is non-editable.
- If AutoPick VPLS VPN ID is checked and a service request already exists that refers to same VPN object, the VPLS VPN ID of the existing service request is allocated to the new service request.

Step 7 If AutoPick VPLS VPN ID was unchecked, enter a VPLS VPN ID in the **VPLS VPN ID** field.

Usage notes:

- The VPLS VPN ID value must be an integer value corresponding to a VPN ID.
- When a VPLS VPN ID is manually allocated, Prime Fulfillment verifies the VPLS VPN ID to see if it lies within Prime Fulfillment's VC ID pool. If the VPLS VPN ID is in the pool but not allocated, the VPLS VPN ID is allocated to the service request. If the VPLS VPN ID is in the pool and is already in use, Prime Fulfillment prompts you to allocate a different VPLS VPN ID. If the VPLS VPN ID lies outside of the VC ID pool, Prime Fulfillment does not perform any verification about whether the VPLS VPN ID allocated. The operator must ensure the VPLS VPN ID is available.
- The VPLS VPN ID can be entered only while creating a service. If you are editing the service request, the VPLS VPN ID field is not editable.

Step 8 Check the **AutoPick VFI Name** check box if you want Prime Fulfillment to choose a virtual forwarding instance (VFI) name.

If you do not check this check box, you can provide the VFI name in the VFI Name field, as covered in the next step.

Usage notes:

- When AutoPick VFI name is checked, Prime Fulfillment generates a VFI name in the following format:

  VPN name-VC ID

- This attribute is useful when importing an existing service into Prime Fulfillment and mapping it to a service request which has been created for this purpose. Manually specifying the VFI name in the service request allows the VFI name to be matched to that of existing service.

Step 9 If AutoPick VFI Name was unchecked, enter a VFI name in the **VFI Name** field.

Step 10 Choose the **Discovery Mode** type for VPLS autodiscovery.

The choices are:

- **Manual**—Does not provision VPLS autodiscovery on VPLS PE devices configured by the service request. In this case, when a new PE is device is added or removed from the VPLS domain, manual configuration of each neighbor in the VPLS domain is required.
- **Auto Discovery**—Provisions VPLS autodiscovery on VPLS PE devices configured by the service request. With VPLS autodiscovery enabled, neighbor devices automatically detect when PEs are added or removed from the VPLS domain.

For details on how this feature is supported in Prime Fulfillment, device preconfiguration requirements, and limitations, see Provisioning VPLS Autodiscovery on Devices using EVC Service Requests, page 3-167.

Step 11 The **Configure Bridge Domain** check box is check by default and cannot be changed.

Usage notes:

- For VPLS, all configurations are under the SVI.
- When the EVC feature is used, all configurations are under the SVI and also associated to a bridge domain.

Step 12 Check the **Use Split Horizon** check box to enable split horizon with bridge domain.

Usage notes:

- The Use Split Horizon attribute is disabled by default.
- The Use Split Horizon attribute can be used only when the Configure Bridge Domain check box is checked (enabled).
- When Use Split Horizon is enabled, the **bridge domain** command in the CLI will be generated with split horizon. When it is disabled, the **bridge domain** command will be generated without split horizon.

Step 13 Click the “Click here” link of the **Description** attribute to enter a description label for the service request.

A dialogue appears in which you can enter a description.

Step 14 To set up direct connect links, see the section Setting Direct Connect Links, page 3-41.

Step 15 To set up links with L2 access nodes, see the section Setting Links with L2 Access Nodes, page 3-52.

Local Core Connectivity

This section covers the case in which the MPLS Core Connectivity Type for the EVC Ethernet policy is LOCAL.

To set the attributes in the first section of the Link Page window, perform the following steps.

Step 1 The **Job ID** and **SR ID** fields are read-only.

When the service request is being created for the first time, the fields display a value of NEW. When an existing service request is being modified, the values of the fields indicate the respective IDs that the Prime Fulfillment database holds within the editing flow of the service request.

Step 2 The **Policy** field is read-only.

It displays the name of the policy on which the service request is based.

Step 3 Click **Select VPN** to choose a VPN for use with this service request.

The Select VPN window appears with the VPNs defined in the system.

Note The same VPN can be used by service requests with LOCAL and PSEUDOWIRE core types. If a VPN for a service request is used with VPLS core type, the same VPN cannot be used for service requests with LOCAL or PSEUDOWIRE core type.

Step 4 Choose a **VPN Name** in the Select column.

Step 5 Click **Select**.

The EVC Service Request Editor window appears with the VPN name displayed.

Step 6 Check the **Configure Bridge Domain** check box to determine bridge domain characteristics.

Usage notes:

- If Configure Bridge Domain is checked, all links will have the same bridge domain ID allocated from the VLAN pool on the N-PE. All non-EVC links will have the Service Provider VLAN as the bridge domain ID. On the other hand, if no EVC links are added, the Service Provider VLAN will be allocated first and this will be used as the bridge domain ID when EVC links are added.
- If Configure Bridge Domain is unchecked, a maximum of two links that terminate on the same N-PE can be added. (This uses the **connect** command available in the EVC infrastructure.)

  Note See the following comments for details on how Prime Fulfillment autogenerates the connect name.

  Because the device only accepts a maximum of 15 characters for the connect name, the connect name is generated using the following format:

  CustomerNameTruncatedToMaxPossibleCharacters_ServiceRequestJobID

  For example, if the customer name is NorthAmericanCustomer and the service request job ID is 56345, the autogenerated connect name would be NorthAmer_56345.

  The CLI generated would be:

  ```
  connect NorthAmer_56345 GigabitEthernet7/0/5 11 GigabitEthernet7/0/4 18
  ```

  In this case, 11 and 18 are service instance IDs.

- If the policy setting for Configure Bridge Domain is non-editable, the option in the service request will be read-only.

Step 7 Check the **Use Split Horizon** check box to enable split horizon with bridge domain.

Usage notes:

- The Use Split Horizon attribute is disabled by default.
- The Use Split Horizon attribute can be used only when the Configure Bridge Domain check box is checked (enabled).
- When Use Split Horizon is enabled, the **bridge domain** command in the CLI will be generated with split horizon. When it is disabled, the **bridge domain** command will be generated without split horizon.

Step 8 Click the "Click here" link of the **Description** attribute to enter a description label for the service request.

A dialogue appears in which you can enter a description.

Step 9 To set up direct connect links, see the section Setting Direct Connect Links, page 3-41.

Step 10 To set up links with L2 access nodes, see the section Setting Links with L2 Access Nodes, page 3-52.

Setting up Links to the N-PE

The lower two sections of the EVC Service Request Editor window allow you to set up links to the N-PE. For direct connect links, the CE is directly connected to the N-PE, with no intermediate L2 access nodes. For links with L2 access nodes, there are intermediate devices present between the CE and NPE requiring NPCs to be created in Prime Fulfillment.

The Direct Connect Links section of the window is where you set up links that directly connect to the N-PE. No NPC are involved. The Links with L2 Access Nodes section is where you set up links with L2 (Ethernet) access nodes. NPCs are involved.

See the appropriate section, depending on which type of link you are setting up:

- Setting Direct Connect Links, page 3-41
- Setting Links with L2 Access Nodes, page 3-52

Note Many of steps for setting up the two link types are the same. The basic workflow for setting up links, as well as the attributes to be set, are presented in the section Setting Direct Connect Links, page 3-41. Even if you are setting up links with L2 access nodes, it will be helpful to refer to the information presented in that section, as the section on L2 access nodes only covers the unique steps for such links.

Setting Direct Connect Links

To set up the direct connect links, perform the following steps. Most of these steps apply to links with L2 access nodes also.

Step 1 Click **Add** to add a link.

A new numbered row for the link attributes appears.

Step 2 Click **Select NPE** in N-PE column.

The Select PE Device window appears. This window displays the list of currently defined PEs.

a. The **Show PEs with** drop-down list shows PEs by Provider, PE Region Name, or by Device Name.

b. The **Find** button allows a search for a specific PE or a refresh of the window.

c. The **Rows per page** drop-down list allows the user to configure the number of entries displayed on the screen at one time.

Step 3 In the **Select** column, choose the PE device name for the link.

Step 4 Click **Select**.

The EVC Service Request Editor window reappears displaying the name of the selected PE in the N-PE column.

Step 5 Choose the UNI interface from the drop-down list in the UNI column.

Note Prime Fulfillment only displays the available interfaces for the service, based on the configuration of the underlying interfaces, existing service requests that might be using the interface, and the customer associated with the service request. You can click the **Detail** button to display a pop-up window with information on the available interfaces, such as interface name, customer name, VPN name, job ID, service request ID, service request type, translation type, and VLAN ID information.

Note When the UNI is configured on an N-PE device running IOS XR, the Standard UNI Port attribute is not supported. All the CLIs related to Standard UNI Port and UNI Port Security are ignored in this case.

Step 6 Check the **EVC** check box to mark the link for configuring service instance for the links.

Note The EVC check box is mentioned at this stage because the setting of the check box alters the behavior of the link editing function available in the Link Attributes column. This is covered in the next steps.

Note The EVC check box is unchecked by default. The default value for the check box can be changed by setting the value of the DCPL property Pr ovisioning\ProvDrv\CheckFlexUniCheckBox.

Editing the Link Attributes

The next steps document the use of the **Edit** link in the Link Attributes column. (In the case where the link attributes have already been set, this link changes from **Edit** to **Change**.) The link editing workflow changes depending on the status of the EVC check box for the link. If the EVC check box is checked, the editing workflow involves setting attributes in two windows, for two sets of link attributes:

- The EVC Details
- Standard UNI Details

If the EVC check box for the link is not checked, only the Standard UNI Details window is presented.

In the steps that follow, both scenarios covered.

Step 7 Click **Edit** in the Link Attributes column to specify the UNI attributes.

EVC Details Window

If the EVC check box is checked, the EVC Details window appears.

All of the fields in the EVC Details screen are enabled based on the policy settings. For example, if Both Tags is selected in the policy and is editable, then the Match Inner and Outer Tags check box will be selected and editable in this window. The behavior is similar for the other attributes in the EVC Details window

Step 8 Check the **AutoPick Service Instance ID** check box to specify that the service instance ID will be autogenerated and allocated to the link during service request creation.

If the check box is unchecked, you must specify the service instance ID (see the next step).

Usage notes:

- The service instance ID represents an Ethernet Flow Point (EFP) on an interface in the EVC infrastructure. The service instance ID is locally significant to the interface. This ID has to be unique only at the interface level. The ID must be a value from 1 to 8000.
- There are no resource pools available in Prime Fulfillment from which to allocate the service instance IDs.
- In the case of a manually provided service instance ID, it is the responsibility of the operator to maintain the uniqueness of the ID at the interface level.
- This attribute is not displayed for IOS XR devices.

Step 9 If the AutoPick Service Instance ID check box is not checked, enter an appropriate value for the service instance ID in the **Service Instance ID** field.

This attribute is not displayed for IOS XR devices.

Step 10 Check the **AutoPick Service Instance Name** check box to specify that the service instance name will be autogenerated.

If the check box is unchecked, you can specify the service instance name (see the next step).

Usage notes:

- If the check box is checked, the Service Instance Name text field is disabled.
- The service instance name is autogeneratedi n the following pattern: *CustomerName_ServiceRequestJobID*.
- For example configlets, see EVC (AutoPick Service Instance Name), page 3-214, EVC (User-Provided Service Instance Name, Pseudowire Core Connectivity), page 3-216, and EVC (User-Provided Service Instance Name, Local Core Connectivity), page 3-217.
- This attribute is not displayed for IOS XR devices.

Step 11 If the AutoPick Service Instance Name check box is not checked, enter an appropriate value for the service instance ID in the **Service Instance Name** field.

Usage notes:

- The text string representing the service instance name must be 40 characters or less and contain no spaces. Other special characters are allowed.
- If AutoPick Service Instance Name is unchecked and no service instance name is entered in the text field, then Prime Fulfillment does not generate the global **ethernet evc evcname** command in the device configuration generated by the service request.

Step 12 Check the **AutoPick Bridge Domain/VLAN ID** check box to have Prime Fulfillment autopick the VLAN ID for the service request during service request creation.

If this check box is unchecked, the you must specify a bridge domain VLAN ID (see the next step).

Usage notes:

- AutoPick Bridge Domain/VLAN ID consumes a global VLAN ID on the device.
- The bridge domain VLAN ID is picked from the existing Prime Fulfillment VLAN pool.

Step 13 If the AutoPick Bridge Domain/VLAN ID check box is unchecked, enter an appropriate value in the **Bridge Domain/VLAN ID** field.

Note This configuration applies in conjunction with the Configure Bridge Domain option in the EVC Service Request Editor window. If the option is not enabled in that window, then AutoPick Bridge Domain/VLAN ID check box is redundant and not required.

When a VLAN ID is manually allocated, Prime Fulfillment verifies the VLAN ID to see if it lies within Prime Fulfillment's VLAN ID pool. If the VLAN ID is in the pool but not allocated, the VLAN ID is allocated to the service request. If the VLAN ID is in the pool and is already in use, Prime Fulfillment prompts you to allocate a different VLAN ID. If the VLAN ID lies outside of the Prime Fulfillment VLAN ID pool, Prime Fulfillment does not perform any verification about whether the VLAN ID allocated. The operator must ensure the VLAN ID is available.

Step 14 Check the **AutoPick Bridge Domain/VLAN ID Secondary N-PE** check box to have Prime Fulfillment autopick the bridge domain VLAN ID for the secondary N-PE of a dual-homed ring during service request creation.

If this check box is unchecked, the you must specify a secondary bridge domain VLAN ID for the secondary N-PE (see the next step).

Usage notes:

- This attribute is only applicable in the case of a dual-homed ring (a ring that terminates on two different N-PEs). Prime Fulfillment supports having a separate bridge domain VLAN ID for the secondary N-PE.
- In a dual-homed ring, if the two N-PEs are in different access domains, Prime Fulfillment allocates the bridge domain VLAN IDs from both primary and secondary N-PE access domains. When both are in the same Access Domain, Prime Fulfillment allocates a common VLAN ID from the Access Domain to which these belong.
- AutoPick Bridge Domain/VLAN ID Secondary N-PE consumes a global VLAN ID on the device.
- The bridge domain VLAN ID is picked from the existing Prime Fulfillment VLAN pool.
- This attribute is not displayed for IOS XR devices.

Step 15 If the AutoPick Bridge Domain/VLAN ID Secondary N-PE check box is unchecked, enter an appropriate value in the **Bridge Domain/VLAN ID Secondary N-PE** field.

Step 16 Check the **Match Inner and Outer Tags** check box to enable service requests created with the policy to match both the inner and outer VLAN tags of the incoming frames.

If you do not check this check box, service requests created with the policy will match only the outer VLAN tag of the incoming frames.

Checking the Match Inner and Outer Tags attribute causes the Inner VLAN ID and Outer VLAN ID fields (covered in the next steps) to appear.

Step 17 If the Match Inner and Outer Tags check box is checked, enter the inner and outer VLAN tags in the **Inner VLAN ID** and **Outer VLAN ID** fields.

Usage notes:

- You can specify single values, single ranges, multiples values, multiple ranges, or combinations of these. Examples:
  - 10
  - 10, 15,17
  - 10-15
  - 10-15,17-20

  - 10,20-25
- If the Inner VLAN Ranges attribute is set to true in the policy, the Inner VLAN ID field can take a range of inner VLAN tags.
- If the Outer VLAN Ranges attribute is set to true in the policy, the Outer VLAN ID field can take a range of Outer VLAN tags.

Step 18 If the Match Inner and Outer Tags check box is unchecked, enter the outer VLAN tag in the **Outer VLAN ID** field.

Note The VLAN specified in Outer VLAN ID will be provisioned on the rest of the L2 access nodes (if the link has any), including the customer-facing UNI.

Note You may also have Prime Fulfillment autopick the outer VLAN ID as covered in the next step.

Step 19 Check the **AutoPick Outer VLAN** check box to have Prime Fulfillment autopick the outer VLAN ID from a previously created outer VLAN ID resource pool.

If this check box is unchecked, the operator will be prompted to specify an outer VLAN ID.

Note Use of the AutoPick Outer VLAN attribute requires that two elements have already been set up in Prime Fulfillment. One is an Interface Access Domain, which is a logical element that groups the physical ports of an N-PE device. The other is an EVC Outer VLAN resource pool, which is used by the Interface Access Domain. For instructions on how to set up these elements, see the sections Setting Up Resources, page 2-40, and Resource Pools, page 2-44.

Usage notes:

- AutoPick Outer VLAN can be used for interfaces that support EVC functionality
- AutoPick Outer VLAN consumes a VLAN ID on the interface that supports EVC.
- The bridge domain VLAN ID is picked from the existing Prime Fulfillment VLAN pool.

Step 20 In the VLAN Rewrite section of the window, choose a **Rewrite Type** from the drop-down list.

The choices are:

- **Pop**
- **Push**
- **Translate**

The subsequent attributes in the GUI change depending on the choice of Rewrite Type, as covered in the next steps.

Step 21 If Pop is the Rewrite Type, two check boxes appear:

a. Check the **Pop Outer Tag** check box to pop the outer VLAN ID tag of the incoming frames that fulfill the match criteria. If this check box is unchecked, the outer tag of the incoming traffic will not be popped.

b. Check the **Pop Inner Tag** check box to pop the inner VLAN ID tag of the incoming frames that fulfill the match-criteria. If this check box is unchecked, the inner tag will not be changed.

   Note that if Pop Inner Tag is checked, Pop Outer Tag is automatically checked.

Step 22 If Push is the Rewrite Type, two text boxes appear:

a. In the text box **Outer VLAN ID**, enter an outer VLAN ID tag that will be imposed on the incoming frames that fulfill the match criteria. All service requests created with this setting push a dot1q outer tag on the incoming frames matching the match criteria. If a value is not provided, the push operation is ignored and not configured on the device.

b. In the text box **Inner VLAN ID**, enter an inner VLAN ID tag that will be imposed on the incoming frames that fulfill the match criteria. All service requests created with this setting push a dot1q inner tag on the incoming frames matching the match criteria. The Inner VLAN tag cannot be pushed without an Outer VLAN tag. That is, when pushing an Inner VLAN tag, the Outer VLAN tag also must be defined.

Step 23 If Translate is the Rewrite Type, a **Translation Type** drop-down list appears.

The choices available in this list vary depending on the setting of the Match Inner and Outer Tags attribute (set in a previous step).

a. If the Match Inner and Outer Tags check box is checked (true), choose a translation type of **1:1**, **1:2**, **2:1**, or **2:2** from the Translation Type drop-down list.

   – If you choose 1:1 or 2:1, enter a value in the **Outer VLAN ID** text box that appears. The outer tag of all the incoming frames that fulfill the match criteria will be translated to this ID.

   – If you choose 1:2 or 2:2, enter values in the **Outer VLAN ID** and **Inner VLAN ID** text boxes that appear. The outer and inner tags of all the incoming frames that fulfill the match criteria will be translated to these IDs.

b. If the Match Inner and Outer Tags check box is unchecked (false), choose a translation type of **1:1** or **1:2** from the Translation Type drop-down list.

   – If you choose 1:1, enter a value in the **Outer VLAN ID** text box that appears. The outer tag of all the incoming frames that fulfill the match criteria will be translated to this ID.

   – If you choose 1:2, enter values in the **Outer VLAN ID** and **Inner VLAN ID** text boxes that appear. The outer and inner tags of all the incoming frames that fulfill the match criteria will be translated to these IDs.

Step 24 Clicked **Next** to save the settings in the EVC Details window.

The Standard UNI Details window appears.

Step 25 Continue with setting the standard UNI link attributes in the next steps.

Editing the Standard UNI Attributes

The following steps cover setting the attributes in the Standard UNI Details window. In the case of a link which is not set as an EVC link (by not checking the EVC check box in the EVC Service Request Editor window), editing the link attributes begins with this window.

Note The attributes that appear in the Standard UNI Details window are dynamically configured by Prime Fulfillment. Some of the attributes covered in the steps below might not appear in the window, depending on the policy and service request settings or the link type. For example, if the MPLS core connectivity type of the EVC policy is VPLS or local, the pseudowire-related attributes will not appear. Also, setting the link as EVC or non-EVC will change the attributes that appear in the window. In addition, attributes are filtered based on device type (IOS or IOS XR). These and other cases are noted in the steps, for reference.

Step 26 The **N-PE/U-PE Information** and **Interface Name** fields display the PE device and interface name selected in previous steps.

These fields are read-only.

Step 27 Choose an **Encapsulation** type from the drop-down list.

The choices are:

- **DOT1QTRUNK**—Configures the UNI as a trunk with 802.1q encapsulation. If the UNI belongs to a directly connected and EVC link, this setting signifies that the incoming frames are 802.1q encapsulated and that they match the VLAN ID configured for the link. This specific topology does not involve a trunk UNI as such.
- **DOT1QTUNNEL**—Configures the UNI as an 802.1q tunnel (also known as a dot1q tunnel or Q-in-Q) port.
- **ACCESS**—Configures the UNI as an access port.

This attribute allows you to deploy different types of UNI encapsulation on different links of a service.

Usage notes:

- When a U-PE running with IOS is added in the same circuit terminating on an ASR 9000 (functioning in an N-PE role), the all three encapsulation types values will be visible in the drop-down list of the Encapsulation attribute.
- DOT1QTUNNEL is not directly supported for ASR 9000 devices.
- In the case of direct connect links for which EVC is enabled (by checking the EVC check box in the EVC Service Request Editor window), the choices for the Encapsulation type are DOT1Q and DEFAULT.

Step 28 In the **PE/UNI Interface Description** field, enter a description for the interface, if desired.

Step 29 Check the **UNI Shutdown** check box if you want to leave the UNI port shut during service activation (for example, when the service provider wants to deploy a service in the network but wants to activate it at a later time).

Step 30 Specify the type of **VLAN Translation** for the service request by clicking the appropriate radio button.

The choices are:

- **No**—No VLAN translation is performed. (This is the default.)
- **1:1**—1:1 VLAN translation.
- **2:1**—2:1 VLAN translation.
- **1:2**—1:2 VLAN translation.
- **2:2**—2:2 VLAN translation.

Usage notes:

- The VLAN Translation attribute does not appear for direct connect links if the EVC check box is enabled. It does appear for the following combinations:
  - Direct connect links with EVC check box disabled.
  - L2 access nodes with EVC check box enabled or disabled.
- Choosing a selection other than No causes other fields to appear in the GUI, which you can set based on your configuration:
  - **CE VLAN**—Provide a value between 1 and 4096.
  - **Auto Pick**—Check this check box to have Prime Fulfillment autopick the outer VLAN from the VLAN resource pool.
  - **Outer VLAN**—If Auto Pick is unchecked, provide a value between 1 and 4096.

  - **Select where 2:1 or 2:2 translation takes place**—Specify the device where the 2;1 or 2:2 VLAN translation will take place. If you choose Auto, the VLAN translation takes place at the device closest to the UNI port.

- VLAN translation, and all standard UNI and port security attributes are applicable for links with L2 access. If the UNI is on an N-PE, these attributes will not appear.
- When the VLAN translation takes place on a U-PE or PE-AGG device, the VLAN translation command is configured on the NNI interface of the selected device. When the VLAN translation takes place on an NP-E, the VLAN translation command is configured on the UNI interface of the device.
- When there are two NNI interfaces in a ring-based environment, VLAN translation is applied for both of these NNI interfaces.
- 1:1 and 2:1 VLAN translations are supported with the same syntax as for non-EVC (switchport-based N-PE syntax) terminating attachment circuits.

Step 31 Check the **N-PE Pseudo-wire on SVI** check box to have Prime Fulfillment generate forwarding commands under SVIs (switch virtual interfaces).

By default, this check box is not checked. In this case, Prime Fulfillment generates forwarding commands under the service instance.

For an EVC link, the attribute N-PE Pseudo-wire on SVI is dependent on the value of the attribute Configure with Bridge Domain (this is available in the service request workflow in the EVC Service Request Editor window). N-PE Pseudo-wire on SVI, if enabled, will be reflected only when Configure with Bridge Domain is set to true. Otherwise, the service request will not be created with xconnect under SVI, even if N-PE Pseudo-wire on SVI is enabled.

Usage notes:

- For an EVC link, the attribute N-PE Pseudo-wire on SVI is dependent on the value of the attribute Configure with Bridge Domain (in the EVC Service Request Editor window). N-PE Pseudo-wire on SVI, if enabled, will be reflected only when Configure with Bridge Domain is set to true. Otherwise, the service request will not be created with xconnect under SVI, even if N-PE pseudo-wire on SVI is enabled.
- Prime Fulfillment supports a hybrid configuration for EVC service requests. In a hybrid configuration, the forwarding commands (such as xconnect) for one side of an attachment circuit can be configured under a service instance, and the xconnect configuration for the other side of the attachment circuit can be configured under a switch virtual interface (SVI).
- N-PE Pseudo-wire on SVI is applicable for all connectivity types (PSEUDOWIRE, VPLS, and LOCAL), but a hybrid SVI configuration is possible only for pseudowire connectivity.
- When MPLS Core Connectivity Type is set as VPLS, the N-PE Pseudo-wire on SVI attribute is always enabled in the policy and service request.
- When MPLS Core Connectivity Type is set as LOCAL connectivity type, the N-PE Pseudo-wire on SVI attribute is always disabled in the policy and service request.
- For examples of these cases, see configlet examples EVC (Pseudowire Core Connectivity, Bridge Domain, Pseudowire on SVI), page 3-212 and EVC (Pseudowire Core Connectivity, no Bridge Domain, no Pseudowire on SVI), page 3-213.
- For additional information on the N-PE Pseudo-wire on SVI attribute, see the corresponding coverage in the EVC policy section in the section Setting the Interface Attributes, page 3-28.
- The N-PE Pseudo-wire on SVI attribute is not supported for IOS XR devices. All the xconnect commands are configured on L2 subinterfaces.

Step 32 Check the **PW Tunnel Selection** check box if you want to be able to manually select the Traffic Engineering (TE) tunnel for the pseudowire connecting point-to-point N-PEs.

Usage notes:

- Checking the PW Tunnel Selection check box activates the Interface Tunnel attribute field (see the next step).
- This attribute only appears if the MPLS core connectivity type is set as pseudowire in the EVC policy.
- The PW Tunnel Selection attribute is not supported for IOS XR devices.

Step 33 If you checked the PW Tunnel Selection check box, enter the TE tunnel ID in the **Interface Tunnel** text field.

Usage notes:

- Prime Fulfillment uses the tunnel information to create and provision a pseudowire class that describes the pseudowire connection between two N-PEs. This pseudowire class can be shared by more than one pseudowire, as long as the pseudowires share the same tunnel ID and remote loopback address. During service request creation, Prime Fulfillment does not check the validity of the tunnel ID number. That is, Prime Fulfillment does not verify the existence of the tunnel.
- The Interface Tunnel attribute is not supported for IOS XR devices.

Step 34 Check the **AutoPick Bridge Group Name** check box to have Prime Fulfillment autopick the bridge group name during service request creation.

If this check box is unchecked, you are prompted to specify a bridge group name during service request creation (see the next step).

Usage notes:

- This attribute only displays for IOS XR devices.
- If the AutoPick Bridge Group Name check box is unchecked, enter an bridge group name in the **Bridge Group Name** text field.
- The AutoPick Bridge Group Name and Bridge Group Name attributes only appear if Configure Bridge Domain was enabled in the EVC Service Request Editor window earlier in the service request workflow.

Step 35 Check the **AutoPick Bridge Domain/VLAN ID** check box to have Prime Fulfillment autopick the VLAN ID during service request creation.

If this check box is unchecked, you are prompted to specify a VLAN ID during service request creation (see the next step).

Usage notes:

- AutoPick Bridge Domain/VLAN ID consumes a global VLAN ID on the device.
- The bridge domain VLAN ID is picked from the existing Prime Fulfillment VLAN pool.
- The AutoPick Bridge Domain/VLAN ID attribute appears for both Cisco 7600 and ASR 9000 devices. It will be displayed only for non-EVC links.

Step 36 If the AutoPick Bridge Domain/VLAN ID check box is unchecked, enter an ID number in the **Bridge Domain/VLAN ID** text field.

Usage notes:

- If AutoPick Bridge Domain/VLAN ID is checked, this field is non-editable.

- When a VLAN ID is manually allocated, Prime Fulfillment verifies the VLAN ID to see if it lies within Prime Fulfillment's VLAN ID pool. If the VLAN ID is in the pool but not allocated, the VLAN ID is allocated to the service request. If the VLAN ID is in the pool and is already in use, Prime Fulfillment prompts you to allocate a different VLAN ID. If the VLAN ID lies outside of the Prime Fulfillment VLAN ID pool, Prime Fulfillment does not perform any verification about whether the VLAN ID allocated. The operator must ensure the VLAN ID is available.
- The Bridge Domain/VLAN ID text field appears for both Cisco 7600 and ASR 9000 devices. It will be displayed only for non-EVC links.

Step 37 Check the **AutoPick Bridge Domain Name** check box to have Prime Fulfillment autopick the bridge domain name during service request creation.

If this check box is unchecked, you are prompted to specify a bridge domain name during service request creation (see the next step).

Usage notes:

- The AutoPick Bridge Domain Name attribute appears only for Cisco ASR 9000 devices.
- The AutoPick Bridge Domain Name attribute only appears if Configure Bridge Domain was enabled in the EVC Service Request Editor window earlier in the service request workflow.

Step 38 If the AutoPick Bridge Domain Name check box is unchecked, enter a bridge domain name in the **Bridge Domain Name** text field.

Usage notes:

- Bridge Domain Name field appears only for Cisco ASR 9000 devices.
- The Bridge Domain Name attribute only appears if Configure Bridge Domain was enabled in the EVC Service Request Editor window earlier in the service request workflow.

Step 39 Check the **Use PseudoWireClass** check box to enable the selection of a pseudowire class.

This attribute is unchecked by default.

Usage notes:

- The pseudowire class name is used for provisioning pw-class commands on IOS XR devices. See Creating and Modifying Pseudowire Classes, page 3-14 for additional information on pseudowire class support for IOS XR devices.
- If Use PseudoWireClass is checked, an additional attribute, **PseudoWireClass**, appears in the GUI. Click the **Select** button of PseudoWireClass attribute to choose a pseudowire class previously created in Prime Fulfillment.
- The Use PseudoWireClass attribute is only available if the MPLS core connectivity type was set as PSEUDOWIRE in the Service Options window (see Setting the Service Options, page 3-20).
- Use PseudoWireClass is only applicable for IOS XR devices.
- The Use PseudoWireClass and PseudoWireClass attributes only appear if Configure Bridge Domain was not enabled in the EVC Service Request Editor window earlier in the service request workflow.

Step 40 For **L2VPN Group Name** choose one of the following from the drop-down list:

- **ISC**
- **VPNSC**

Usage notes:

- This attribute is used for provisioning the L2VPN group name on IOS XR devices.

Note The choices in the drop-down list are derived from a configurable DCPL property. For information about how to define the L2VPN Group Name choices available in the drop-down list, see Defining L2VPN Group Names for IOS XR Devices, page 3-18.

- The L2VPN Group Name attribute is not available if the MPLS core connectivity type was set as VPLS in the Service Options window (see Setting the Service Options, page 3-20).
- L2VPN Group Name is only applicable for IOS XR devices.
- The L2VPN Group Name attribute only appears if Configure Bridge Domain was not enabled in the EVC Service Request Editor window earlier in the service request workflow.

Step 41 Enter an **E-Line Name** to specify the point-to-point (p2p) E-line name.

Usage notes:

- If no value is specified for the **E-Line Name**, Prime Fulfillment autogenerates a default name as follows:
  - For PSEUDOWIRE core connectivity type, the format is:

    DeviceName--VC_ID
  - For LOCAL core connectivity type, the format is:

    DeviceName--VLAN_ID

  If the default name is more than 32 characters, the device names are truncated.
- The E-Line Name attribute is not available if the MPLS core connectivity type was set as VPLS in the Service Options window (see Setting the Service Options, page 3-20).
- E-Line Name is only applicable for IOS XR devices.
- The E-Line Name attribute only appears if Configure Bridge Domain was not enabled in the EVC Service Request Editor window earlier in the service request workflow.

Step 42 Click **OK** to save the Standard UNI settings and return to the EVC SR window.

The value in the Link Attributes column now displays as “Changed,” signifying that the link settings have been updated. You can edit the link attributes now or at a future time by clicking on the Changed link and modifying the settings in the Standard UNI Details window.

See Modifying the EVC Service Request, page 3-53 for details on editing the link attributes.

Step 43 To add another link click the **Add** button and set the attributes for the new link as in the previous steps in this section.

Step 44 To delete a link, check the check box in the first column of the row for that link and click the **Delete** button.

Step 45 If you want to set up links with L2 access nodes for this service request, see Setting Links with L2 Access Nodes, page 3-52.

Step 46 When you have completed setting the attributes in the EVC Service Request Editor window, click the **Save** button at the bottom of the window to save the settings and create the EVC service request.

If any attributes are missing or incorrectly set, Prime Fulfillment displays a warning in the lower left of the window. Make any corrections or updates needed (based on the information provided by Prime Fulfillment), and click the **Save** button.

For information on modifying an EVC service request see the section Modifying the EVC Service Request, page 3-53. For additional information about saving an EVC service request, see Saving the EVC Service Request, page 3-54.

Setting Links with L2 Access Nodes

The Links with L2 Access Nodes section of the EVC Service Request Editor window allows you to set up links with L2 (Ethernet) access nodes. These are similar to direct connect links, except that they have L2/Ethernet access nodes beyond the N-PE (towards the CE). Therefore, NPCs are involved. The steps for setting up links with L2 access nodes are similar to those covered in the section Setting Direct Connect Links, page 3-41. See that section for detailed steps on the following common operations:

- Adding and deleting links.
- Selecting the N-PE.
- Choosing the UNI interface.
- Setting the link as an EVC link.
- Editing the standard and EVC link attributes.

The main difference in setting up links with L2 access does is specifying the NPC details.

To set the NPC details for links with L2 access nodes, perform the following steps.

Step 1 The first step in the process of adding a link using NPCs is selecting the U-PE/PE-AGG device, rather than the N-PE.

If only one NPC exists for the chosen interface, that NPC is autopopulated in the Circuit Details column, and you need not choose it explicitly.

If more then one NPC is available, click **Select one circuit** in the Circuit Selection column. The NPC window appears, enabling you to choose the appropriate NPC.

Step 2 Click **OK**.

Each time you choose a PE and its interface, the NPC that was set up from this PE and interface is automatically displayed under **Circuit Selection**. This means that you do not have to further specify the PE to complete the link.

If you want to review the details of this NPC, click **Circuit Details** in the Circuit Details column. The NPC Details window appears and lists the circuit details for this NPC.

Step 3 For details about editing link attributes, adding or deleting links, or using the EVC check box, see the corresponding steps in the section Setting Direct Connect Links, page 3-41.

Step 4 When you have completed setting the attributes in the EVC Service Request Editor window, click the **Save** button at the bottom of the window to save the settings and create the EVC service request.

If any attributes are missing or incorrectly set, Prime Fulfillment displays a warning in the lower left of the window. Make any corrections or updates needed (based on the information provided by Prime Fulfillment), and click the **Save** button.

For information on modifying an EVC service request see the section Modifying the EVC Service Request, page 3-53. For additional information about saving an EVC service request, see Saving the EVC Service Request, page 3-54.

Modifying the EVC Service Request

You can modify an EVC service request if you must change or modify the links or other settings of the service request.

To modify an EVC service request, perform the following steps.

Step 1 Choose **Operate > Service Requests > Service Request Manager**.

The Service Request Manager window appears, showing service requests available in Prime Fulfillment.

Step 2 Check a check box for a service request.

Step 3 Click **Edit**.

EVC Service Request Editor window appears.

Step 4 Modify any of the attributes, as desired.

See the sections start with "Setting the Service Request Details" section on page 3-35 for detailed coverage of setting attributes in this window.

Note Once the VC ID, VPLS VPN ID, and VLAN ID have been set in a service request they cannot be modified.

Step 5 To add a template/data file to an attachment circuit, see the section Using Templates and Data Files with an EVC Ethernet Service Request, page 3-53.

Step 6 When you are finished editing the EVC service request, click **Save**.

For additional information about saving an EVC service request, see Saving the EVC Service Request, page 3-54.

Using Templates and Data Files with an EVC Ethernet Service Request

Prime Fulfillment does not support configuration of all the available CLI commands on a device being managed by the application. In order to configure such commands on the devices, you can use Prime Fulfillment Template Manager functionality. Templates can be associated at the policy level on a per-device role basis. Templates can be overridden at service request level, if the policy-level setting permits the operator to do so.

To associate templates and data files in a service request select any link in the Service Request Editor window and click the **Template** button at the bottom of the window.

Note If the template feature has not been enabled in the associated policy then the Template button will not be available for selection.

The SR Template Association window appears. In this window, you can associate templates at a per-device level. The SR Template Association window lists the devices comprising the link, the device roles, and the template(s)/data file(s) associated with the devices. In this case, the template(s)/data file(s) have not yet been set up.

For further instructions on how to associate templates and data files with a service request, see Using Templates with Service Requests, page 9-24.

Saving the EVC Service Request

To save an EVC Ethernet service request, perform the following steps.

Step 1 When you have finished setting the attributes for the EVC Ethernet service request, click **Save** to create the service request.

If the EVC service request is successfully created, you will see the Service Request Manager window. The newly created EVC Ethernet service request is added with the state of REQUESTED.

Step 2 If, however, the EVC Ethernet service request creation failed for some reason (for example, a value chosen is out of bounds), you are warned with an error message.

In such a case, you should correct the error and save the service request again.

Step 3 If you are ready to deploy the EVC Ethernet service request, see Deploying Service Requests, page 8-10.

Creating an EVC ATM-Ethernet Interworking Policy

This section contains an overview of EVC ATM-Ethernet Interworking support in Prime Fulfillment, as well as the basic steps to create an EVC ATM-Ethernet Interworking policy. It contains the following subsections:

- Defining the EVC Ethernet Policy, page 3-18
- Setting the Service Options, page 3-20
- Setting the ATM Interface Attributes, page 3-58
- Setting the EVC Attributes, page 3-22
- Setting the Interface Attributes, page 3-28
- Enabling Template Association, page 3-33

For information on creating EVC ATM-Ethernet service requests, see Managing an EVC ATM-Ethernet Interworking Service Request, page 3-69.

Note For a general overview of EVC support in Prime Fulfillment, see the chapter “Layer 2 Concepts” in the *Cisco Prime Fulfillment Theory of Operations Guide 6.2*.

Defining the EVC ATM-Ethernet Interworking Policy

You must define an EVC ATM-Ethernet Interworking policy before you can provision a service. A policy can be shared by one or more service requests that have similar service requirements.

A policy is a template of most of the parameters needed to define an EVC service request. After you define it, an EVC policy can be used by all the EVC service requests that share a common set of characteristics. You create a new EVC policy whenever you create a new type of service or a service with different parameters. EVC policy creation is normally performed by experienced network engineers.

To define an EVC ATM-Ethernet Interworking policy, you start by setting the service type attributes. To do this, perform the following steps.

Step 1 Choose **Service Design > Policies > Policy Manager**.

The Policy Manager window appears.

Step 2 Click **Create**.

The Policy Editor window appears.

Step 3 Choose **EVC** from the Policy Type drop-down list.

The Policy Editor window appears.

Step 4 Enter a **Policy Name** for the EVC ATM-Ethernet Interworking policy.

Step 5 Choose the **Policy Owner** for the EVC policy.

There are three types of EVC policy ownership:

- Customer ownership
- Provider ownership
- Global ownership—Any service operator can make use of this policy.

This ownership has relevance when the Prime Fulfillment Role-Based Access Control (RBAC) comes into play. For example, an EVC policy that is customer-owned can only be seen by operators who are allowed to work on this customer-owned policy. Similarly, operators who are allowed to work on a provider's network can view, use, and deploy a particular provider-owned policy.

Step 6 Click **Select** to choose the owner of the EVC policy.

The policy owner was established when you created customers or providers during Prime Fulfillment setup. If the ownership is global, the Select function does not appear.

Step 7 Choose the **Policy Type**.

The choices are:

- **ETHERNET**
- **ATM-Ethernet Interworking**

Note This section describes creating the ATM-Ethernet Interworking policy type. For information on using the EVC ETHERNET policy type, see Creating an EVC Ethernet Policy, page 3-18.

Step 8 Click **Next**.

The Service Options window appears.

Step 9 Continue with the steps contained in the next section, Setting the Service Options, page 3-20.

Setting the Service Options

This section describes how to set the service options for the EVC policy.

To set the EVC service options, perform the following steps.

Step 1 Check the **CE Directly Connected to EVC** check box if the CEs are directly connected to the N-PE.

This check box is not checked by default.

Note The **Editable** check box gives you the option of making a field editable. If you check the **Editable** check box, the service operator who is using this EVC policy can modify the editable parameter during EVC service request creation.

Usage notes:

- If the check box is checked, a service request created using this policy can have only directly connected links. No Ethernet access nodes will be involved.
- If the check box is unchecked, a service request created using this policy might or might not have Ethernet access nodes in the links.
- When a CE is directly connected to the N-PE, NPCs are not applicable to the link while creating service requests.
- When a CE is not directly connected to the N-PE, NPCs are used during service request creation, as per standard Prime Fulfillment behavior. There is no change in NPC implementation to support EVC functionality.

Step 2 Check the **All Links Terminate on EVC** check box if all links need to be configured with EVC features.

This check box is not check by default. Usage notes:

- If the check box is checked, a service request created using such policy will have all links using the EVC feature.
- If the check box is unchecked, zero or more links can use the EVC feature. This ensures that existing platforms can still be used in one or more links while delivering the services. This allows the possibility of a link with EVC support being added in the future.

  Note If the check box is unchecked, in the service request creation process the user must indicate whether or not the created link is EVC or non-EVC.

- If no links are expected to use the EVC feature even in the future (for example, if the provider is not planning to upgrade to the EVC infrastructure for the service that is being created), existing Prime Fulfillment policy types (L2VPN or VPLS) can be used instead of EVC.

Step 3 Choose an **MPLS Core Connectivity Type** from the drop-down list.

Note The core option supports MPLS only. There is no L2TPv3 support for this service.

The choices are:

- **PSEUDOWIRE**—Choose this option to allow connectivity between two N-PEs across the MPLS core. This option does not limit the service to point-to-point (E-Line). This is because even with the PSEUDOWIRE option selected, there can still be multiple CEs connected to a bridge domain on one or both sides of the pseudowire.
- **LOCAL**—Choose this option for local connect cases in which there is no connectivity required across the MPLS core.

  Local connect supports the following scenarios:

  – All interfaces on the N-PE are EVC-capable and using the EVC infrastructure. This is configured by associating all of the customer traffic on these interfaces to a bridge domain. This consumes a VLAN ID on the N-PE (equal to the bridge domain ID).

  – Some interfaces on the N-PE are EVC-capable, while others are switch-port-based. In such cases, all of the customer traffic on the interfaces that are configured with the EVC infrastructure are associated to a bridge domain. The traffic on the non-EVC interfaces (and all the access nodes/interfaces beyond this N-PE) are configured with the Service Provider VLAN ID, where the Service Provider VLAN ID is the same as the bridge domain ID for the EVC-based services.

  – Only two interfaces on the N-PE are involved, and both are based on EVC-capable line cards. In the first case, the operator might choose not to configure the bridge domain option. In this case, the **connect** command that is used for the local connects are used, and the global VLAN is conserved on the device. If the operator chooses to configure with the bridge domain option, both interfaces are associated to a bridge domain ID, so that additional local links can be added to the service in future. This consumes a VLAN ID (bridge domain ID) on the N-PE.
- **VPLS**—This option is not supported for EVC ATM-Ethernet Interworking policies and services requests.

Note Attributes available in subsequent windows of the policy workflow dynamically change based on the choice made for the MPLS Core Connectivity Type (PSEUDOWIRE or LOCAL). For completeness, all attributes available for the different core types are documented in the following steps. Attributes apply to all core types, unless otherwise noted.

Note Also, some attributes are supported only on IOS or IOS XR platforms. Attributes apply to both platforms, unless otherwise noted. All platform-specific attributes are visible in the policy workflow windows. Later, when a service request is created based on the policy (and specific devices are associated with the service request), platform-specific attributes are filtered from service request windows, depending on the device type (IOS or IOS XR).

Step 4 Check the **Configure With Bridge Domain** check box to determine bridge domain characteristics.

The behavior of the Configure With Bridge-Domain option works in tandem with the choice you selected in the MPLS Core Connectivity Type option, as follows.

- **PSEUDOWIRE** as the MPLS Core Connectivity Type. There are two cases:

  A. With EVC:

  – If **Configure With Bridge Domain** is checked, the policy configures pseudowires under SVIs associated to the bridge domain.

  – If **Configure With Bridge Domain** is unchecked, the policy will configure pseudowires directly under the service instance. This conserves the global VLAN.

B. Without EVC:

- If **Configure With Bridge Domain** is checked, the policy configures pseudowires as in L2VPN services (with SVIs).
- If **Configure With Bridge Domain** is unchecked, the policy configures pseudowires directly under subinterfaces.

Only pseudowires can be either configured directly under service instance of the corresponding EVC-capable interface or under SVIs associated to the bridge domain.

- **LOCAL** as the MPLS Core Connectivity Type:
  - If **Configure With Bridge Domain** is checked, the policy allows either point-to-point or multipoint local connect services.
  - If **Configure With Bridge Domain** is unchecked, Prime Fulfillment allows only point-to-point local connects without bridge domain.

Step 5 Click **Next**.

ATM Interface Attribute window appears.

Step 6 Continue with the steps contained in the next section, Setting the ATM Interface Attributes, page 3-58.

Setting the ATM Interface Attributes

This section describes how to set the ATM Interface attributes for the EVC ATM-Ethernet Interworking policy.

To set the ATM interface attributes, perform the following steps.

Step 1 Choose the **Transport Mode** from the drop-down list.

The choices are:

- **VP**—Virtual path mode. This is the default.
- **VC**—Virtual circuit mode.

Step 2 Choose the **ATM Encapsulation** from the drop-down list.

- **AAL5SNAP**

Step 3 Click **Next**.

The EVC Attribute window appears.

Step 4 Continue with the steps contained in the next section, Setting the EVC Attributes, page 3-22.

Setting the EVC Attributes

This section describes how to set the EVC attributes for the EVC ATM-Ethernet Interworking policy.

EVC attributes are organized under the following categories:

- Service Attributes

- VLAN Match Criteria
- VLAN Rewrite Criteria

The following sections describe how to set the options under each category.

Setting the Service Attributes

To set the EVC service attributes, perform the following steps.

Step 1 Check the **AutoPick Service Instance ID** check box to specify that the service instance ID will be autogenerated and allocated to the link during service request creation.

If the check box is unchecked, while setting the Prime Fulfillment link attributes during service request creation, Prime Fulfillment will prompt the operator to specify the service instance ID.

Usage notes:

- The service instance ID represents an Ethernet Flow Point (EFP) on an interface in the EVC infrastructure. The service instance ID is locally significant to the interface. This ID has to be unique only at the interface level. The ID must be a value from 1 to 8000.
- There are no resource pools available in Prime Fulfillment from which to allocate the service instance IDs.
- It is the responsibility of the operator creating the service request to maintain the uniqueness of the ID at the interface level.

Step 2 Check the **AutoPick Service Instance Name** check box to have Prime Fulfillment autogenerate a service instance name when you create a service request based on the policy. The autogenerated value is in the following pattern: *CustomerName_ServiceRequestJobID*.

If the check box is unchecked, then you can enter a value during service request creation.

Step 3 Check the **Enable PseudoWire Redundancy** check box to enable pseudowire redundancy (alternative termination device) under certain conditions.

Usage notes:

- Enable Pseudo Wire Redundancy is only available if the MPLS Core Connectivity Type was set as PSEUDOWIRE in the Service Options window (see Setting the Service Options, page 3-20).

Step 4 Check the **AutoPick VC ID** check box to have Prime Fulfillment autopick the VC ID during service request creation.

If this check box is unchecked, the operator will be prompted to specify a VC ID during service request creation.

Usage notes:

- When AutoPick VC ID is checked, Prime Fulfillment allocates a VC ID for pseudowires from the Prime Fulfillment-managed VC ID resource pool.

Step 5 Check the **AutoPick Bridge Domain/VLAN ID** check box to have Prime Fulfillment autopick the VLAN ID for the service request during service request creation.

If this check box is unchecked, the operator will be prompted to specify a VLAN ID during service request creation.

Usage notes:

- AutoPick Bridge Domain/VLAN ID consumes a global VLAN ID on the device.

- The bridge domain/VLAN ID is picked from the existing Prime Fulfillment VLAN pool. Once the VLAN ID is assigned in the service request, Prime Fulfillment makes the VLAN ID unavailable for subsequent service requests.
- In the case of manual VLAN ID allocation, Prime Fulfillment does not manage the VLAN ID if the ID lies outside the range of an Prime Fulfillment-managed VLAN pool. In this case, the operator must ensure the uniqueness of the ID in the Ethernet access domain. If an operator specifies a VLAN ID that is within the range of an Prime Fulfillment-managed VLAN pool and the VLAN ID is already in use in the access domain, Prime Fulfillment displays an error message indicating that the VLAN ID is in use.

Note on Access VLAN IDs

An access VLAN ID is of local significance to the EVC-capable ports. It should not be confused with the global VLANs. This can be visualized as a partitioning of the Ethernet access network beyond the EVC ports into several subEthernet access domains (one each for an EVC-capable port).

However, all the service interfaces on the Ethernet access nodes beyond the EVC ports will have this very same VLAN ID for a link. This ID must be manually specified by the operator when setting the link attributes during service request creation. The operator must ensure the uniqueness of the ID across the EVC-demarcated Ethernet access domain.

These VLAN IDs are not managed by Prime Fulfillment by means of locally-significant VLAN pools. But once a VLAN ID is assigned for a link in the service request, Prime Fulfillment makes the VLAN unavailable for subsequent service requests within the Ethernet access domain demarcated by the EVC. Likewise, if a manually-specified VLAN is already in use in the access domain delimited by the EVC, Prime Fulfillment will display an error message indicating that the new VLAN ID being specified is already in use on the NPC. The operator will be prompted to specify a different VLAN ID, which will be provisioned on the L2 access nodes.

Step 6 Continue with the steps contained in the next section, Setting the VLAN Matching Criteria Attributes, page 3-25.

Setting the VLAN Matching Criteria Attributes

Prior to the introduction of the EVC capability, service providers could either deploy service-multiplexed services (ERS/ERMS or EVPL/EVCS) or service-bundled services on a single port. Both could not be supported simultaneously due to the limitations in the infrastructure, which only allowed matching the outer-most VLAN tag.

One of the key benefits of EVC support in Prime Fulfillment is to provide a flexible means to examine the VLAN tags (up to two levels) of the incoming frames and associate them to appropriate Ethernet Flow Points (EFPs). This allows service providers to deploy simultaneously both the service-multiplexed and service-bundled services on a single port.

To set the EVC VLAN matching criteria attributes, perform the following steps.

Step 1 Check the **Both Tags** check box to enable service requests created with the policy to match both the inner and outer VLAN tags of the incoming frames.

If you do not check this check box, service requests created with the policy will match only the outer VLAN tag of the incoming frames.

Checking the Both Tags attribute causes the Inner VLAN Ranges attribute (covered in the next steps) to appear in the EVC Attribute window.

Step 2 Check the **Inner VLAN Ranges** check box to enable the range of inner VLAN tags to be specified during service request creation.

If the check box is unchecked, the range of inner VLAN tags are not allowed. In this case, the operator must specify discrete VLAN IDs during service request creation.

Step 3 Check the **Outer VLAN Ranges** check box to enable the range of outer VLAN tags to be specified during service request creation.

If the check box is unchecked, the range of outer VLAN tags are not allowed. In this case, the operator must specify discrete VLAN IDs during service request creation.

Step 4 Check the **AutoPick Outer VLAN** check box to have Prime Fulfillment autopick the outer VLAN ID from a previously created outer VLAN ID resource pool during service request creation.

If this check box is unchecked, the operator will be prompted to specify an outer VLAN ID during service request creation.

Note Use of the AutoPick Outer VLAN attribute requires that two elements have already been set up in Prime Fulfillment. One is an Interface Access Domain, which is a logical element that groups the physical ports of an N-PE device. The other is an EVC Outer VLAN resource pool, which is used by the Interface Access Domain. For instructions on how to set up these elements, see the sections Setting Up Resources, page 2-40, and Resource Pools, page 2-44.

Usage notes:

- AutoPick Outer VLAN can be used for interfaces that support EVC functionality.
- AutoPick Outer VLAN consumes a VLAN ID on the interface that supports EVC.
- The bridge domain VLAN ID is picked from the existing Prime Fulfillment VLAN pool.

Step 5 Continue with the steps contained in the next section, Setting the VLAN Rewrite Criteria Attributes, page 3-26.

Setting the VLAN Rewrite Criteria Attributes

Together with VLAN matching criteria, VLAN rewrite makes the EVC infrastructure very powerful and flexible. The following VLAN rewrite options are supported:

- Pop one or two tags.
- Push one or two tags.
- Translation (1:1, 2:1, 1:2, 2:2).

Be aware of the following considerations when setting the VLAN rewrite criteria attributes:

- Only one kind of rewrite can be done on every CE-facing EVC link.
- All VLAN rewrites are done using the **symmetric** keyword on the ingress traffic (for example, **rewrite ingress tag pop 2 symmetric**).
- For any service instance, only one type of rewrite option (pop, push, or translate) is allowed per instance. For example, if pop out is enabled, push inner, push outer, translate inner, and translate outer are not available.

To set the EVC VLAN rewrite criteria attributes, perform the following steps.

Step 1 Check the **Pop Outer** check box to pop the outer VLAN ID tag of the incoming frames that fulfill the match criteria.

If this check box is unchecked, the outer tag of the incoming traffic is not popped.

Step 2 Check the **Pop Inner** check box to pop the inner VLAN ID tag of the incoming frames that fulfill the match-criteria.

If this check box is unchecked, the inner tag is not popped. Note that, if Pop Inner is checked, Pop Outer is automatically checked.

Step 3 Check the **Push Outer** check box to impose an outer VLAN ID tag onto the incoming frames that fulfill the match criteria.

If this check box is unchecked, no outer tag is imposed on the incoming frames.

Usage notes:

- If Push Outer is checked, all service requests created with the policy push a dot1q outer tag on the incoming frames matching the match criteria. When creating the link during service creation, the operator can specify an outer tag with a value from 1 to 4096.
- This attribute is available regardless of the number of tags used in the match criteria. Whether the incoming traffic is double tagged or single tagged, if Push Outer is enabled, all corresponding service requests push an outer tag. All subsequent nodes consider only the outer-most two tags (if EVC-capable) or just one tag (not EVC-capable) and treat the inner-most tags transparently as payload.
- This VLAN ID is not derived from Prime Fulfillment-managed VLAN ID pools.

Step 4 Check the **Push Inner** check box to impose an inner VLAN ID tag onto the incoming frames that fulfill the match criteria.

This operation pushes both an inner and an outer tag onto the incoming packet, not just an inner tag. If this check box is unchecked, no inner tag is imposed on the incoming frames.

Usage notes:

- If Push Inner is checked, all service requests created with the policy push a dot1q inner tag on the incoming frames matching the match criteria. When creating the link during service creation, the operator can specify an inner tag with a value from 1 to 4096.
- If Push Inner is checked, Push Outer is automatically checked.
- This attribute is available regardless of the number of tags used in the match criteria. Regardless of whether the incoming traffic is double tagged or single tagged, if Push Inner is enabled, all corresponding service requests push an inner tag. All subsequent nodes consider only the outer-most two tags (if EVC-capable) or just one tag (not EVC-capable) and treat the inner-most tags transparently as payload.
- This VLAN ID is not derived from Prime Fulfillment-managed VLAN ID pools.

Step 5 Check the **Translate Outer** check box to allow the operator to specify a target outer VLAN ID during service request creation.

The outer tag of all the incoming frames that fulfill the match criteria are translated to this ID. If the check box is unchecked, no outer tag translation is performed. See Table 3-4.

Step 6 Check the **Translate Inner** check box to allow the operator to specify a target inner VLAN ID during service request creation.

The inner tag of all the incoming frames that fulfill the match criteria are translated to this ID. If the check box is unchecked, no inner tag translation is performed. See Table 3-4.

Note Table 3-4 summarizes the realization of different VLAN translations available in the EVC infrastructure. The second and third columns (Match Outer Tag and Match Inner Tag) refer to policy settings. The last two columns (Translate Outer Tag and Translate Inner Tag) indicate the VLAN translation that occurs on the incoming frames.

[

Table 3-4 VLAN Translation Summary Table

Type	Match Outer Tag	Match Inner Tag	Translate Outer Tag	Translate Inner Tag	Push Outer Tag
1:1	True	N/A	Yes	No	N/A
1:2	True	N/A	N/A	N/A	Yes
2:1	True	True	Yes	No	N/A
2:2	True	True	Yes	Yes	N/A

Step 7 Click **Next**.

The Interface Attribute window appears.

Step 8 Continue with the steps contained in the next section, Setting the Interface Attributes, page 3-28.

Setting the Interface Attributes

This step of creating the EVC ATM-Ethernet Interworking policy involves setting the interface attributes, as shown in the Interface Attribute window. The attributes you can configure in this window are grouped under the following categories:

- UNI Information
- VLAN
- Pseudowire
- ACL
- Security
- UNI Storm Control
- Protocol

In some cases, checking an attribute causes additional attributes to appear in the GUI. This is covered in the steps that follow.

Note If the CE is directly connected to an N-PE, only speed, duplex, UNI shutdown, and other generic options are presented. In this case, port security, storm control, L2 protocol tunneling, and other advanced features are not supported due to the current platform limitations. If these features are needed for a service, the service provider must deploy Layer 2 Ethernet access nodes beyond the EVC to support these requirements.

Note Attributes available in the Interface Attributes window dynamically change based on the choice made for the MPLS Core Connectivity Type (PSEUDOWIRE or LOCAL) in the Service Options window (see Setting the Service Options, page 3-20). For completeness, all attributes available for the different core types are documented in the following steps. Attributes apply to all core types, unless otherwise noted.

To set the EVC interface attributes, perform the following steps.

Step 1 Check the **Standard UNI Port** check box to enable port security.

This is the default. When you uncheck the check box, the port is treated as an uplink with no security features, and the window dynamically changes to eliminate items related to port security.

Note When the UNI is configured on an N-PE device running IOS XR, the Standard UNI Port attribute is not supported. All the CLIs related to Standard UNI Port and UNI Port Security are ignored in this case.

Step 2 Check the **UNI Shutdown** check box if you want to leave the UNI port shut during service activation, for example, when the service provider wants to deploy a service in the network but wants to activate it at a later time.

Step 3 Check the **Keep Alive** check box to configure keepalives on the UNI port.

By default, this check box is unchecked, which causes the command **no keepalive** to be provisioned on the UNI port. This prevents a CPE from sending keepalive packets to the U-PE, for security purposes. This attribute is editable, in order to support modification on a per-service request basis.

Step 4 Enter a **Link Media** (optional) of None, auto-select, rj45, or sfp.

Step 5 Enter a **Link Speed** (optional) of None, 10, 100, 1000, Auto, or nonegotiate.

Step 6 Enter a **Link Duplex** (optional) of None, Full, Half, or Auto.

Step 7 Choose an **Encapsulation** type.

The choices are:

- **DOT1QTRUNK**—Configures the UNI as a trunk with 802.1q encapsulation. If the UNI belongs to a directly connected and EVC link, this setting signifies that the incoming frames are 802.1q encapsulated and that they match the VLAN ID configured for the link. This specific topology does not involve a trunk UNI as such.
- **DOT1QTUNNEL**—Configures the UNI as an 802.1q tunnel (also known as a dot1q tunnel or Q-in-Q) port.
- **ACCESS**—Configures the UNI as an access port.

Step 8 Specify the type of **VLAN Translation** for this policy by clicking the appropriate radio button.

The choices are:

- **No**—No VLAN translation is performed. (This is the default.)
- **1:1**—1:1 VLAN translation. Translates an incoming customer VLAN to another.
- **2:1**—2:1 VLAN translation. Converts both inner and outer VLANs to a single VLAN.
- **1:2**—1:2 VLAN translation. Pushes one more provider VLAN.
- **2:2**—2:2 VLAN translation. Translates both inner and outer VLANs to two other VLANs.

Note For more details on how VLAN translation is supported in EVC ATM-Ethernet services, see the coverage of the VLAN Translation attribute in Managing an EVC ATM-Ethernet Interworking Service Request, page 3-69.

Step 9 Check the **Use PseudoWireClass** check box to enable the selection of a pseudowire class.

This attribute is unchecked by default.

Usage notes:

- The pseudowire class name is used for provisioning pw-class commands on IOS XR devices. See Creating and Modifying Pseudowire Classes, page 3-14 for additional information on pseudowire class support for IOS XR devices.
- If **Use PseudoWireClass** is checked, an additional attribute, **PseudoWireClass**, appears in the GUI. Click the **Select** button of PseudoWireClass attribute to choose a pseudowire class previously created in Prime Fulfillment.
- The Use PseudoWireClass attribute is only available if the MPLS core connectivity type was set as PSEUDOWIRE in the Service Options window (see Setting the Service Options, page 3-20).
- Use PseudoWireClass is only applicable for IOS XR devices.

Step 10 For **L2VPN Group Name** choose one of the following from the drop-down list:

- **ISC**
- **VPNSC**

Usage notes:

- This attribute is used for provisioning the L2VPN group name on IOS XR devices.

  Note The choices in the drop-down list are derived from a configurable DCPL property. For information about how to define the L2VPN Group Name choices available in the drop-down list, see Defining L2VPN Group Names for IOS XR Devices, page 3-18.

- L2VPN Group Name is only applicable for IOS XR devices.

Step 11 Enter an **E-Line Name** to specify the point-to-point (p2p) E-line name.

Usage notes:

- If no value is specified for the **E-Line Name** in either the policy or the service request based on the policy, Prime Fulfillment autogenerates a default name as follows:
  - For PSEUDOWIRE core connectivity type, the format is:

    DeviceName--VC_ID
  - For LOCAL core connectivity type, the format is:

    DeviceName--VLAN_ID

  If the default name is more than 32 characters, the device names are truncated.
- E-Line Name is only applicable for IOS XR devices.

Step 12 Check the **PW Tunnel Selection** check box if you want to be able to manually select the Traffic Engineering (TE) tunnel for the pseudowire connecting point-to-point N-PEs.

This attribute is unchecked by default.

Subsequently, when you create a service request based on this policy, you must specify the TE tunnel ID in a field provided. Prime Fulfillment uses the tunnel information to create and provision a pseudowire class that describes the pseudowire connection between two N-PEs. This pseudowire class can be shared by more than one pseudowire, as long as the pseudowires share the same tunnel ID and remote loopback address. You are responsible to ensure that the tunnel interface and associated ID are configured. During service request creation when you specify the tunnel ID number, Prime Fulfillment does not check the validity of the value. That is, Prime Fulfillment does not verify the existence of the tunnel.

Step 13 Check the **N-PE Pseudo-wire on SVI** check box to have Prime Fulfillment generate forwarding commands under SVIs (switch virtual interfaces).

By default, this check box is not checked. In this case, Prime Fulfillment generates forwarding commands under the service instance.

For an EVC link, the attribute N-PE Pseudo-wire on SVI is dependent on the value of the attribute Configure with Bridge Domain (this is available in the policy workflow in the EVC Policy Editor - Service Options window). N-PE Pseudo-wire on SVI, if enabled, will be reflected only when Configure with Bridge Domain is set to true. Otherwise, the service request will not be created with xconnect under SVI, even if N-PE Pseudo-wire on SVI is enabled.

Usage notes:

- Prime Fulfillment supports a hybrid configuration for EVC service requests. In a hybrid configuration, the forwarding commands (such as xconnect) for one side of an attachment circuit can be configured under a service instance, and the xconnect configuration for the other side of the attachment circuit can be configured under a switch virtual interface (SVI).
- For examples of these cases, see configlet examples EVC (Pseudowire Core Connectivity, Bridge Domain, Pseudowire on SVI), page 3-212 and EVC (Pseudowire Core Connectivity, no Bridge Domain, no Pseudowire on SVI), page 3-213.
- N-PE Pseudo-wire on SVI is applicable for all connectivity types, but a hybrid SVI configuration is possible only for pseudowire connectivity.
- When MPLS Core Connectivity Type is set as LOCAL connectivity type, the N-PE Pseudo-wire on SVI attribute is always disabled in the policy and service request.
- The N-PE Pseudo-wire on SVI attribute is not supported for IOS XR devices. All the xconnect commands are configured on L2 subinterfaces/service instance.
- Table 3-5 shows various use cases for hybrid configuration for EVC service requests.

Table 3-5 Use Cases for Hybrid Configuration for EVC Service Requests

Use Bridge Domain	EVC	N-PE Pseudowire on SVI	CLIs Generated
True	True	True	• xconnect under VLAN interface. • Service instance under main interface.
True	True	False	• xconnect under service instance. • Service instance under main interface.
False	True	N/A	• xconnect under service instance. • Service instance under main interface.
True	False	True	xconnect under VLAN interface.

Table 3-5 Use Cases for Hybrid Configuration for EVC Service Requests

Use Bridge Domain	EVC	N-PE Pseudowire on SVI	CLIs Generated
True	False	False	xconnect under subinterface.
False	False	False	xconnect under subinterface.

Step 14 Check the **Use Existing ACL Name** check box if you want to assign your own named access list to the port.

By default, this check box is not checked and Prime Fulfillment automatically assigns a MAC-based ACL on the customer facing UNI port, based on values you enter in **UNI MAC addresses** (below).

Step 15 Enter a **Port-Based ACL Name** (if you checked the **Use Existing ACL Name** check box, as mentioned in the previous step).

Note Prime Fulfillment does not create this ACL automatically. The ACL must already exist on the device, or be added as part of a template, before the service request is deployed. Otherwise, deployment will fail.

Step 16 Enter one or more Ethernet MAC addresses in **UNI MAC addresses**.

This selection is present only if you uncheck the **Use Existing ACL Name** check box. Click the **Edit** button to bring up a pop-up window in which you enter MAC addresses to be allowed or denied on the port. You can also specify a range of addresses by setting a base MAC address and a filtered MAC address.

Step 17 Check the **UNI Port Security** check box if you to want to provision port security-related CLIs to the UNI port by controlling the MAC addresses that are allowed to go through the interface.

a. For **Maximum Number of MAC address**, enter the number of MAC addresses allowed for port security.

b. For **Aging,** enter the length of time the MAC address can stay on the port security table.

c. For **Violation Action**, choose what action will occur when a port security violation is detected:

- **PROTECT**—Drops packets with unknown source addresses until a sufficient number of secure MAC addresses are removed to drop below the maximum value.
- **RESTRICT**—Drops packets with unknown source addresses until a sufficient number of secure MAC addresses are removed to drop below the maximum value and causes the Security Violation counter to increment.
- **SHUTDOWN**—Puts the interface into the error-disabled state immediately and sends an SNMP trap notification.

d. In the **Secure MAC Addresses** field, enter one or more Ethernet MAC addresses.

Step 18 Check the **Enable Storm Control** check box to help prevent the UNI port from being disrupted by a broadcast, multicast, or unicast storm.

Enter a threshold value for each type of traffic. The value, which can be specified to two significant digits, represents the percentage of the total available bandwidth of the port. If the threshold of a traffic type is reached, further traffic of that type is suppressed until the incoming traffic falls below the threshold level.

Step 19 Check the **Protocol Tunnelling** check box if you want to define the Layer 2 Bridge Protocol Data Unit (BPDU) frames that can be tunneled over the core to the other end.

For each protocol that you choose, enter the shutdown threshold and drop threshold for that protocol:

a. **Enable cdp**—Enable Layer 2 tunnelling on Cisco Discover Protocol (CDP).

b. **cdp shutdown threshold—**Enter the number of packets per second to be received before the interface is shut down.

c. **cdp drop threshold**—Enter the number of packets per second to be received at which point the interface will start dropping CDP packets.

d. **Enable vtp**—Enable Layer 2 tunnelling on VLAN Trunk Protocol (VTP).

e. **vtp shutdown threshold**—Enter the number of packets per second to be received before the interface is shut down.

f. **vtp drop threshold**—Enter the number of packets per second to be received at which point the interface will start dropping VTP packets.

g. **Enable stp—**Enable Layer 2 tunnelling on Spanning Tree Protocol (STP).

h. **stp shutdown threshold—**Enter the number of packets per second to be received before the interface is shut down.

i. **stp drop threshold**—Enter the number of packets per second to be received at which point the interface will start dropping STP packets.

j. **Recovery Interval**—Enter the amount of time, in seconds, to wait before recovering a UNI port.

Step 20 Enter the **MTU Size** in bytes.

The maximum transmission unit (MTU) size is configurable and optional. The default size is 9216, and the range is 1500 to 9216. Prime Fulfillment does not perform an integrity check for this customized value. If a service request goes to the Failed Deploy state because this size is not accepted, you must adjust the size until the Service Request is deployed.

In Cisco Prime Fulfillment 1.0, different platforms support different ranges.

- For the 3750 and 3550 platforms, the MTU range is 1500 to 1546.
- For the Cisco 7600 Ethernet port, the MTU size is always 9216. Even with the same platform and same IOS release, different line cards support the MTU differently. For example, older line cards only take an MTU size of 9216 and newer cards support 1500 to 9216. However, Cisco Prime Fulfillment 1.0 uses 9216 in both cases.
- For the Cisco 7600 SVI (interface VLAN), the MTU size is 1500 to 9216.

Step 21 If you want to enable template association for this policy, click the **Next** button.

See the section "Enabling Template Association" section on page 3-33 for information about this feature.

Note An additional window appears the policy workflow before the Template Association window. This window allows you to create user-defined attributes within the policy (and service requests based on the policy). For background information on how to use the additional information feature, see Appendix H, "Adding Additional Information to Services." If you are not using this feature, click **Next** to proceed to the Template Association window, or else click **Finish** to save the policy.

Step 22 To save the EVC policy, click **Finish**.

To create a service request based on an EVC ATM-Ethernet Interworking policy, see Managing an EVC ATM-Ethernet Interworking Service Request, page 3-69.

Enabling Template Association

The Prime Fulfillment template feature gives you a means to download free-format CLIs to a device. If you enable templates, you can create templates and data files to download commands that are not currently supported by Prime Fulfillment.

Step 1 To enable template association for the policy, click the **Next** button in the Interface Attribute window (before clicking **Finish**).

The Template Association window appears. In this window, you can enable template support and, optionally, associate templates and data files with the policy. For instructions about associating templates with policies and how to use the features in this window, see Using Templates with Service Requests, page 9-24.

Step 2 When you have completed setting up templates and data files for the policy, click **Finish** in the Template Association window to close it and return to the Policy Editor window.

Step 3 To save the EVC ATM-Ethernet Interworking policy, click **Finish**.

To create a service request based on an EVCATM-Ethernet Interworking policy, see Managing an EVC ATM-Ethernet Interworking Service Request, page 3-69.

Managing an EVC ATM-Ethernet Interworking Service Request

This section provides information on how to provision an EVC ATM-Ethernet Interworking service request. It contains the following subsections:

Overview

An EVC ATM-Ethernet Interworking service request allows you to configure interfaces on an N-PE to support the EVC features described in Creating an EVC ATM-Ethernet Interworking Policy, page 3-54. To create an EVC ATM-Ethernet Interworking service request, an EVC ATM-Ethernet Interworking service policy must already be defined, as described in Creating an EVC ATM-Ethernet Interworking Policy, page 3-54. Based on the predefined EVC policy, an operator creates an EVC service request, with or without modifications to the policy, and deploys the service. One or more templates can also be associated to the N-PE as part of the service request.

ATM-Ethernet interworking is supported through the following configurations:

- ATM Transport Mode (VC)
  - ATM-Ethernet Pseudowire
  - ATM-ATM Local connect
  - ATM-Ethernet Local connect
- ATM Transport Mode (VP)
  - ATM-ATM Local connect

The following steps are involved in creating an EVC ATM-Ethernet Interworking service request:

- Choose an existing EVC ATM-Ethernet Interworking policy.
- Choose a VPN.

Note When working with VPN objects in the context of EVC policies and service requests, only the VPN name and customer attributes are relevant. Other VPN attributes related to MPLS and VPLS are ignored.

- Specify a bridge domain configuration (if applicable).
- Specify a service request description.
- Specify automatic or manual allocation of the VC ID or VPLS VPN ID.
- Add direct connect links (if applicable).
- Add links with L2 access nodes (if applicable).
- Choose the N-PE and UNI interface for links.
- For links with L2 access nodes, choose a Named Physical Circuit (NPC) if more than one NPC exists from the N-PE or the UNI interface.
- Edit the link attributes.
- Modify the service request.
- Save the service request.

For sample configlets for EVC ATM-Ethernet Interworking scenarios, see Sample Configlets, page 3-176.

Creating an EVC ATM-Ethernet Interworking Service Request

To create an EVC ATM-Ethernet Interworking service request, perform the following steps.

Step 1 Choose **Operate > Service Requests > Service Request Manager**.

The Service Request Manager window appears.

Step 2 Click **Create**.

The Service Request Editor window appears.

Step 3 From the Policy drop-down list, choose an EVC ATM-Ethernet Interworking policy from the policies previously created (see Creating an EVC ATM-Ethernet Interworking Policy, page 3-54).

The EVC Service Request Editor window appears. The new service request inherits all the properties of the chosen EVC ATM-Ethernet Interworking policy, such as all the editable and non-editable features and pre-set parameters.

Step 4 Continue with the steps contained in the next section, Setting the Service Request Details, page 3-35.

Setting the Service Request Details

After you have selected the EVC policy to be used as the basis of the service request, the EVC Service Request Editor window appears. It is divided into three main sections:

- Link Page
- Direct Connect Links (no NPCs)
- Links with L2 Access Nodes (involves NPCs)

This window enables you to specify options for the service request, as well as configure directly connected links and links with L2 access nodes. The options displayed in first section of the window change, depending on the MPLS Core Connectivity Type that was specified in the policy (pseudowire or local). For clarity, each of these scenarios is presented in a separate section below, to highlight the different window configurations and behavior of the displayed options.

Proceed to the appropriate section, as determined by the MPLS Core Connectivity Type for the policy:

- Pseudowire Core Connectivity, page 3-35
- Local Core Connectivity, page 3-39

Instructions for setting up direct connect links and links with L2 access nodes are presented in later sections.

Pseudowire Core Connectivity

This section covers the case in which the MPLS Core Connectivity Type for the EVC ATM-Ethernet Interworking policy is PSEUDOWIRE.

To set the attributes in the first section of the EVC Service Request Editor window, perform the following steps.

Note The **Job ID** and **SR ID** fields are read-only. When the service request is being created for the first time, the fields display a value of NEW. When an existing service request is being modified, the values of the fields indicate the respective IDs that the Prime Fulfillment database holds within the editing flow of the service request.

Note The **Policy** field is read-only. It displays the name of the policy on which the service request is based. Clicking on the read-only policy name displays a list of all the attribute values set within the policy.

Step 1 Click **Select VPN** to choose a VPN for use with this service request.

The Select VPN window appears with the VPNs defined in the system.

Note The same VPN can be used by service requests with LOCAL and PSEUDOWIRE core types. If a VPN for a service request is used with VPLS core type, the same VPN cannot be used for service requests with LOCAL or PSEUDOWIRE core type.

Step 2 Choose a **VPN Name** in the Select column.

Step 3 Click **Select**.

The EVC Service Request Editor window appears with the VPN name displayed.

Step 4 Check the **AutoPick VC ID** check box if you want Prime Fulfillment to choose a VC ID.

If you do not check this check box, you will be prompted to provide the ID in the VC ID field, as covered in the next step.

When AutoPick VC ID is checked, Prime Fulfillment allocates a VC ID for pseudowires from the Prime Fulfillment-managed VC ID resource pool. In this case, the text field for the VC ID option is non-editable.

Step 5 If AutoPick VC ID was unchecked, enter a VC ID in the **VC ID** field.

Usage notes:

- The AutoPick VC ID attribute appears during the creation of an EVC pseudowire service request.
- The VC ID value must be an integer value corresponding to a VC ID.
- When a VC ID is manually allocated, Prime Fulfillment verifies the VC ID to see if it lies within Prime Fulfillment's VC ID pool. If the VC ID is in the pool but not allocated, the VC ID is allocated to the service request. If the VC ID is in the pool and is already in use, Prime Fulfillment prompts you to allocate a different VC ID. If the VC ID lies outside of the Prime Fulfillment VC ID pool, Prime Fulfillment does not perform any verification about whether or not the VC ID allocated. The operator must ensure the VC ID is available.
- The VC ID can be entered only while creating a service. If you are editing the service request, the VC ID field is not editable.

Step 6 Check the **Configure Bridge Domain** check box to determine bridge domain characteristics.

The behavior of the Configure Bridge Domain option works in tandem with the choice you selected in the MPLS Core Connectivity Type option in the EVC policy, which in this case is pseudowire core connectivity. There are two cases:

- With EVC:
  - If **Configure With Bridge Domain** is checked, the policy will configure pseudowires under SVIs associated to the bridge domain.
  - If **Configure With Bridge Domain** is unchecked, the policy will configure pseudowires directly under the service instance. This will conserve the global VLAN.
- Without EVC:
  - If **Configure With Bridge Domain** is checked, the policy will configure pseudowires under SVIs.
  - If **Configure With Bridge Domain** is unchecked, the policy will configure pseudowires directly under subinterfaces.

  Pseudowires can be configured either directly under service instance of the corresponding EVC-capable interface or under SVIs associated to the bridge domain.

Step 7 Check the **Use Split Horizon** check box to enable split horizon with bridge domain.

Usage notes:

- The Use Split Horizon attribute is disabled by default.
- The Use Split Horizon attribute can be used only when the Configure Bridge Domain check box is checked (enabled).

- When Use Split Horizon is enabled, the **bridge domain** command in the CLI will be generated with split horizon. When it is disabled, the **bridge domain** command will be generated without split horizon.

Step 8 Click the "Click here" link of the **Description** attribute to enter a description label for the service request.

This is useful for searching the Prime Fulfillment database for the particular service request.

A dialogue appears in which you can enter a description.

Step 9 To set up direct connect links, see the section Setting Direct Connect Links, page 3-41.

Step 10 To set up links with L2 access nodes, see the section Setting Links with L2 Access Nodes, page 3-52.

Local Core Connectivity

This section covers the case in which the MPLS Core Connectivity Type for the EVC ATM-Ethernet Interworking policy is LOCAL.

To set the attributes in the first section of the EVC Service Request Editor window, perform the following steps.

Step 1 The **Job ID** and **SR ID** fields are read-only.

When the service request is being created for the first time, the fields display a value of NEW. When an existing service request is being modified, the values of the fields indicate the respective IDs that the Prime Fulfillment database holds within the editing flow of the service request.

Step 2 The **Policy** field is read-only.

It displays the name of the policy on which the service request is based.

Step 3 Click **Select VPN** to choose a VPN for use with this service request.

The Select VPN window appears with the VPNs defined in the system.

Note The same VPN can be used by service requests with LOCAL and PSEUDOWIRE core types. If a VPN for a service request is used with VPLS core type, the same VPN cannot be used for service requests with LOCAL or PSEUDOWIRE core type.

Step 4 Choose a **VPN Name** in the Select column.

Step 5 Click **Select**.

The EVC Service Request Editor window appears with the VPN name displayed.

Step 6 Check the **Configure Bridge Domain** check box to determine bridge domain characteristics.

Usage notes:

- If Configure Bridge Domain is checked, all links will have the same bridge domain ID allocated from the VLAN pool on the N-PE. All non-EVC links will have the Service Provider VLAN as the bridge domain ID. On the other hand, if no EVC links are added, the Service Provider VLAN will be allocated first and this will be used as the bridge domain ID when EVC links are added.
- If Configure Bridge Domain is unchecked, a maximum of two links that terminate on the same N-PE can be added. (This uses the **connect** command available in the EVC infrastructure.) This is only supported for ATM-ATM local connect.

Note See the following comments for details on how Prime Fulfillment autogenerates the connect name.

Because the device only accepts a maximum of 15 characters for the connect name, the connect name is generated using the following format:

CustomerNameTruncatedToMaxPossibleCharacters_ServiceRequestJobID

For example, if the customer name is NorthAmericanCustomer and the service request job ID is 56345, the autogenerated connect name would be NorthAmer_56345.

The CLI generated would be:

```
connect NorthAmer_56345 ATM7/0/5 11 ATM7/0/4 18
```

In this case, 11 and 18 are service instance VPIs.

- If the policy setting for Configure Bridge Domain is non-editable, the option in the service request will be read-only.

Step 7 Check the **Use Split Horizon** check box to enable split horizon with bridge domain.

Usage notes:

- The Use Split Horizon attribute is disabled by default.
- The Use Split Horizon attribute can be used only when the Configure Bridge Domain check box is checked (enabled).
- When Use Split Horizon is enabled, the **bridge domain** command in the CLI will be generated with split horizon. When it is disabled, the **bridge domain** command will be generated without split horizon.

Step 8 Click the "Click here" link of the **Description** attribute to enter a description label for the service request.

A dialogue appears in which you can enter a description.

Step 9 To set up direct connect links, see the section Setting Direct Connect Links, page 3-41.

Step 10 To set up links with L2 access nodes, see the section Setting Links with L2 Access Nodes, page 3-52.

Setting up Links to the N-PE

The lower two sections of the EVC Service Request Editor window allow you to set up links to the N-PE. For direct connect links, the CE is directly connected to the N-PE, with no intermediate L2 access nodes. For links with L2 access nodes, there are intermediate devices present between the CE and NPE requiring NPCs to be created in Prime Fulfillment.

The Direct Connect Links section of the window is where you set up links that directly connect to the N-PE. No NPCs are involved. ATM links are supported for direct connect links.

The Links with L2 Access Nodes section is where you set up links with L2 (Ethernet) access nodes. NPCs are involved.

Note ATM interfaces cannot be in L2 access nodes.

See the appropriate section, depending on which type of link you are setting up:

- Setting Direct Connect Links, page 3-41
- Setting Links with L2 Access Nodes, page 3-52

Note Many of steps for setting up the two link types are the same. The basic workflow for setting up links, as well as the attributes to be set, are presented in the following section Setting Direct Connect Links, page 3-41. Even if you are setting up links with L2 access nodes, it will be helpful to refer to the information presented in that section, as the section on L2 access nodes only covers the unique steps for such links.

Setting Direct Connect Links

To set up the direct connect links, perform the following steps. Most of these steps apply to links with L2 access nodes also.

Step 1 Click **Add** to add a link.

A new numbered row for the link attributes appears.

Step 2 Click **Select N-PE** in N-PE column.

The Select PE Device window appears. This window displays the list of currently defined PEs.

a. The **Show PEs with** drop-down list shows PEs by Provider, PE Region Name, or by Device Name.

b. The **Find** button allows a search for a specific PE or a refresh of the window.

c. The **Rows per page** drop-down list allows the user to configure the number of entries displayed on the screen at one time.

Step 3 In the **Select** column, choose the PE device name for the link.

Step 4 Click **Select**.

The EVC Service Request Editor window reappears displaying the name of the selected PE in the NPE column.

Step 5 Choose the UNI interface from the drop-down list in the UNI column.

Note Prime Fulfillment only displays the available interfaces for the service, based on the configuration of the underlying interfaces, existing service requests that might be using the interface, and the customer associated with the service request. You can click the **Details** button to display a pop-up window with information on the available interfaces, such as interface name, customer name, VPN name, job ID, service request ID, service request type, translation type, and VLAN ID information.

Note When the UNI is configured on an N-PE device running IOS XR, the Standard UNI Port attribute is not supported. All the CLIs related to Standard UNI Port and UNI Port Security are ignored in this case.

Step 6 Check the **EVC check box** to mark the link for configuring service instance for the links.

Note The EVC check box is mentioned at this stage because the setting of the check box alters the behavior of the link editing function available in the Link Attributes column. This is covered in the next steps.

Note The EVC check box is unchecked by default. The default value for the check box can be changed by setting the value of the DCPL property Pr ovisioning\ProvDrv\CheckFlexUniCheckBox.

Editing the Link Attributes

The next steps document the use of the **Edit** link in the Link Attributes column. (In the case where the link attributes have already been set, this link changes from **Edit** to **Change**.) The link editing workflow changes depending on the status of the EVC check box for the link. If the EVC check box is checked, the editing workflow involves setting attributes in two windows, for two sets of link attributes:

- The EVC Details
- Standard UNI Details

If the EVC check box for the link is not checked, only the Standard UNI Details window is presented.

In the steps that follow, both scenarios covered.

Note If you are setting up and ATM link (by choosing an ATM interface as the UNI on the N-PE device, there is a different workflow. The check box in the EVC column dynamically disappears, and clicking the Edit link in the link attributes column brings up the ATM-Ethernet Attributes window. For information on using this window to set up an ATM link, see Setting the ATM Link Attributes, page 3-85.

Step 7 Click **Edit** in the Link Attributes column to specify the UNI attributes.

EVC Details Window

If the EVC check box is checked, the EVC Details window appears

All of the fields in the EVC Details window are enabled based on the policy settings. For example, if Both Tags is selected in the policy and is editable, then the Match Inner and Outer Tags check box will be selected and editable in this window. The behavior is similar for the other attributes in the EVC Details window

Step 8 Check the **AutoPick Service Instance ID** check box to specify that the service instance ID will be autogenerated and allocated to the link during service request creation.

If the check box is unchecked, you must specify the service instance ID (see the next step).

Usage notes:

- The service instance ID represents an Ethernet Flow Point (EFP) on an interface in the EVC infrastructure. The service instance ID is locally significant to the interface. This ID has to be unique only at the interface level. The ID must be a value from 1 to 8000.
- There are no resource pools available in Prime Fulfillment from which to allocate the service instance IDs.
- In the case of a manually provided service instance ID, it is the responsibility of the operator to maintain the uniqueness of the ID at the interface level.
- This attribute is not displayed for IOS XR devices.

Step 9 If the AutoPick Service Instance ID check box is not checked, enter an appropriate value for the service instance ID in the **Service Instance ID** field.

Step 10 Check the **AutoPick Service Instance Name** check box to specify that the service instance name will be autogenerated.

If the check box is unchecked, you can specify the service instance name (see the next step).

Usage notes:

- If the check box is checked, the Service Instance Name text field is disabled.
- The service instance name is autogenerated in the following pattern: *CustomerName_ServiceRequestJobID*.
- For example configlets, see EVC (No AutoPick Service Instance Name, No Service Instance Name), page 3-215, EVC (User-Provided Service Instance Name, Pseudowire Core Connectivity), page 3-216, and EVC (User-Provided Service Instance Name, Local Core Connectivity), page 3-217.
- This attribute is not displayed for IOS XR devices.

Step 11 If the AutoPick Service Instance Name check box is not checked, enter an appropriate value for the service instance ID in the **Service Instance Name** field.

Usage notes:

- The text string representing the service instance name must be 40 characters or less and contain no spaces. Other special characters are allowed.
- If AutoPick Service Instance Name is unchecked and no service instance name is entered in the text field, then Prime Fulfillment does not generate the global **ethernet evc evcname** command in the device configuration generated by the service request.

Step 12 Check the **AutoPick Bridge Domain/VLAN ID** check box to have Prime Fulfillment autopick the VLAN ID for the service request during service request creation.

If this check box is unchecked, the you must specify a bridge domain VLAN ID (see the next step).

Usage notes:

- AutoPick Bridge Domain/VLAN ID consumes a global VLAN ID on the device.
- The bridge domain VLAN ID is picked from the existing Prime Fulfillment VLAN pool.
- This attribute is not displayed for IOS XR devices.

Step 13 If the AutoPick Bridge Domain/VLAN ID check box is unchecked, enter an appropriate value in the **Bridge Domain/VLAN ID** field.

Note This configuration applies in conjunction with the Configure Bridge Domain option in the EVC Service Request Editor window. If the option is not enabled in that window, then AutoPick Bridge Domain/VLAN ID check box is redundant and not required.

When a VLAN ID is manually allocated, Prime Fulfillment verifies the VLAN ID to see if it lies within Prime Fulfillment's VLAN ID pool. If the VLAN ID is in the pool but not allocated, the VLAN ID is allocated to the service request. If the VLAN ID is in the pool and is already in use, Prime Fulfillment prompts you to allocate a different VLAN ID. If the VLAN ID lies outside of the Prime Fulfillment VLAN ID pool, Prime Fulfillment does not perform any verification about whether the VLAN ID allocated. The operator must ensure the VLAN ID is available.

Step 14 Check the **AutoPick Bridge Domain/VLAN ID Secondary N-PE** check box to have Prime Fulfillment autopick the bridge domain VLAN ID for the secondary N-PE of a dual-homed ring during service request creation.

If this check box is unchecked, the you must specify a secondary bridge domain VLAN ID for the secondary N-PE (see the next step).

Usage notes:

- This attribute is only applicable in the case of a dual-homed ring (a ring that terminates on two different N-PEs). Prime Fulfillment supports having a separate bridge domain VLAN ID for the secondary N-PE.
- In a dual-homed ring, if the two N-PEs are in different access domains, Prime Fulfillment allocates the bridge domain VLAN IDs from both primary and secondary N-PE access domains. When both are in the same Access Domain, Prime Fulfillment allocates a common VLAN ID from the Access Domain to which these belong.
- AutoPick Bridge Domain/VLAN ID Secondary N-PE consumes a global VLAN ID on the device.
- The bridge domain VLAN ID is picked from the existing Prime Fulfillment VLAN pool.
- This attribute is not displayed for IOS XR devices.

Step 15 If the AutoPick Bridge Domain/VLAN ID Secondary N-PE check box is unchecked, enter an appropriate value in the **Bridge Domain/VLAN ID Secondary N-PE** field.

Step 16 Check the **Match Inner and Outer Tags** check box to enable service requests created with the policy to match both the inner and outer VLAN tags of the incoming frames.

If you do not check this check box, service requests created with the policy will match only the outer VLAN tag of the incoming frames.

Checking the Match Inner and Outer Tags attribute causes the Inner VLAN ID and Outer VLAN ID fields (covered in the next steps) to appear.

Step 17 If the Match Inner and Outer Tags check box is checked, enter the inner and outer VLAN tags in the **Inner VLAN ID** and **Outer VLAN ID** fields.

Usage notes:

- You can specify single values, single ranges, multiples values, multiple ranges, or combinations of these. Examples:
  - 10
  - 10, 15,17
  - 10-15
  - 10-15,17-20
  - 10,20-25
- If the Inner VLAN Ranges attribute is set to true in the policy, the Inner VLAN ID field can take a range of inner VLAN tags.
- If the Outer VLAN Ranges attribute is set to true in the policy, the Outer VLAN ID field can take a range of Outer VLAN tags.

Step 18 If the Match Inner and Outer Tags check box is unchecked, enter the outer VLAN tag in the **Outer VLAN ID** field.

Note The VLAN specified in Outer VLAN ID will be provisioned on the rest of the L2 access nodes (if the link has any), including the customer-facing UNI.

Note You may also have Prime Fulfillment autopick the outer VLAN ID as covered in the next step.

Step 19 Check the **AutoPick Outer VLAN** check box to have Prime Fulfillment autopick the outer VLAN ID from a previously created outer VLAN ID resource pool.

If this check box is unchecked, the operator will be prompted to specify an outer VLAN ID.

Note Use of the AutoPick Outer VLAN attribute requires that two elements have already been set up in Prime Fulfillment. One is an Interface Access Domain, which is a logical element that groups the physical ports of an N-PE device. The other is an EVC Outer VLAN resource pool, which is used by the Interface Access Domain. For instructions on how to set up these elements, see the sections Setting Up Resources, page 2-40, and Resource Pools, page 2-44.

Usage notes:

- AutoPick Outer VLAN can be used for interfaces that support EVC functionality
- AutoPick Outer VLAN consumes a VLAN ID on the interface that supports EVC.
- The bridge domain VLAN ID is picked from the existing Prime Fulfillment VLAN pool.

Step 20 In the VLAN Rewrite section of the window, choose a **Rewrite Type** from the drop-down list.

The choices are:

- **Pop**
- **Push**
- **Translate**

The subsequent attributes in the GUI change depending on the choice of Rewrite Type, as covered in the next steps.

Step 21 If Pop is the Rewrite Type, two check boxes appear:

a. Check the **Pop Outer Tag** check box to pop the outer VLAN ID tag of the incoming frames that fulfill the match criteria. If this check box is unchecked, the outer tag of the incoming traffic will not be popped.

b. Check the **Pop Inner Tag** check box to pop the inner VLAN ID tag of the incoming frames that fulfill the match-criteria. If this check box is unchecked, the inner tag will not be changed.

Note that if Pop Inner Tag is checked, Pop Outer Tag is automatically checked.

Step 22 If Push is the Rewrite Type, two text boxes appear:

a. In the text box **Outer VLAN ID**, enter an outer VLAN ID tag that will be imposed on the incoming frames that fulfill the match criteria. All service requests created with this setting push a dot1q outer tag on the incoming frames matching the match criteria. If a value is not provided, the push operation is ignored and not configured on the device.

b. In the text box **Inner VLAN ID**, enter an inner VLAN ID tag that will be imposed on the incoming frames that fulfill the match criteria. All service requests created with this setting push a dot1q inner tag on the incoming frames matching the match criteria. The Inner VLAN tag cannot be pushed without an Outer VLAN tag. That is, when pushing an Inner VLAN tag, the Outer VLAN tag also must be defined.

Step 23 If Translate is the Rewrite Type, a **Translation Type** drop-down list appears.

The choices available in this list vary depending on the setting of the Match Inner and Outer Tags attribute (set in a previous step).

a. If the Match Inner and Outer Tags check box is checked (true), choose a translation type of **1:1**, **1:2**, **2:1**, or **2:2** from the Translation Type drop-down list.

– If you choose 1:1 or 2:1, enter a value in the **Outer VLAN ID** text box that appears. The outer tag of all the incoming frames that fulfill the match criteria will be translated to this ID.

- If you choose 1:2 or 2:2, enter values in the **Outer VLAN ID** and **Inner VLAN ID** text boxes that appear. The outer and inner tags of all the incoming frames that fulfill the match criteria will be translated to these IDs.

b. If the Match Inner and Outer Tags check box is unchecked (false), choose a translation type of **1:1** or **1:2** from the Translation Type drop-down list.

- If you choose 1:1, enter a value in the **Outer VLAN ID** text box that appears. The outer tag of all the incoming frames that fulfill the match criteria will be translated to this ID.
- If you choose 1:2, enter values in the **Outer VLAN ID** and **Inner VLAN ID** text boxes that appear. The outer and inner tags of all the incoming frames that fulfill the match criteria will be translated to these IDs.

Step 24 Clicked **Next** to save the settings in the EVC Details window.

The Standard UNI Details window appears.

Step 25 Continue with setting the standard UNI link attributes in the next steps.

Editing the Standard UNI Attributes

The following steps cover setting the attributes in the Standard UNI Details window. In the case of a link which is not set as an EVC link (by not checking the EVC check box in the Service Request Details window), editing the link attributes begins with this window.

Note The attributes that appear in the Standard UNI Details window are dynamically configured by Prime Fulfillment. Some of the attributes covered in the steps below might not appear in the window, depending on the policy and service request settings or the link type. For example, if the MPLS core connectivity type of the EVC policy is local, the pseudowire-related attributes will not appear. Also, setting the link as EVC or non-EVC will change the attributes that appear in the window. In addition, attributes are filtered based on device type (IOS or IOS XR). These cases are noted in the steps, for reference.

Step 26 The **N-PE/U-PE Information** and **Interface Name** fields display the PE device and interface name selected in previous steps.

These fields are read-only.

Step 27 Choose an **Encapsulation** type from the drop-down list.

The choices are:

- **DOT1QTRUNK**—Configures the UNI as a trunk with 802.1q encapsulation. If the UNI belongs to a directly connected and EVC link, this setting signifies that the incoming frames are 802.1q encapsulated and that they match the VLAN ID configured for the link. This specific topology does not involve a trunk UNI as such.
- **DOT1QTUNNEL**—Configures the UNI as an 802.1q tunnel (also known as a dot1q tunnel or Q-in-Q) port.
- **ACCESS**—Configures the UNI as an access port.

This attribute allows you to deploy different types of UNI encapsulation on different links of a service.

Usage notes:

- In the case of direct connect links for which EVC is enabled (by checking the EVC check box in the EVC Service Request Editor window), the choices for the Encapsulation type are DOT1Q and DEFAULT.

Step 28 In the **PE/UNI Interface Description** field, enter a description for the interface, if desired.

Step 29 Check the **UNI Shutdown** check box if you want to leave the UNI port shut during service activation (for example, when the service provider wants to deploy a service in the network but wants to activate it at a later time).

Step 30 Specify the type of **VLAN Translation** for the service request by clicking the appropriate radio button.

The choices are:

- **No**—No VLAN translation is performed. (This is the default.)
- **1:1**—1:1 VLAN translation.
- **2:1**—2:1 VLAN translation.
- **1:2**—1:2 VLAN translation.
- **2:2**—2:2 VLAN translation.

Usage notes:

- The VLAN Translation attribute does not appear for direct connect links if the EVC check box is enabled. It does appear for the following combinations:
  - Direct connect links with EVC check box disabled.
  - L2 access nodes with EVC check box enabled or disabled.
- Choosing a selection other than No causes other fields to appear in the GUI, which you can set based on your configuration:
  - **CE VLAN**—Provide a value between 1 and 4096.
  - **Auto Pick**—Check this check box to have Prime Fulfillment autopick the outer VLAN from the VLAN resource pool.
  - **Outer VLAN**—If Auto Pick is unchecked, provide a value between 1 and 4096.
  - **Select where 2:1 or 2:2 translation takes place**—Specify the device where the 2;1 or 2:2 VLAN translation will take place. If you choose Auto, the VLAN translation takes place at the device closest to the UNI port.
- VLAN translation, and all standard UNI and port security attributes are applicable for links with L2 access. If the UNI is on an N-PE, these attributes will not appear.
- When the VLAN translation takes place on a U-PE or PE-AGG device, the VLAN translation command is configured on the NNI interface of the selected device. When the VLAN translation takes place on an NP-E, the VLAN translation command is configured on the UNI interface of the device.
- When there are two NNI interfaces in a ring-based environment, VLAN translation is applied for both of these NNI interfaces.
- 1:1 and 2:1 VLAN translations are supported with the same syntax as for non-EVC (switchport-based N-PE syntax) terminating attachment circuits.

Step 31 Check the **N-PE Pseudo-wire on SVI** check box to have Prime Fulfillment generate forwarding commands under SVIs (switch virtual interfaces).

By default, this check box is not checked. In this case, Prime Fulfillment generates forwarding commands under the service instance.

For an EVC link, the attribute N-PE Pseudo-wire on SVI is dependent on the value of the attribute Configure with Bridge Domain (this is available in the service request workflow in the EVC Service Request Editor window). N-PE Pseudo-wire on SVI, if enabled, will be reflected only when Configure with Bridge Domain is set to true. Otherwise, the service request will not be created with xconnect under SVI, even if N-PE Pseudo-wire on SVI is enabled.

Usage notes:

- For an EVC link, the attribute N-PE Pseudo-wire on SVI is dependent on the value of the attribute Configure with Bridge Domain (in the EVC Service Request Editor window). N-PE Pseudo-wire on SVI, if enabled, will be reflected only when Configure with Bridge Domain is set to true. Otherwise, the service request will not be created with scanned under SVI, even if N-PE pseudo-wire on SVI is enabled.
- Prime Fulfillment supports a hybrid configuration for EVC service requests. In a hybrid configuration, the forwarding commands (such as scanned) for one side of an attachment circuit can be configured under a service instance, and the xconnect configuration for the other side of the attachment circuit can be configured under a switch virtual interface (SVI).
- N-PE Pseudo-wire on SVI is applicable for all connectivity types (PSEUDOWIRE or LOCAL), but a hybrid SVI configuration is possible only for pseudowire connectivity.
- When MPLS Core Connectivity Type is set as LOCAL connectivity type, the N-PE Pseudo-wire on SVI attribute is always disabled in the policy and service request.
- For examples of these cases, see configlet examples EVC (Pseudowire Core Connectivity, Bridge Domain, Pseudowire on SVI), page 3-212 and EVC (Pseudowire Core Connectivity, no Bridge Domain, no Pseudowire on SVI), page 3-213.
- For additional information on the N-PE Pseudo-wire on SVI attribute, see the corresponding coverage in the EVC policy section in the section Setting the Interface Attributes, page 3-28.
- The N-PE Pseudo-wire on SVI attribute is not supported for IOS XR devices. All the xconnect commands are configured on L2 subinterfaces/service instance.

Step 32 Check the **Use Existing PW Class** check box to enable the selection of a pseudowire class.

This attribute is unchecked by default.

Usage notes:

- If Use Existing PW Class is checked, an additional attribute, **Existing PW Class Name**, appears in the GUI. Enter the name of a pseudowire class which already exists in the device.
- If Use Existing PW Class is checked, the PW Tunnel Selection and Interface Tunnel attributes will disappear from the window. This is to prevent Prime Fulfillment from generating the pseudowire class.
- The Use PseudoWireClass attribute is only available if the MPLS core connectivity type was set as PSEUDOWIRE in the Service Options window (see Pseudowire Core Connectivity, page 3-35).
- Use PseudoWireClass is only applicable for IOS XR devices.

Step 33 Check the **PW Tunnel Selection** check box if you want to be able to manually select the Traffic Engineering (TE) tunnel for the pseudowire connecting point-to-point N-PEs.

Usage notes:

- Checking the PW Tunnel Selection check box activates the Interface Tunnel attribute field (see the next step).
- This attribute only appears if the MPLS core connectivity type is set as pseudowire in the EVC policy.

Step 34 If you checked the PW Tunnel Selection check box, enter the TE tunnel ID in the **Interface Tunnel** text field.

Prime Fulfillment uses the tunnel information to create and provision a pseudowire class that describes the pseudowire connection between two N-PEs. This pseudowire class can be shared by more than one pseudowire, as long as the pseudowires share the same tunnel ID and remote loopback address. During service request creation, Prime Fulfillment does not check the validity of the tunnel ID number. That is, Prime Fulfillment does not verify the existence of the tunnel.

Step 35 Check the **Use PseudoWireClass** check box to enable the selection of a pseudowire class.

This attribute is unchecked by default.

Usage notes:

- The pseudowire class name is used for provisioning pw-class commands on IOS XR devices. See Creating and Modifying Pseudowire Classes, page 3-14 for additional information on pseudowire class support for IOS XR devices.
- If Use PseudoWireClass is checked, an additional attribute, **PseudoWireClass**, appears in the GUI. Click the **Select** button of PseudoWireClass attribute to choose a pseudowire class previously created in Prime Fulfillment.
- The Use PseudoWireClass attribute is only available if the MPLS core connectivity type was set as PSEUDOWIRE in the Service Options window (see Setting the Service Options, page 3-20).
- Use PseudoWireClass is only applicable for IOS XR devices.

Step 36 Check the **AutoPick Bridge Group Name** check box to have Prime Fulfillment autopick the bridge group name during service request creation.

If this check box is unchecked, you are prompted to specify a bridge group name during service request creation (see the next step).

Usage notes:

- This attribute only displays for IOS XR devices.
- If the AutoPick Bridge Group Name check box is unchecked, enter an bridge group name in the **Bridge Group Name** text field.

Step 37 Check the **AutoPick Bridge Domain/VLAN ID** check box to have Prime Fulfillment autopick the VLAN ID during service request creation.

If this check box is unchecked, you are prompted to specify a VLAN ID during service request creation (see the next step).

Usage notes:

- AutoPick Bridge Domain/VLAN ID consumes a global VLAN ID on the device.
- The bridge domain VLAN ID is picked from the existing Prime Fulfillment VLAN pool.
- The AutoPick Bridge Domain/VLAN ID attribute appears for both Cisco 7600 and ASR 9000 devices. It will be displayed only for non-EVC links.

Step 38 If the AutoPick Bridge Domain/VLAN ID check box is unchecked, enter an ID number in the **Bridge Domain/VLAN ID** text field.

Usage notes:

- If AutoPick Bridge Domain/VLAN ID is checked, this field is non-editable.
- When a VLAN ID is manually allocated, Prime Fulfillment verifies the VLAN ID to see if it lies within Prime Fulfillment's VLAN ID pool. If the VLAN ID is in the pool but not allocated, the VLAN ID is allocated to the service request. If the VLAN ID is in the pool and is already in use, Prime Fulfillment prompts you to allocate a different VLAN ID. If the VLAN ID lies outside of the Prime Fulfillment VLAN ID pool, Prime Fulfillment does not perform any verification about whether the VLAN ID allocated. The operator must ensure the VLAN ID is available.

- The Bridge Domain/VLAN ID text field appears for both Cisco 7600 and ASR 9000 devices. It will be displayed only for non-EVC links.

Step 39 For **L2VPN Group Name** choose one of the following from the drop-down list:

- **ISC**
- **VPNSC**

Usage notes:

- This attribute is used for provisioning the L2VPN group name on IOS XR devices.

Note The choices in the drop-down list are derived from a configurable DCPL property. For information about how to define the L2VPN Group Name choices available in the drop-down list, see Defining L2VPN Group Names for IOS XR Devices, page 3-18.

- L2VPN Group Name is only applicable for IOS XR devices.

Step 40 Enter an **E-Line Name** to specify the point-to-point (p2p) E-line name.

Usage notes:

- If no value is specified for the **E-Line Name**, Prime Fulfillment autogenerates a default name as follows:
  - For PSEUDOWIRE core connectivity type, the format is:

    DeviceName--VC_ID
  - For LOCAL core connectivity type, the format is:

    DeviceName--VLAN_ID

  If the default name is more than 32 characters, the device names are truncated.
- E-Line Name is only applicable for IOS XR devices.

Step 41 Click **OK** to save the Standard UNI settings and return to the EVC SR window.

The value in the Link Attributes column now displays as "Changed," signifying that the link settings have been updated. You can edit the link attributes now or at a future time by clicking on the Changed link and modifying the settings in the Standard UNI Details window.

See Modifying the EVC Service Request, page 3-53, for details on editing the link attributes.

Step 42 To add another link click the **Add** button and set the attributes for the new link as in the previous steps in this section.

Step 43 To delete a link, check the check box in the first column of the row for that link and click the **Delete** button.

Step 44 If you want to set up links with L2 access nodes for this service request, see Setting Links with L2 Access Nodes, page 3-52.

Step 45 When you have completed setting the attributes in the EVC Service Request Editor window, click the **Save** button at the bottom of the window to save the settings and create the EVC service request.

If any attributes are missing or incorrectly set, Prime Fulfillment displays a warning in the lower left of the window. Make any corrections or updates needed (based on the information provided by Prime Fulfillment), and click the **Save** button.

For information on modifying an EVC service request see the section Modifying the EVC Service Request, page 3-53. For additional information about saving an EVC service request, see Saving the EVC Service Request, page 3-54.

Setting the ATM Link Attributes

This section describes how to set up a direct connect link as an ATM link.

To set up the ATM link, perform the following steps.

Step 1 In the Direct Connect Links section of the EVC Service Request Editor window, specify the device for which you would like to set up an ATM link.

Step 2 Choose an ATM interface for the UNI.

Note ATM interfaces are displayed in the drop-down list in the UNI column only if the EVC service request is based on an ATM-Ethernet Interworking policy type.

When you choose an ATM interface, the check box in the EVC column dynamically disappears from the GUI.

Step 3 In the Link Attributes column, click the **Edit** link of the device on which you want to add an ATM link.

The ATM UNI Details window appears.

All of the fields in the ATM UNI Details window are enabled based on the policy settings.

Step 4 Choose the **Transport Mode** from the drop-down list.

The choices are:

- **VP**—Virtual path mode. This is the default.
- **VC**—Virtual circuit mode.

Step 5 Choose the **ATM Encapsulation** from the drop-down list.

- **AAL5SNAP**

Step 6 To specify the ATM virtual channel descriptor (VCD)/subinterface number, enter a value in the **ATM VCD/Sub-Interface #** field.

The value can be from 1 to 2147483647.

Step 7 To specify the ATM virtual path identifier (VPI), enter a value in the **ATM VPI** field.

The value can be from 0 to 255.

Step 8 To specify the ATM virtual channel identifier (VCI), a value in the **ATM VCI** field.

The value can be from 32 to 65535.

Step 9 Check the **UNI Shutdown** check box if you want to leave the UNI port shut during service activation (for example, when the service provider wants to deploy a service in the network but wants to activate it at a later time).

Step 10 Check the **Use Existing PW Class** check box to enable the selection of a pseudowire class.

This attribute is unchecked by default.

Usage notes:

- If Use Existing PW Class is checked, an additional attribute, **Existing PW Class Name**, appears in the GUI. Enter the name of a pseudowire class which already exists in the device.
- If Use Existing PW Class is checked, the PW Tunnel Selection and Interface Tunnel attributes will disappear from the window. This is to prevent Prime Fulfillment from generating the pseudowire class.
- The Use PseudoWireClass attribute is only available if the MPLS core connectivity type was set as PSEUDOWIRE in the Service Options window (see Pseudowire Core Connectivity, page 3-35).
- Use PseudoWireClass is only applicable for IOS XR devices.

Step 11 Check the **N-PE Pseudo-wire on SVI** check box to have Prime Fulfillment generate forwarding commands under SVIs (switch virtual interfaces).

By default, this check box is not checked. In this case, Prime Fulfillment generates forwarding commands under the service instance.

For an EVC link, the attribute N-PE Pseudo-wire on SVI is dependent on the value of the attribute Configure with Bridge Domain (this is available in the service request workflow in the EVC Service Request Editor window). N-PE Pseudo-wire on SVI, if enabled, will be reflected only when Configure with Bridge Domain is set to true. Otherwise, the service request will not be created with xconnect under SVI, even if N-PE Pseudo-wire on SVI is enabled.

Usage notes:

- For an ATM link, the attribute N-PE Pseudo-wire on SVI is dependent on the value of the attribute Configure with Bridge Domain (in the EVC Service Request Editor window). N-PE Pseudo-wire on SVI, if enabled, will be reflected only when Configure with Bridge Domain is set to true. Otherwise, the service request will not be created with xconnect under SVI, even if N-PE pseudo-wire on SVI is enabled.
- Prime Fulfillment supports a hybrid configuration for EVC service requests. In a hybrid configuration, the forwarding commands (such as xconnect) for one side of an attachment circuit can be configured under a service instance, and the xconnect configuration for the other side of the attachment circuit can be configured under a switch virtual interface (SVI).
- N-PE Pseudo-wire on SVI is applicable for all connectivity types (PSEUDOWIRE or LOCAL), but a hybrid SVI configuration is possible only for pseudowire connectivity.
- When MPLS Core Connectivity Type is set as LOCAL connectivity type, the N-PE Pseudo-wire on SVI attribute is always disabled in the policy and service request.
- For examples of these cases, see configlet examples EVC (Pseudowire Core Connectivity, Bridge Domain, Pseudowire on SVI), page 3-212 and EVC (Pseudowire Core Connectivity, no Bridge Domain, no Pseudowire on SVI), page 3-213.
- For additional information on the N-PE Pseudo-wire on SVI attribute, see the corresponding coverage in the EVC policy section in the section Setting the Interface Attributes, page 3-28.
- The N-PE Pseudo-wire on SVI attribute is not supported for IOS XR devices. All the xconnect commands are configured on L2 subinterfaces/service instance.

Step 12 Check the **PW Tunnel Selection** check box if you want to be able to manually select the Traffic Engineering (TE) tunnel for the pseudowire connecting point-to-point N-PEs.

Usage notes:

- Checking the PW Tunnel Selection check box activates the Interface Tunnel attribute field (see the next step).
- This attribute only appears if the MPLS core connectivity type is set as pseudowire in the EVC policy.

Step 13 If you checked the PW Tunnel Selection check box, enter the TE tunnel ID in the **Interface Tunnel** text field.

Prime Fulfillment uses the tunnel information to create and provision a pseudowire class that describes the pseudowire connection between two N-PEs. This pseudowire class can be shared by more than one pseudowire, as long as the pseudowires share the same tunnel ID and remote loopback address. During service request creation, Prime Fulfillment does not check the validity of the tunnel ID number. That is, Prime Fulfillment does not verify the existence of the tunnel.

Step 14 Check the **AutoPick Bridge Domain/VLAN ID** check box to have Prime Fulfillment autopick the VLAN ID during service request creation.

If this check box is unchecked, you are prompted to specify a VLAN ID during service request creation (see the next step).

Usage notes:

- AutoPick Bridge Domain/VLAN ID consumes a global VLAN ID on the device.
- The bridge domain VLAN ID is picked from the existing Prime Fulfillment VLAN pool.

Step 15 If the AutoPick Bridge Domain/VLAN ID check box is unchecked, enter an ID number in the **Bridge Domain/VLAN ID** text field.

Usage notes:

- If AutoPick Bridge Domain/VLAN ID is checked, this field is non-editable.
- When a VLAN ID is manually allocated, Prime Fulfillment verifies the VLAN ID to see if it lies within Prime Fulfillment's VLAN ID pool. If the VLAN ID is in the pool but not allocated, the VLAN ID is allocated to the service request. If the VLAN ID is in the pool and is already in use, Prime Fulfillment prompts you to allocate a different VLAN ID. If the VLAN ID lies outside of the Prime Fulfillment VLAN ID pool, Prime Fulfillment does not perform any verification about whether the VLAN ID allocated. The operator must ensure the VLAN ID is available.

Step 16 Click **OK** to save the ATM UNI Details settings and return to the EVC Service Request Editor window.

The value in the Link Attributes column now displays as "Changed," signifying that the link settings have been updated. You can edit the link attributes now or at a future time by clicking on the Changed link and modifying the settings in the Standard UNI Details window.

See Modifying the EVC Service Request, page 3-53 for details on editing the link attributes.

Step 17 To add another link click the **Add** button and set the attributes for the new link as in the previous steps in this section.

Step 18 To delete a link, check the check box in the first column of the row for that link and click the **Delete** button.

Step 19 If you want to set up links with L2 access nodes for this service request, see Setting Links with L2 Access Nodes, page 3-52.

Step 20 When you have completed setting the attributes in the EVC Service Request Editor window, click the **Save** button at the bottom of the window to save the settings and create the EVC service request.

If any attributes are missing or incorrectly set, Prime Fulfillment displays a warning in the lower left of the window. Make any corrections or updates needed (based on the information provided by Prime Fulfillment), and click the **Save** button.

For information on modifying an EVC service request see the section Modifying the EVC Service Request, page 3-53. For additional information about saving an EVC service request, see Saving the EVC Service Request, page 3-54.

Setting Links with L2 Access Nodes

The Links with L2 Access Nodes section of the EVC Service Request Editor window allows you to set up links with L2 (Ethernet) access nodes. These are similar to direct connect links, except that they have L2/Ethernet access nodes beyond the N-PE (towards the CE). Therefore, NPCs are involved.

Note ATM links are not supported in L2 access nodes. ATM links must be set up as direct connect links. For more information, see Setting the ATM Link Attributes, page 3-85.

The steps for setting up links with L2 access nodes are similar to those covered in the section Setting Direct Connect Links, page 3-41. See that section for detailed steps on the following common operations:

- Adding and deleting links.
- Selecting the N-PE.
- Choosing the UNI interface.
- Setting the link as an EVC link.
- Editing the standard and EVC link attributes.

The main difference in setting up links with L2 access does is specifying the NPC details.

To set the NPC details for links with L2 access nodes, perform the following steps.

Step 1 The first step in the process of adding a link using NPCs is selecting the U-PE/PE-AGG device, rather than the N-PE.

If only one NPC exists for the chosen interface, that NPC is autopopulated in the Circuit Details column, and you need not choose it explicitly.

If more then one NPC is available, click **Select one circuit** in the Circuit Selection column. The NPC window appears, enabling you to choose the appropriate NPC.

Step 2 Click **OK**.

Each time you choose a PE and its interface, the NPC that was set up from this PE and interface is automatically displayed under **Circuit Selection**. This means that you do not have to further specify the PE to complete the link.

If you want to review the details of this NPC, click **Circuit Details** in the Circuit Details column. The NPC Details window appears and lists the circuit details for this NPC.

Step 3 For details about editing link attributes, adding or deleting links, or using the EVC check box, see the corresponding steps in the section Setting Direct Connect Links, page 3-41.

Step 4 When you have completed setting the attributes in the EVC Service Request Editor window, click the **Save** button at the bottom of the window to save the settings and create the EVC service request.

If any attributes are missing or incorrectly set, Prime Fulfillment displays a warning in the lower left of the window. Make any corrections or updates needed (based on the information provided by Prime Fulfillment), and click the **Save** button.

For information on modifying an EVC service request see the section Modifying the EVC Service Request, page 3-53. For additional information about saving an EVC service request, see Saving the EVC Service Request, page 3-54.

Modifying the EVC Service Request

You can modify an EVC service request if you must change or modify the links or other settings of the service request.

To modify an EVC service request, perform the following steps.

Step 1 Choose **Operate > Service Requests > Service Request Manager**.

The Service Request Manager window appears, showing service request available in Prime Fulfillment.

Step 2 Check a check box for a service request.

Step 3 Click **Edit**.

EVC Service Request Editor window appears.

Step 4 Modify any of the attributes, as desired.

See the sections start with Setting the Service Request Details, page 3-35, for detailed coverage of setting attributes in this window.

Note Once the VC ID, VPLS VPN ID, and VLAN ID have been set in a service request they cannot be modified.

Step 5 To add a template/data file to an attachment circuit, see the section Using Templates and Data Files with an EVC Ethernet Service Request, page 3-53.

Step 6 When you are finished editing the EVC service request, click **Save**.

For additional information about saving an EVC service request, see Saving the EVC Service Request, page 3-54.

Using Templates and Data Files with an EVC Service Request

Prime Fulfillment does not support configuration of all the available CLI commands on a device being managed by the application. In order to configure such commands on the devices, you can use Prime Fulfillment Template Manager functionality. Templates can be associated at the policy level on a per-device role basis. Templates can be overridden at service request level, if the policy-level setting permits the operator to do so.

To associate templates and data files in a service request select any link in the EVC Service Request Editor window and click the **Template** button at the bottom of the window.

Note If the template feature has not been enabled in the associated policy then the Template button will not be available for selection.

The SR Template Association window appears. In this window, you can associate templates at a per-device level.

TheTemplate Association window lists the devices comprising the link, the device roles, and the template(s)/data file(s) associated with the devices. In this case, the template(s)/data file(s) have not yet been set up.

For further instructions on how to associate templates and data files with a service request, see Using Templates with Service Requests, page 9-24.

Saving the EVC Service Request

To save an EVC service request, perform the following steps.

Step 1 When you have finished setting the attributes for the EVC service request, click **Save** to create the service request.

If the EVC service request is successfully created, you will see the Service Request Manager window.

The newly created EVC service request is added with the state of REQUESTED.

Step 2 If, however, the EVC service request creation failed for some reason (for example, a value chosen is out of bounds), you are warned with an error message.

In such a case, you should correct the error and save the service request again.

Step 3 If you are ready to deploy the EVC service request, see Deploying Service Requests, page 8-10.

Creating an L2VPN Policy

This section covers the basic steps to create an L2VPN policy. It contains the following subsections:

Defining an L2VPN Policy

You must define an L2VPN policy before you can provision a Prime Fulfillment service. An L2VPN policy defines the common characteristics shared by the end-to-end wire attributes and Attachment Circuit (AC) attributes.

A policy is a template of most of the parameters needed to define an L2VPN service request. After you define it, an L2VPN policy can be used by all the L2VPN service requests that share a common set of characteristics. You create a new L2VPN policy whenever you create a new type of service or a service with different parameters. L2VPN policy creation is normally performed by experienced network engineers.

A policy can be shared by one or more service requests that have similar service requirements. The Editable check box gives the network operator the option of making a field editable. If the value is set to editable, the service request creator can change to other valid values for the particular policy item. If the value is *not* set to editable, the service request creator cannot change the policy item.

You can also associate Prime Fulfillment templates and data files with a policy. See Chapter 9, "Managing Templates and Data Files" for more about using templates and data files in policies.

It is also possible to create user-defined attributes within a policy (and service requests based on the policy). For background information on how to use the additional information feature, see Appendix H, "Adding Additional Information to Services."

The four major categories of an L2VPN policy correspond to the four major services that L2VPN provides:

- Point-to-point Ethernet Relay Service (ERS). The Metro Ethernet Forum (MEF) name for this service is Ethernet Virtual Private Line (EVPL). For more information about terms used to denote L2VPN services in this guide, see the section "Layer 2 Terminology Conventions" in the L2VPN Concepts chapter in the *Cisco Prime Fulfillment Theory of Operations Guide 6.2*.
- Point-to-point Ethernet Wire Service (EWS). The MEF name for this service is Ethernet Private Line (EPL).
- Frame Relay over MPLS (FRoMPLS)
- ATM over MPLS (ATMoMPLS)

To define an L2VPN policy in Prime Fulfillment, perform the following steps.

Step 1 Choose **Service Design > Policies > Policy Manager**.

The Policy Manager window appears.

Step 2 Click **Create**.

The Policy Editor window appears.

Step 3 Choose **L2VPN** from the Policy Type drop-down list.

The Policy Editor window appears.

Step 4 Enter a **Policy Name** for the L2VPN policy.

Step 5 Choose the **Policy Owner** for the L2VPN policy.

There are three types of L2VPN policy ownership:

- Customer ownership
- Provider ownership
- Global ownership—Any service operator can make use of this L2VPN policy.

This ownership has relevance when the Prime Fulfillment Role-Based Access Control (RBAC) comes into play. For example, an L2VPN policy that is customer-owned can only be seen by operators who are allowed to work on this customer-owned policy.

Similarly, operators who are allowed to work on a provider's network can view, use, and deploy a particular provider-owned policy.

Step 6 Click **Select** to choose the owner of the L2VPN.

(If you choose Global ownership, the Select function is not available.) The Select Customer window or the Select Provider window appears and you can choose an owner of the policy and click **Select**.

Step 7 Choose the **Service Type** of the L2VPN policy.

There are four service types for L2VPN policies:

- L2VPN ERS (EVPL)
- L2VPN EWS (EPL)
- Frame Relay
- ATM

Subsequent sections cover setting up the policies for each of these services.

Step 8 Check the **CE Present** check box if you want Prime Fulfillment to ask the service operator who uses this L2VPN policy to provide a CE router and interface during service activation.

The default is CE present in the service.

If you do not check the **CE Present** check box, Prime Fulfillment asks the service operator, during service activation, only for the U-PE or the N-PE router and customer-facing interface.

Step 9 Click **Next**.

The next sections contain examples of setting policies for the service types, with and without a CE present.

Defining an Ethernet ERS (EVPL) Policy with a CE

This section describes defining an Ethernet ERS (EVPL) policy with CE present.

Perform the following steps.

Step 1 In the Service Information window of the Policy Editor, choose **L2VPN ERS** for the Policy Type.

Step 2 Check the **CE Present** check box.

Step 3 Click **Next**.

The Interface Type window appears.

Step 4 Check the **Standard UNI Port** check box to enable port security.

This is the default. When you uncheck the check box, the port is treated as an uplink with no security features, and the window dynamically changes to eliminate items related to port security.

Note The **Standard UNI Port** attribute will be unavailable within service requests based on this policy if the UNI is on an N-PE device running IOS XR.

Step 5 Choose an **Interface Type** from the drop-down list.

You can choose a particular interface on a U-PE or N-PE interface based on the service provider's POP design. The interfaces are:

- **ANY** (Any interface can be chosen.)
- **Port-Channel** (A bundle of ports that share the same characteristics—this gives the service provider the ability to aggregate bandwidth and protection.)
- **Ethernet**
- **FastEthernet**

- **GE-WAN**
- **GigabitEthernet**
- **TenGigabitEthernet**
- **TenGigE**

The value defined here functions as a filter to restrict the interface types an operator can see during L2VPN service request creation.

Step 6 Enter an **Interface Format** as the slot number/port number for the CE interface (for example, **1/0** indicates that the interface is located at slot 1, port 0).

This is especially useful to specify here if you know that the link will always go through a particular interface's slot/port location on all or most of the network devices in the service.

Step 7 Choose an **Encapsulation** type.

The choices are:

- **DOT1Q**
- **DEFAULT**

If **DEFAULT** is the CE encapsulation type, Prime Fulfillment shows another field for the UNI port type.

Note If the Interface Type is ANY, Prime Fulfillment will not ask for an **Encapsulation** type in the policy.

Step 8 Check the **UNI Shutdown** check box if you want to leave the UNI port shut during service activation, for example, when the service provider wants to deploy a service in the network but wants to activate it at a later time.

Step 9 Check the **Keep Alive** check box to configure keepalives on the UNI port.

By default, this check box is unchecked, which causes the command **no keepalive** to be provisioned on the UNI port. This prevents a CPE from sending keepalive packets to the U-PE, for security purposes. This attribute is editable to support modification on a per-service request basis.

Step 10 Check the **ANY** check box to display all interface types as choices for the UNI interface (when creating service requests based on this policy).

This check box is checked by default.

Step 11 Check the **UNI** check box to display all interfaces defined as type UNI as choices for the UNI interface (when creating service requests based on this policy).

This check box is checked by default.

Step 12 Check the **VLAN ID AutoPick** check box if you want Prime Fulfillment to choose a VLAN ID.

If you do not check this check box, you will be prompted to provide the VLAN in a Provider VLAN ID field during service activation.

Step 13 Check the **VC ID AutoPick** check box if you want Prime Fulfillment to choose a VC ID.

If you do not check this check box, you will be prompted to provide the VC ID in a VC ID field during service activation.

Step 14 Enter a **VLAN NAME** (optional) to specify a name to describe the VLAN.

The name must be one token (no spaces allowed.) The limit for the VLAN name is 32 characters. The name has to be unique.

Step 15 Check the **Use PseudoWireClass** check box to enable the selection of a pseudowire class.

This attribute is only applicable for IOS XR devices. If the check box is checked, an additional attribute, **PseudoWireClass**, appears in the GUI. Click the **Select** button of PseudoWireClass attribute to choose a pseudowire class previously created in Prime Fulfillment. The pseudowire class name is used for provisioning pw-class commands on IOS XR devices. See Creating and Modifying Pseudowire Classes, page 3-14, for additional information on pseudowire class support for IOS XR devices.

Step 16 Choose an **L2VPN Group Name** from the drop-down list.

The choices are:

- **ISC**
- **VPNSC**

This attribute is used for provisioning the L2VPN group name on IOS XR devices.

Note The choices in the drop-down list are derived from a configurable DCPL property. For information about how to define the L2VPN Group Name choices available in the drop-down list, see Defining L2VPN Group Names for IOS XR Devices, page 3-18.

Step 17 Enter an **E-Line Name** to specify the point-to-point (p2p) E-line name.

This attribute is only applicable for IOS XR devices. If no value is specified for the **p2p** name, Prime Fulfillment generates a default name consisting of the names of the two PEs forming the pseudowire, separated by hyphens (for example, 6503-A----6503-B). If the default name is more than 32 characters, the device names are truncated.

Step 18 Enter a **Link Media** type (optional) of None, auto-select, rj45, or sfp.

Usage notes:

- The default is None.
- When this attribute is used, a new CLI will be generated in the UNI interface to define the media type.
- The Link Media attribute is supported only for ME3400 platforms.

Step 19 Enter a **Link Speed** (optional) of None, 10, 100, 1000, Auto, or nonegotiate.

Step 20 Enter a **Link Duplex** (optional) of None, Full, Half, or Auto.

Step 21 Check the **Use Existing ACL Name** check box if you want assign your own named access list to the port.

By default, this box is unchecked and Prime Fulfillment automatically assigns a MAC-based ACL on the customer facing UNI port, based on values you enter in **UNI MAC addresses** (below).

Step 22 Enter a **Port-Based ACL Name** (if you checked the **Use Existing ACL Name** check box, as mentioned in the previous step).

Note Prime Fulfillment does not create this ACL automatically. The ACL must already exist on the device, or be added as part of a template, before the service request is deployed. Otherwise, deployment will fail.

Step 23 Enter one or more Ethernet MAC addresses in **UNI MAC addresses**.

This selection is present only if you uncheck the **Use Existing ACL Name** check box. Click the **Edit** button to bring up a pop-up window in which you enter MAC addresses to be allowed or denied on the port. You can also specify a range of addresses by setting a base MAC address and a filtered MAC address.

Step 24 Choose a **UNI Port Type**.

The choices are:

- **Access Port**
- **Trunk with Native VLAN**

Note Enter a UNI Port Type only if the encapsulation type is DEFAULT.

Step 25 Check the **UNI Port Security** check box if you to want to provision port security-related CLIs to the UNI port by controlling the MAC addresses that are allowed to go through the interface.

a. For **Maximum Number of MAC address**, enter the number of MAC addresses allowed for port security.

b. For **Aging,** enter the length of time the MAC address can stay on the port security table.

c. For **Violation Action**, choose what action will occur when a port security violation is detected:

- **PROTECT**—Drops packets with unknown source addresses until a sufficient number of secure MAC addresses are removed to drop below the maximum value.
- **RESTRICT**—Drops packets with unknown source addresses until a sufficient number of secure MAC addresses are removed to drop below the maximum value and causes the Security Violation counter to increment.
- **SHUTDOWN**—Puts the interface into the error-disabled state immediately and sends an SNMP trap notification.

d. In the **Secure MAC Addresses** field, enter one or more Ethernet MAC addresses.

Step 26 Check the **Enable Storm Control** check box to help prevent the UNI port from being disrupted by a broadcast, multicast, or unicast storm.

Enter a threshold value for each type of traffic. The value, which can be specified to two significant digits, represents the percentage of the total available bandwidth of the port. If the threshold of a traffic type is reached, further traffic of that type is suppressed until the incoming traffic falls below the threshold level.

Step 27 Check the **N-PE Pseudo-wire On SVI** check box to configure the pseudowire connection on the switched virtual interface of the OSM card.

This check box is checked by default. If the check box is not checked, the pseudowire will be provisioned on the subinterface of the PFC card, if it is available. This option is only available for C76xx devices.

Note The **N-PE Pseudo-wire on SVI** attribute will be unavailable within service requests based on this policy for devices running IOS XR.

Step 28 Specify the type of **VLAN Translation** for this policy by clicking the appropriate radio button.

The choices are:

- **No**—No VLAN translation is performed. (This is the default.)
- **1:1**—1:1 VLAN translation.
- **2:1**—2:1 VLAN translation.

Note For detailed coverage of setting up VLAN translation, see Setting Up VLAN Translation for L2VPN ERS (EVPL) Services, page 3-171.

Step 29 Check the **PW Tunnel Selection** check box if you want to be able to manually select the Traffic Engineering (TE) tunnel for the pseudowire connecting point-to-point N-PEs.

This attribute is unchecked by default

Subsequently, when you create a service request based on this policy, you must specify the TE tunnel ID in a field provided. Prime Fulfillment uses the tunnel information to create and provision a pseudowire class that describes the pseudowire connection between two N-PEs. This pseudowire class can be shared by more than one pseudowire, as long as the pseudowires share the same tunnel ID and remote loopback address. You are responsible to ensure that the tunnel interface and associated ID are configured. During service request creation when you specify the tunnel ID number, Prime Fulfillment does not check the validity of the value. That is, Prime Fulfillment does not verify the existence of the tunnel.

Note The **PW Tunnel Selection** attribute will be unavailable within service requests based on this policy for devices running IOS XR.

Step 30 Click the **Next** button, if you want to enable template support for the policy.

The Template Association window appears. In this window, you can enable template support and, optionally, associate templates and data files with the policy. For instructions about associating templates with policies and how to use the features in this window, see Using Templates with Service Requests, page 9-24. When you have completed setting up templates and data files for the policy, click **Finish** in the Template Association window to close it and return to the Policy Editor window.

Note An additional window appears the policy workflow before the Template Association window. This window allows you to create user-defined attributes within the policy (and service requests based on the policy). For background information on how to use the additional information feature, see Appendix H, "Adding Additional Information to Services." If you are not using this feature, click **Next** to proceed to the Template Association window, or else click **Finish** to save the policy.

Step 31 Click **Finish**.

Defining an Ethernet ERS (EVPL) Policy without a CE

This section describes defining an Ethernet ERS (EVPL) policy without a CE present.

Perform the following steps.

Step 1 In the Service Information window of the Policy Editor, choose **L2VPN ERS** for the Policy Type.

Step 2 Uncheck the **CE Present** check box.

Step 3 Click **Next**.

The Interface Type window appears.

Step 4 Choose a N-PE/U-PE **Interface Type** from the drop-down list.

You can choose a particular interface as a CE, N-PE, or U-PE interface based on the service provider's POP design. The interfaces are:

- **ANY** (Any interface can be chosen.)
- **Port-Channel** (A bundle of ports that share the same characteristics—this gives the service provider the ability to aggregate bandwidth and protection.)
- **Ethernet**
- **FastEthernet**
- **GE-WAN**
- **GigabitEthernet**
- **TenGigabitEthernet**
- **TenGigE**

The value defined here functions as a filter to restrict the interface types an operator can see during L2VPN service request creation.

Step 5 Check the **Standard UNI Port** check box to enable port security.

This is the default. When you uncheck the check box, the port is treated as an uplink with no security features, and the window dynamically changes to eliminate items related to port security.

Note The **Standard UNI Port** attribute will be unavailable within service requests based on this policy if the UNI is on an N-PE device running IOS XR.

Step 6 Enter an **Interface Format** as the slot number/port number for the PE interface (for example, **1/0** indicates that the interface is located at slot 1, port 0).

This is especially useful to specify here if you know that the link will always go through a particular interface's slot/port location on all or most of the network devices in the service.

Step 7 Choose an **Encapsulation** type.

The choices are:

- **DOT1Q**
- **DEFAULT**

If **DEFAULT** is the CE encapsulation type, Prime Fulfillment shows another field for the UNI port type.

Note If the Interface Type is ANY, Prime Fulfillment will not ask for an **Encapsulation** type in the policy.

Step 8 Check the **UNI Shutdown** check box if you want to leave the UNI port shut during service activation, for example, when the service provider wants to deploy a service in the network but wants to activate it at a later time.

Step 9 Check the **Keep Alive** check box to configure keepalives on the UNI port.

By default, this check box is unchecked, which causes the command **no keepalive** to be provisioned on the UNI port. This prevents a CPE from sending keepalive packets to the U-PE, for security purposes. This attribute is editable to support modification on a per-service request basis.

Step 10 Check the **ANY** check box to display all interface types as choices for the UNI interface (when creating service requests based on this policy).

This check box is checked by default.

Step 11 Check the **UNI** check box to display all interfaces defined as type UNI as choices for the UNI interface (when creating service requests based on this policy).

This check box is checked by default.

Step 12 Check the **VLAN ID AutoPick** check box if you want Prime Fulfillment to choose a VLAN ID. If you do not check this check box, you will be prompted to provide the VLAN in a Provider VLAN ID field during service activation.

Step 13 Check the **VC ID AutoPick** check box if you want Prime Fulfillment to choose a VC ID.

If you do not check this check box, you will be prompted to provide the VC ID in a VC ID field during service activation.

Step 14 Check the **Use PseudoWireClass** check box to enable the selection of a pseudowire class.

This attribute is only applicable for IOS XR devices. If the check box is checked, an additional attribute, **PseudoWireClass**, appears in the GUI. Click the **Select** button of PseudoWireClass attribute to choose a pseudowire class previously created in Prime Fulfillment. The pseudowire class name is used for provisioning pw-class commands on IOS XR devices. See Creating and Modifying Pseudowire Classes, page 3-14, for additional information on pseudowire class support for IOS XR devices.

Step 15 Choose an **L2VPN Group Name** from the drop-down list.

The choices are:

- **ISC**
- **VPNSC**

This attribute is used for provisioning the L2VPN group name on IOS XR devices.

Note The choices in the drop-down list are derived from a configurable DCPL property. For information about how to define the L2VPN Group Name choices available in the drop-down list, see Defining L2VPN Group Names for IOS XR Devices, page 3-18.

Step 16 Enter an **E-Line Name** to specify the point-to-point (p2p) E-line name.

This attribute is only applicable for IOS XR devices. If no value is specified for the p2p name, Prime Fulfillment generates a default name consisting of the names of the two PEs forming the pseudowire, separated by hyphens (for example, 6503-A----6503-B). If the default name is more than 32 characters, the device names are truncated.

Step 17 Enter a **VLAN NAME** (optional) to specify a name to describe the VLAN.

The name must be one token (no spaces allowed.) The limit for the VLAN name is 32 characters. The name has to be unique.

Step 18 Enter a **Link Media** type (optional) of None, auto-select, rj45, or sfp.

Usage notes:

- The default is None.
- When this attribute is used, a new CLI will be generated in the UNI interface to define the media type.
- The Link Media attribute is supported only for ME3400 platforms.

Step 19 Enter a **Link Speed** (optional) of None, 10, 100, 1000, Auto, or nonegotiate.

Step 20 Enter a **Link Duplex** (optional) of None, Full, Half, or Auto.

Step 21 Check the **Use Existing ACL Name** check box if you want assign your own named access list to the port.

By default, this check box is unchecked and Prime Fulfillment automatically assigns a MAC-based ACL on the customer facing UNI port, based on values you enter in **UNI MAC addresses** (below).

Step 22 Enter a **Port-Based ACL Name** (if you checked the **Use Existing ACL Name** check box, as mentioned in the previous step).

Note Prime Fulfillment does not create this ACL automatically. The ACL must already exist on the device, or be added as part of a template, before the service request is deployed. Otherwise, deployment will fail.

Step 23 Enter one or more Ethernet MAC addresses in **UNI MAC addresses**.

This selection is present only if you unchecked the **Use Existing ACL Name** check box. Click the **Edit** button to bring up a pop-up window in which you enter MAC addresses to be allowed or denied on the port. You can also specify a range of addresses by setting a base MAC address and a filtered MAC address.

Step 24 Choose a **UNI Port Type**.

The choices are:

- **Access Port**
- **Trunk with Native VLAN**

Note Enter a UNI Port Type only if the encapsulation type is DEFAULT.

Step 25 Check the **UNI Port Security** check box if you to want to provision port security-related CLIs to the UNI port by controlling the MAC addresses that are allowed to go through the interface.

a. For **Maximum Number of MAC address**, enter the number of MAC addresses allowed for port security.

b. For **Aging,** enter the length of time the MAC address can stay on the port security table.

c. For **Violation Action**, choose what action will occur when a port security violation is detected:

- **PROTECT**—Drops packets with unknown source addresses until a sufficient number of secure MAC addresses are removed to drop below the maximum value.
- **RESTRICT**—Drops packets with unknown source addresses until a sufficient number of secure MAC addresses are removed to drop below the maximum value and causes the Security Violation counter to increment.
- **SHUTDOWN**—Puts the interface into the error-disabled state immediately and sends an SNMP trap notification.

d. In the **Secure MAC Addresses** field, enter one or more Ethernet MAC addresses.

Step 26 Check the **Enable Storm Control** check box to help prevent the UNI port from being disrupted by a broadcast, multicast, or unicast storm.

Enter a threshold value for each type of traffic. The value, which can be specified to two significant digits, represents the percentage of the total available bandwidth of the port. If the threshold of a traffic type is reached, further traffic of that type is suppressed until the incoming traffic falls below the threshold level.

Step 27 Check the **N-PE Pseudo-wire On SVI** check box to configure the pseudowire connection on the switched virtual interface of the OSM card.

This check box is checked by default. If the check box is not checked, the pseudowire will be provisioned on the subinterface of the PFC card, if it is available. This option is only available for C76xx devices.

Note The **N-PE Pseudo-wire On SVI** attribute will be unavailable within service requests based on this policy for devices running IOS XR.

Step 28 Specify the type of **VLAN Translation** for this policy by clicking the appropriate radio button.

The choices are:

- **No**—No VLAN translation is performed. (This is the default.)
- **1:1**—1:1 VLAN translation.
- **2:1**—2:1 VLAN translation.

Note For detailed coverage of setting up VLAN translation, see Setting Up VLAN Translation for L2VPN ERS (EVPL) Services, page 3-171.

Step 29 Check the **PW Tunnel Selection** check box if you want to be able to manually select the Traffic Engineering (TE) tunnel for the pseudowire connecting point-to-point N-PEs.

This attribute is unchecked by default

Subsequently, when you create a service request based on this policy, you must specify the TE tunnel ID in a field provided. Prime Fulfillment uses the tunnel information to create and provision a pseudowire class that describes the pseudowire connection between two N-PEs. This pseudowire class can be shared by more than one pseudowire, as long as the pseudowires share the same tunnel ID and remote loopback address. You are responsible to ensure that the tunnel interface and associated ID are configured. During service request creation when you specify the tunnel ID number, Prime Fulfillment does not check the validity of the value. That is, Prime Fulfillment does not verify the existence of the tunnel.

Note The **PW Tunnel Selection** attribute will be unavailable within service requests based on this policy for devices running IOS XR.

Step 30 Click the **Next** button, if you want to enable template support for the policy.

The Template Association window appears. In this window, you can enable template support and, optionally, associate templates and data files with the policy. For instructions about associating templates with policies and how to use the features in this window, see Using Templates with Service Requests, page 9-24. When you have completed setting up templates and data files for the policy, click **Finish** in the Template Association window to close it and return to the Policy Editor window.

Note An additional window appears the policy workflow before the Template Association window. This window allows you to create user-defined attributes within the policy (and service requests based on the policy). For background information on how to use the additional information feature, see Appendix H, "Adding Additional Information to Services." If you are not using this feature, click **Next** to proceed to the Template Association window, or else click **Finish** to save the policy.

Step 31 Click **Finish**.

Defining an Ethernet EWS (EPL) Policy with a CE

This section describes defining an Ethernet EWS (EPL) policy with CE present.

Perform the following steps.

Step 1 In the Service Information window of the Policy Editor, choose **L2VPN EWS** for the Policy Type.

Step 2 Check the **CE Present** check box.

Step 3 Click **Next**.

The Interface Type window appears.

Step 4 Check the **Standard UNI Port** check box to enable port security.

This is the default. When you uncheck the check box, the port is treated as an uplink with no security features, and the window dynamically changes to eliminate items related to port security.

Note The **Standard UNI Port** attribute will be unavailable within service requests based on this policy if the UNI is on an N-PE device running IOS XR.

Note In previous releases, the only Layer 2 VPN support for EWS (EPL) was from EWS (EPL) to EWS (EPL). In ISC 4.1.2 and later, support is also from EWS (EPL) to Network to Network Interface (NNI) as a trunk port. To create this new type of service request, you need to create an EWS (EPL) "hybrid" policy by unchecking the standard UNI flag. When using the EWS (EPL) hybrid policy for service request creation, check the **Standard UNI Port flag** for the EWS (EPL) side of the connection and uncheck the standard UNI flag for the NNI side of the connection.

Note In the case of hybrid services, UNI on an N-PE running IOS XR is not supported.

Step 5 Choose an **Interface Type** from the drop-down list.

You can choose a particular interface on a U-PE or N-PE interface based on the service provider's POP design. The interfaces are:

- **ANY** (Any interface can be chosen.)
- **Port-Channel** (A bundle of ports that share the same characteristics—this gives the service provider the ability to aggregate bandwidth and protection.)
- **Ethernet**
- **FastEthernet**
- **GE-WAN**
- **GigabitEthernet**
- **TenGigabitEthernet**
- **TenGigE**

The value defined here functions as a filter to restrict the interface types an operator can see during L2VPN service request creation.

Step 6 Enter an **Interface Format** as the slot number/port number for the CE interface (for example, **1/0** indicates that the interface is located at slot 1, port 0).

This is especially useful to specify here if you know that the link will always go through a particular interface's slot/port location on all or most of the network devices in the service.

Step 7 Choose an **Encapsulation** type.

The choices are:

- **DOT1Q**
- **DEFAULT**

If **DEFAULT** is the CE encapsulation type, Prime Fulfillment shows another field for the UNI port type.

Note If the Interface Type is ANY, Prime Fulfillment will not ask for an **Encapsulation** type in the policy.

Step 8 Check the **UNI Shutdown** check box if you want to leave the UNI port shut during service activation, for example, when the service provider wants to deploy a service in the network but wants to activate it at a later time.

Step 9 Check the **Keep Alive** check box to configure keepalives on the UNI port.

By default, this check box is unchecked, which causes the command **no keepalive** to be provisioned on the UNI port. This prevents a CPE from sending keepalive packets to the U-PE, for security purposes. This attribute is editable to support modification on a per-service request basis.

Step 10 Check the **ANY** check box to display all interface types as choices for the UNI interface (when creating service requests based on this policy).

This check box is checked by default.

Step 11 Check the **UNI** check box to display all interfaces defined as type UNI as choices for the UNI interface (when creating service requests based on this policy).

This check box is checked by default.

Step 12 Check the **VLAN ID AutoPick** check box if you want Prime Fulfillment to choose a VLAN ID.

If you do not check this check box, you will be prompted to provide the VLAN in a Provider VLAN ID field during service activation.

Step 13 Check the **VC ID AutoPick** check box if you want Prime Fulfillment to choose a VC ID.

If you do not check this check box, you will be prompted to provide the VC ID in a VC ID field during service activation.

Step 14 Check the **Use PseudoWireClass** check box to enable the selection of a pseudowire class.

This attribute is only applicable for IOS XR devices. If the check box is checked, an additional attribute, **PseudoWireClass**, appears in the GUI. Click the **Select** button of PseudoWireClass attribute to choose a pseudowire class previously created in Prime Fulfillment. The pseudowire class name is used for provisioning pw-class commands on IOS XR devices. See Creating and Modifying Pseudowire Classes, page 3-14 for additional information on pseudowire class support for IOS XR devices.

Step 15 Choose an **L2VPN Group Name** from the drop-down list.

The choices are:

- **ISC**
- **VPNSC**

This attribute is used for provisioning the L2VPN group name on IOS XR devices.

Note The choices in the drop-down list are derived from a configurable DCPL property. For information about how to define the L2VPN Group Name choices available in the drop-down list, see Defining L2VPN Group Names for IOS XR Devices, page 3-18.

Step 16 Enter an **E-Line Name** to specify the point-to-point (p2p) E-line name.

This attribute is only applicable for IOS XR devices. If no value is specified for the p2p name, Prime Fulfillment generates a default name consisting of the names of the two PEs forming the pseudowire, separated by hyphens (for example, 6503-A----6503-B). If the default name is more than 32 characters, the device names are truncated.

Step 17 Enter a **VLAN NAME** (optional) to specify a name to describe the VLAN.

The name must be one token (no spaces allowed.) The limit for the VLAN name is 32 characters. The name has to be unique.

Step 18 Check the **VLAN ID AutoPick** check box if you want Prime Fulfillment to choose a VLAN ID. If you do not check this check box, you will be prompted to provide the VLAN in a Provider VLAN ID field during service activation.

Step 19 Enter a **Link Media** type (optional) of None, auto-select, rj45, or sfp.

Usage notes:

- The default is None.
- When this attribute is used, a new CLI will be generated in the UNI interface to define the media type.
- The Link Media attribute is supported only for ME3400 platforms.

Step 20 Enter a **Link Speed** (optional) of None, 10, 100, 1000, Auto, or nonegotiate.

Step 21 Enter a **Link Duplex** (optional) of None, Full, Half, or Auto.

Step 22 Check the **Use Existing ACL Name** check box if you want assign your own named access list to the port. By default, this check box is not checked and Prime Fulfillment automatically assigns a MAC-based ACL on the customer facing UNI port, based on values you enter in **UNI MAC addresses** (below).

Step 23 Enter a **Port-Based ACL Name** (if you checked the **Use Existing ACL Name** check box, as mentioned in the previous step).

Note Prime Fulfillment does not create this ACL automatically. The ACL must already exist on the device, or be added as part of a template, before the service request is deployed. Otherwise, deployment will fail.

Step 24 Enter one or more Ethernet MAC addresses in **UNI MAC addresses**.

This selection is present only if you uncheck the **Use Existing ACL Name** check box. Click the **Edit** button to bring up a pop-up window in which you enter MAC addresses to be allowed or denied on the port. You can also specify a range of addresses by setting a base MAC address and a filtered MAC address.

Step 25 Check the **UNI Port Security** check box if you to want to provision port security-related CLIs to the UNI port by controlling the MAC addresses that are allowed to go through the interface.

a. For **Maximum Number of MAC address**, enter the number of MAC addresses allowed for port security.

b. For **Aging,** enter the length of time the MAC address can stay on the port security table.

c. For **Violation Action**, choose what action will occur when a port security violation is detected:

- **PROTECT**—Drops packets with unknown source addresses until a sufficient number of secure MAC addresses are removed to drop below the maximum value.
- **RESTRICT**—Drops packets with unknown source addresses until a sufficient number of secure MAC addresses are removed to drop below the maximum value and causes the Security Violation counter to increment.
- **SHUTDOWN**—Puts the interface into the error-disabled state immediately and sends an SNMP trap notification.

d. In the **Secure MAC Addresses** field, enter one or more Ethernet MAC addresses.

Step 26 Check the **Enable Storm Control** check box to help prevent the UNI port from being disrupted by a broadcast, multicast, or unicast storm. Enter a threshold value for each type of traffic.

The value, which can be specified to two significant digits, represents the percentage of the total available bandwidth of the port. If the threshold of a traffic type is reached, further traffic of that type is suppressed until the incoming traffic falls below the threshold level.

Step 27 Check the **Protocol Tunnelling** check box if you want to define the Layer 2 Bridge Protocol Data Unit (BPDU) frames that can be tunneled over the core to the other end.

For each protocol that you choose, enter the shutdown threshold and drop threshold for that protocol:

a. **Enable cdp**—Enable Layer 2 tunnelling on Cisco Discover Protocol (CDP).

b. **cdp shutdown threshold—**Enter the number of packets per second to be received before the interface is shut down.

c. **cdp drop threshold**—Enter the number of packets per second to be received at which point the interface will start dropping CDP packets.

d. **Enable vtp**—Enable Layer 2 tunnelling on VLAN Trunk Protocol (VTP).

e. **vtp shutdown threshold**—Enter the number of packets per second to be received before the interface is shut down.

f. **vtp drop threshold**—Enter the number of packets per second to be received at which point the interface will start dropping VTP packets.

g. **Enable stp—**Enable Layer 2 tunnelling on Spanning Tree Protocol (STP).

h. **stp shutdown threshold—**Enter the number of packets per second to be received before the interface is shut down.

i. **stp drop threshold**—Enter the number of packets per second to be received at which point the interface will start dropping STP packets.

j. **Recovery Interval**—Enter the amount of time, in seconds, to wait before recovering a UNI port.

Step 28 Check the **N-PE Pseudo-wire On SVI** check box to configure the pseudowire connection on the switched virtual interface of the OSM card.

This check box is checked by default. If the check box is not checked, the pseudowire will be provisioned on the subinterface of the PFC card, if it is available. This option is only available for C76xx devices.

Note The **N-PE Pseudo-wire On SVI** attribute will be unavailable within service requests based on this policy for devices running IOS XR.

Step 29 Enter the **MTU Size** in bytes.

The maximum transmission unit (MTU) size is configurable and optional. The default size is 9216, and the range is 1500 to 9216. Prime Fulfillment does not perform an integrity check for this customized value. If a service request goes to the Failed Deploy state because this size is not accepted, you must adjust the size until the Service Request is deployed.

In Cisco Prime Fulfillment 1.0, different platforms support different ranges.

- For the 3750 and 3550 platforms, the MTU range is 1500-1546.
- For the 7600 ethernet port, the MTU size is always 9216. Even with the same platform and same IOS release, different line cards support the MTU differently. For example, older line cards only take an MTU size of 9216 and newer cards support 1500-9216. However, Cisco Prime Fulfillment 1.0 uses 9216 in both cases.
- For the 7600 SVI (interface VLAN), the MTU size is 1500-9216.

Step 30 Check the **PW Tunnel Selection** check box if you want to be able to manually select the Traffic Engineering (TE) tunnel for the pseudowire connecting point-to-point N-PEs.

This attribute is unchecked by default

Subsequently, when you create a service request based on this policy, you must specify the TE tunnel ID in a field provided. Prime Fulfillment uses the tunnel information to create and provision a pseudowire class that describes the pseudowire connection between two N-PEs. This pseudowire class can be shared by more than one pseudowire, as long as the pseudowires share the same tunnel ID and remote loopback address. You are responsible to ensure that the tunnel interface and associated ID are configured. During service request creation when you specify the tunnel ID number, Prime Fulfillment does not check the validity of the value. That is, Prime Fulfillment does not verify the existence of the tunnel.

Note The **PW Tunnel Selection** attribute will be unavailable within service requests based on this policy for devices running IOS XR.

Step 31 Click the **Next** button, if you want to enable template support for the policy.

The Template Association window appears. In this window, you can enable template support and, optionally, associate templates and data files with the policy. For instructions about associating templates with policies and how to use the features in this window, see Using Templates with Service Requests, page 9-24. When you have completed setting up templates and data files for the policy, click **Finish** in the Template Association window to close it and return to the Policy Editor window.

Note An additional window appears the policy workflow before the Template Association window. This window allows you to create user-defined attributes within the policy (and service requests based on the policy). For background information on how to use the additional information feature, see Appendix H, "Adding Additional Information to Services." If you are not using this feature, click **Next** to proceed to the Template Association window, or else click **Finish** to save the policy.

Step 32 Click **Finish**.

Defining an Ethernet EWS (EPL) Policy without a CE

This section describes how to define an Ethernet EWS (EPL) policy without a CE present.

Perform the following steps.

Step 1 In the Service Information window of the Policy Editor, choose **L2VPN EWS** for the Policy Type.

Step 2 Uncheck the **CE Present** check box.

Step 3 Click **Next**.

The Interface Type window appears.

Step 4 Choose a N-PE/U-PE **Interface Type** from the drop-down list.

You can choose a particular interface as a CE, N-PE, or U-PE interface based on the service provider's POP design. The interfaces are:

- **ANY** (Any interface can be chosen.)
- **Port-Channel** (A bundle of ports that share the same characteristics—this gives the service provider the ability to aggregate bandwidth and protection.)
- **Ethernet**
- **FastEthernet**
- **GE-WAN**
- **GigabitEthernet**
- **TenGigabitEthernet**
- **TenGigE**

The value defined here functions as a filter to restrict the interface types an operator can see during L2VPN service request creation.

Step 5 Check the **Standard UNI Port** check box to enable port security.

This is the default. When you uncheck the check box, the port is treated as an uplink with no security features, and the window dynamically changes to eliminate items related to port security.

Note The **Standard UNI Port** attribute will be unavailable within service requests based on this policy if the UNI is on an N-PE device running IOS XR.

Note In previous releases, the only Layer 2 VPN support for EWS (EPL) was from EWS (EPL) to EWS (EPL). In ISC 4.1.2 and later, support is also from EWS (EPL) to Network to Network Interface (NNI) as a trunk port. To create this new type of service request, you need to create an EWS (EPL) "hybrid" policy by unchecking the standard UNI flag. When using the EWS (EPL) hybrid policy for service request creation, check the **Standard UNI Port flag** for the EWS (EPL) side of the connection and uncheck the standard UNI flag for the NNI side of the connection.

Note In the case of hybrid services, UNI on an N-PE running IOS XR is not supported.

Step 6 Enter an **Interface Format** as the slot number/port number for the PE interface (for example, **1/0** indicates that the interface is located at slot 1, port 0).

This is especially useful to specify here if you know that the link will always go through a particular interface's slot/port location on all or most of the network devices in the service.

Step 7 Choose an **Encapsulation** type.

The choices are:

- **DOT1Q**
- **DEFAULT**

If **DEFAULT** is the CE encapsulation type, Prime Fulfillment shows another field for the UNI port type.

Note If the Interface Type is ANY, Prime Fulfillment will not ask for an **Encapsulation** type in the policy.

Step 8 Check the **UNI Shutdown** check box if you want to leave the UNI port shut during service activation, for example, when the service provider wants to deploy a service in the network but wants to activate it at a later time.

Step 9 Check the **Keep Alive** check box to configure keepalives on the UNI port.

By default, this check box is unchecked, which causes the command **no keepalive** to be provisioned on the UNI port. This prevents a CPE from sending keepalive packets to the U-PE, for security purposes. This attribute is editable to support modification on a per-service request basis.

Step 10 Check the **ANY** check box to display all interface types as choices for the UNI interface (when creating service requests based on this policy).

This check box is not checked by default.

Step 11 Check the **UNI** check box to display all interfaces defined as type UNI as choices for the UNI interface (when creating service requests based on this policy).

This check box is not checked by default.

Step 12 Check the **VLAN ID AutoPick** check box if you want Prime Fulfillment to choose a VLAN ID.

If you do not check this check box, you will be prompted to provide the VLAN in a Provider VLAN ID field during service activation.

Step 13 Check the **VC ID AutoPick** check box if you want Prime Fulfillment to choose a VC ID.

If you do not check this check box, you will be prompted to provide the VC ID in a VC ID field during service activation.

Step 14 Check the **Use PseudoWireClass** check box to enable the selection of a pseudowire class.

This attribute is only applicable for IOS XR devices. If the check box is checked, an additional attribute, **PseudoWireClass**, appears in the GUI. Click the **Select** button of PseudoWireClass attribute to choose a pseudowire class previously created in Prime Fulfillment. The pseudowire class name is used for provisioning pw-class commands on IOS XR devices. See Creating and Modifying Pseudowire Classes, page 3-14, for additional information on pseudowire class support for IOS XR devices.

Step 15 Choose an **L2VPN Group Name** from the drop-down list.

The choices are:

- **ISC**
- **VPNSC**

This attribute is used for provisioning the L2VPN group name on IOS XR devices.

Note The choices in the drop-down list are derived from a configurable DCPL property. For information about how to define the L2VPN Group Name choices available in the drop-down list, see Defining L2VPN Group Names for IOS XR Devices, page 3-18.

Step 16 Enter an **E-Line Name** to specify the point-to-point (p2p) E-line name.

This attribute is only applicable for IOS XR devices. If no value is specified for the p2p name, Prime Fulfillment generates a default name consisting of the names of the two PEs forming the pseudowire, separated by hyphens (for example, 6503-A----6503-B). If the default name is more than 32 characters, the device names are truncated.

Step 17 Enter a **VLAN NAME** (optional) to specify a name to describe the VLAN.

The name must be one token (no spaces allowed.) The limit for the VLAN name is 32 characters. The name has to be unique.

Step 18 Enter a **Link Media** type (optional) of None, auto-select, rj45, or sfp.

Usage notes:

- The default is None.
- When this attribute is used, a new CLI will be generated in the UNI interface to define the media type.
- The Link Media attribute is supported only for ME3400 platforms.

Step 19 Enter a **Link Speed** (optional) of None, 10, 100, 1000, Auto, or nonegotiate.

Step 20 Enter a **Link Duplex** (optional) of None, Full, Half, or Auto.

Step 21 Check the **Use Existing ACL Name** check box if you want assign your own named access list to the port.

By default, this check box is not checked and Prime Fulfillment automatically assigns a MAC-based ACL on the customer facing UNI port, based on values you enter in **UNI MAC addresses** (below).

Step 22 Enter a **Port-Based ACL Name** (if you checked the **Use Existing ACL Name** check box, as mentioned in the previous step).

Note Prime Fulfillment does not create this ACL automatically. The ACL must already exist on the device, or be added as part of a template, before the service request is deployed. Otherwise, deployment will fail.

Step 23 Enter one or more Ethernet MAC addresses in **UNI MAC addresses**.

This selection is present only if you uncheck the **Use Existing ACL Name** check box. Click the **Edit** button to bring up a pop-up window in which you enter MAC addresses to be allowed or denied on the port. You can also specify a range of addresses by setting a base MAC address and a filtered MAC address.

Step 24 Check the **UNI Port Security** check box if you to want to provision port security-related CLIs to the UNI port by controlling the MAC addresses that are allowed to go through the interface.

a. For **Maximum Number of MAC address**, enter the number of MAC addresses allowed for port security.

b. For **Aging,** enter the length of time the MAC address can stay on the port security table.

c. For **Violation Action**, choose what action will occur when a port security violation is detected:

- **PROTECT**—Drops packets with unknown source addresses until a sufficient number of secure MAC addresses are removed to drop below the maximum value.
- **RESTRICT**—Drops packets with unknown source addresses until a sufficient number of secure MAC addresses are removed to drop below the maximum value and causes the Security Violation counter to increment.

- **SHUTDOWN**—Puts the interface into the error-disabled state immediately and sends an SNMP trap notification.

d. In the **Secure MAC Addresses** field, enter one or more Ethernet MAC addresses.

Step 25 Check the **Enable Storm Control** check box to help prevent the UNI port from being disrupted by a broadcast, multicast, or unicast storm.

Enter a threshold value for each type of traffic. The value, which can be specified to two significant digits, represents the percentage of the total available bandwidth of the port. If the threshold of a traffic type is reached, further traffic of that type is suppressed until the incoming traffic falls below the threshold level.

Step 26 Check the **Protocol Tunnelling** check box if you want to define the Layer 2 Bridge Protocol Data Unit (BPDU) frames that can be tunneled over the core to the other end.

For each protocol that you check, enter the shutdown threshold and drop threshold for that protocol:

a. **Enable cdp**—Enable Layer 2 tunnelling on Cisco Discover Protocol (CDP).

b. **cdp shutdown threshold—**Enter the number of packets per second to be received before the interface is shut down.

c. **cdp drop threshold**—Enter the number of packets per second to be received at which point the interface will start dropping CDP packets.

d. **Enable vtp**—Enable Layer 2 tunnelling on VLAN Trunk Protocol (VTP).

e. **vtp shutdown threshold**—Enter the number of packets per second to be received before the interface is shut down.

f. **vtp drop threshold**—Enter the number of packets per second to be received at which point the interface will start dropping VTP packets.

g. **Enable stp—**Enable Layer 2 tunnelling on Spanning Tree Protocol (STP).

h. **stp shutdown threshold—**Enter the number of packets per second to be received before the interface is shut down.

i. **stp drop threshold**—Enter the number of packets per second to be received at which point the interface will start dropping STP packets.

j. **Recovery Interval**—Enter the amount of time, in seconds, to wait before recovering a UNI port.

Step 27 Check the **N-PE Pseudo-wire On SVI** check box to configure the pseudowire connection on the switched virtual interface of the OSM card.

This check box is checked by default. If the check box is not checked, the pseudowire will be provisioned on the subinterface of the PFC card, if it is available. This option is only available for C76xx devices.

Note The **N-PE Pseudo-wire On SVI** attribute will be unavailable within service requests based on this policy for devices running IOS XR.

Step 28 Enter the **MTU Size** in bytes.

The maximum transmission unit (MTU) size is configurable and optional. The default size is 9216, and the range is 1500 to 9216. Prime Fulfillment does not perform an integrity check for this customized value. If a service request goes to the Failed Deploy state because this size is not accepted, you must adjust the size until the Service Request is deployed.

In Cisco Prime Fulfillment 1.0, different platforms support different ranges.

- For the 3750 and 3550 platforms, the MTU range is 1500-1546.

- For the 7600 ethernet port, the MTU size is always 9216. Even with the same platform and same IOS release, different line cards support the MTU differently. For example, older line cards only take an MTU size of 9216 and newer cards support 1500-9216. However, Cisco Prime Fulfillment 1.0 uses 9216 in both cases.
- For the 7600 SVI (interface VLAN), the MTU size is 1500-9216.

Step 29 Check the **PW Tunnel Selection** check box if you want to be able to manually select the Traffic Engineering (TE) tunnel for the pseudowire connecting point-to-point N-PEs.

This attribute is unchecked by default

Subsequently, when you create a service request based on this policy, you must specify the TE tunnel ID in a field provided. Prime Fulfillment uses the tunnel information to create and provision a pseudowire class that describes the pseudowire connection between two N-PEs. This pseudowire class can be shared by more than one pseudowire, as long as the pseudowires share the same tunnel ID and remote loopback address. You are responsible to ensure that the tunnel interface and associated ID are configured. During service request creation when you specify the tunnel ID number, Prime Fulfillment does not check the validity of the value. That is, Prime Fulfillment does not verify the existence of the tunnel.

Note The **PW Tunnel Selection** attribute will be unavailable within service requests based on this policy for devices running IOS XR.

Step 30 Click the **Next** button, if you want to enable template support for the policy.

The Template Association window appears. In this window, you can enable template support and, optionally, associate templates and data files with the policy. For instructions about associating templates with policies and how to use the features in this window, see Using Templates with Service Requests, page 9-24. When you have completed setting up templates and data files for the policy, click **Finish** in the Template Association window to close it and return to the Policy Editor window.

Note An additional window appears the policy workflow before the Template Association window. This window allows you to create user-defined attributes within the policy (and service requests based on the policy). For background information on how to use the additional information feature, see Appendix H, "Adding Additional Information to Services." If you are not using this feature, click **Next** to proceed to the Template Association window, or else click **Finish** to save the policy.

Step 31 Click **Finish**.

Defining a Frame Relay Policy with a CE

This section describes how to define a Frame Relay policy with CE present.

Perform the following steps.

Step 1 In the Service Information window of the Policy Editor, choose **Frame Relay** for the Policy Type.

Step 2 Check the **CE Present** check box.

Step 3 Click **Next**.

The Interface Type window appears.

Step 4 Check the **UNI Shutdown** check box if you want to leave the UNI port shut during service activation, for example, when the service provider wants to deploy a service in the network but wants to activate it at a later time.

Step 5 Choose the **Interface Type** for the **CE** from the drop-down list.

The choices are:

- **ANY**
- **Serial**
- **MFR**
- **POS**
- **Hssi**
- **BRI**

Step 6 Enter an **Interface Format** as the slot number/port number for the CE interface (for example, **1/0** indicates that the interface is located at slot 1, port 0).

This is especially useful to specify here if you know that the link will always go through a particular interface's slot/port location on all or most of the network devices in the service.

Step 7 Choose the CE Encapsulation type.

The choices are:

- **FRAME RELAY**
- **FRAME RELAY IETF**

Note If the Interface Type is ANY, Prime Fulfillment will not ask for an **Encapsulation** type in the policy.

Step 8 Check the **Use PseudoWireClass** check box to enable the selection of a pseudowire class.

This attribute is only applicable for IOS XR devices. If the check box is checked, an additional attribute, **PseudoWireClass**, appears in the GUI. Click the **Select** button of PseudoWireClass attribute to choose a pseudowire class previously created in Prime Fulfillment. The pseudowire class name is used for provisioning pw-class commands on IOS XR devices. See Creating and Modifying Pseudowire Classes, page 3-14, for additional information on pseudowire class support for IOS XR devices.

Step 9 Choose an **L2VPN Group Name** from the drop-down list.

The choices are:

- **ISC**
- **VPNSC**

This attribute is used for provisioning the L2VPN group name on IOS XR devices.

Note The choices in the drop-down list are derived from a configurable DCPL property. For information about how to define the L2VPN Group Name choices available in the drop-down list, see Defining L2VPN Group Names for IOS XR Devices, page 3-18.

Step 10 Enter an **E-Line Name** to specify the point-to-point (p2p) E-line name.

This attribute is only applicable for IOS XR devices. If no value is specified for the p2p name, Prime Fulfillment generates a default name consisting of the names of the two PEs forming the pseudowire, separated by hyphens (for example, 6503-A----6503-B). If the default name is more than 32 characters, the device names are truncated.

Step 11 Check the **PW Tunnel Selection** check box if you want to be able to manually select the Traffic Engineering (TE) tunnel for the pseudowire connecting point-to-point N-PEs.

This attribute is unchecked by default

Subsequently, when you create a service request based on this policy, you must specify the TE tunnel ID in a field provided. Prime Fulfillment uses the tunnel information to create and provision a pseudowire class that describes the pseudowire connection between two N-PEs. This pseudowire class can be shared by more than one pseudowire, as long as the pseudowires share the same tunnel ID and remote loopback address. You are responsible to ensure that the tunnel interface and associated ID are configured. During service request creation when you specify the tunnel ID number, Prime Fulfillment does not check the validity of the value. That is, Prime Fulfillment does not verify the existence of the tunnel.

Step 12 Click the **Next** button, if you want to enable template support for the policy.

The Template Association window appears. In this window, you can enable template support and, optionally, associate templates and data files with the policy. For instructions about associating templates with policies and how to use the features in this window, see Using Templates with Service Requests, page 9-24. When you have completed setting up templates and data files for the policy, click **Finish** in the Template Association window to close it and return to the Policy Editor window.

Note An additional window appears the policy workflow before the Template Association window. This window allows you to create user-defined attributes within the policy (and service requests based on the policy). For background information on how to use the additional information feature, see Appendix H, "Adding Additional Information to Services." If you are not using this feature, click **Next** to proceed to the Template Association window, or else click **Finish** to save the policy.

Step 13 Click **Finish**.

Defining a Frame Relay Policy without a CE

This section describes how to define a Frame Relay policy without a CE present. Perform the following steps.

Step 1 In the Service Information window of the Policy Editor, choose **Frame Relay** for the Policy Type.

Step 2 Uncheck the **CE Present** check box.

Step 3 Click **Next**.

The Interface Type window appears.

Step 4 Check the **UNI Shutdown** check box if you want to leave the UNI port shut during service activation, for example, when the service provider wants to deploy a service in the network but wants to activate it at a later time.

Step 5 Choose the N-PE/U-PE **Interface Type** for the **CE** from the drop-down list.

The choices are:

- **ANY**
- **Serial**
- **MFR**
- **POS**
- **Hssi**
- **BRI**

Step 6 Enter an **Interface Format** as the slot number/port number for the PE interface (for example, **1/0** indicates that the interface is located at slot 1, port 0).

This is especially useful to specify here if you know that the link will always go through a particular interface's slot/port location on all or most of the network devices in the service.

Step 7 Choose the N-PE/U-PE **Encapsulation** type.

The choices are:

- **FRAME RELAY**
- **FRAME RELAY IETF**

Note If the Interface Type is ANY, Prime Fulfillment will not ask for an **Encapsulation** type in the policy.

Step 8 Check the **Use PseudoWireClass** check box to enable the selection of a pseudowire class.

This attribute is only applicable for IOS XR devices. If the check box is checked, an additional attribute, **PseudoWireClass**, appears in the GUI. Click the **Select** button of PseudoWireClass attribute to choose a pseudowire class previously created in Prime Fulfillment. The pseudowire class name is used for provisioning pw-class commands on IOS XR devices. See Creating and Modifying Pseudowire Classes, page 3-14, for additional information on pseudowire class support for IOS XR devices.

Step 9 Choose an **L2VPN Group Name** from the drop-down list.

The choices are:

- **ISC**
- **VPNSC**

This attribute is used for provisioning the L2VPN group name on IOS XR devices.

Note The choices in the drop-down list are derived from a configurable DCPL property. For information about how to define the L2VPN Group Name choices available in the drop-down list, see Defining L2VPN Group Names for IOS XR Devices, page 3-18.

Step 10 Enter an **E-Line Name** to specify the point-to-point (p2p) E-line name.

This attribute is only applicable for IOS XR devices. If no value is specified for the p2p name, Prime Fulfillment generates a default name consisting of the names of the two PEs forming the pseudowire, separated by hyphens (for example, 6503-A----6503-B). If the default name is more than 32 characters, the device names are truncated.

Step 11 Check the **PW Tunnel Selection** check box if you want to be able to manually select the Traffic Engineering (TE) tunnel for the pseudowire connecting point-to-point N-PEs.

This attribute is unchecked by default

Subsequently, when you create a service request based on this policy, you must specify the TE tunnel ID in a field provided. Prime Fulfillment uses the tunnel information to create and provision a pseudowire class that describes the pseudowire connection between two N-PEs. This pseudowire class can be shared by more than one pseudowire, as long as the pseudowires share the same tunnel ID and remote loopback address. You are responsible to ensure that the tunnel interface and associated ID are configured. During service request creation when you specify the tunnel ID number, Prime Fulfillment does not check the validity of the value. That is, Prime Fulfillment does not verify the existence of the tunnel.

Step 12 Click the **Next** button, if you want to enable template support for the policy.

The Template Association window appears. In this window, you can enable template support and, optionally, associate templates and data files with the policy. For instructions about associating templates with policies and how to use the features in this window, see Using Templates with Service Requests, page 9-24. When you have completed setting up templates and data files for the policy, click **Finish** in the Template Association window to close it and return to the Policy Editor window.

Note An additional window appears the policy workflow before the Template Association window. This window allows you to create user-defined attributes within the policy (and service requests based on the policy). For background information on how to use the additional information feature, see Appendix H, "Adding Additional Information to Services." If you are not using this feature, click **Next** to proceed to the Template Association window, or else click **Finish** to save the policy.

Step 13 Click **Finish**.

Defining an ATM Policy with a CE

This section describes how to define an ATM policy with CE present.

Perform the following steps.

Step 1 In the Service Information window of the Policy Editor, choose **ATM** for the Policy Type.

Step 2 Check the **CE Present** check box.

Step 3 Click **Next**.

The Interface Type window appears.

Step 4 Choose the **Transport Mode** from the drop-down list.

The choices are:

- **VP**—Virtual path mode. This is the default.
- **VC**—Virtual circuit mode.
- **PORT**—Port mode. (Only supported for the IOS XR 3.7 platform.) Usage notes:
  - If you choose PORT as the transport mode, the attributes **ATM VCD/Sub-interface #** and **ATM VPI** will be disabled in the Link Attributes window of the service request based on this policy.
  - If you choose PORT as the transport mode, three attributes for setting timer values a will appear in the Link Attributes window of the service request based on this policy. These attributes are **Timer1**, **Timer2**, and **Timer3**. They are used to add timer values. The permissible range for these values is 50 to 4095. This feature is supported only for an N-PE as a UNI device.

– If you choose PORT as the transport mode, two attributes for setting cell packing will appear in the Link Attributes window of the service request based on this policy. These attributes are **Maximum no. of cells to be packed** and **Cell packing timer**. This feature is supported only for an N-PE as a UNI device.

Step 5 Choose the **CE Interface Type** from the drop-down list.

The choices are:

- **ANY**
- **ATM**
- **Switch**

Step 6 Enter an **Interface Format** as the slot number/port number for the CE interface (for example, **1/0** indicates that the interface is located at slot 1, port 0).

This is especially useful to specify here if you know that the link will always go through a particular interface's slot/port location on all or most of the network devices in the service.

Step 7 Choose a **CE Encapsulation**.

The choices are:

- **AAL5SNAP**
- **AAL5MUX**
- **AAL5NLPID**
- **AAL2**

Note If the Interface Type is ANY, Prime Fulfillment will not ask for an **Encapsulation** type in the policy.

Step 8 Check the **UNI Shutdown** check box if you want to leave the UNI port shut during service activation, for example, when the service provider wants to deploy a service in the network but wants to activate it at a later time.

Step 9 Check the **Use PseudoWireClass** check box to enable the selection of a pseudowire class.

This attribute is only applicable for IOS XR devices. If the check box is checked, an additional attribute, **PseudoWireClass**, appears in the GUI. Click the **Select** button of PseudoWireClass attribute to choose a pseudowire class previously created in Prime Fulfillment. The pseudowire class name is used for provisioning pw-class commands on IOS XR devices. See Creating and Modifying Pseudowire Classes, page 3-14, for additional information on pseudowire class support for IOS XR devices.

Step 10 Choose an **L2VPN Group Name** from the drop-down list.

The choices are:

- **ISC**
- **VPNSC**

This attribute is used for provisioning the L2VPN group name on IOS XR devices.

Note The choices in the drop-down list are derived from a configurable DCPL property. For information about how to define the L2VPN Group Name choices available in the drop-down list, see Defining L2VPN Group Names for IOS XR Devices, page 3-18.

Step 11 Enter an **E-Line Name** to specify the point-to-point (p2p) E-line name.

This attribute is only applicable for IOS XR devices. If no value is specified for the p2p name, Prime Fulfillment generates a default name consisting of the names of the two PEs forming the pseudowire, separated by hyphens (for example, 6503-A----6503-B). If the default name is more than 32 characters, the device names are truncated.

Step 12 Check the **PW Tunnel Selection** check box if you want to be able to manually select the Traffic Engineering (TE) tunnel for the pseudowire connecting point-to-point N-PEs.

This attribute is unchecked by default

Subsequently, when you create a service request based on this policy, you must specify the TE tunnel ID in a field provided. Prime Fulfillment uses the tunnel information to create and provision a pseudowire class that describes the pseudowire connection between two N-PEs. This pseudowire class can be shared by more than one pseudowire, as long as the pseudowires share the same tunnel ID and remote loopback address. You are responsible to ensure that the tunnel interface and associated ID are configured. During service request creation when you specify the tunnel ID number, Prime Fulfillment does not check the validity of the value. That is, Prime Fulfillment does not verify the existence of the tunnel.

Step 13 Click the **Next** button, if you want to enable template support for the policy.

The Template Association window appears. In this window, you can enable template support and, optionally, associate templates and data files with the policy. For instructions about associating templates with policies and how to use the features in this window, see Using Templates with Service Requests, page 9-24. When you have completed setting up templates and data files for the policy, click **Finish** in the Template Association window to close it and return to the Policy Editor window.

Note An additional window appears the policy workflow before the Template Association window. This window allows you to create user-defined attributes within the policy (and service requests based on the policy). For background information on how to use the additional information feature, see Appendix H, "Adding Additional Information to Services." If you are not using this feature, click **Next** to proceed to the Template Association window, or else click **Finish** to save the policy.

Step 14 Click **Finish**.

Defining an ATM Policy without a CE

This section describes how to define an ATM policy without a CE present.

Perform the following steps.

Step 1 In the Service Information window of the Policy Editor, choose **ATM** for the Policy Type.

Step 2 Uncheck the **CE Present** check box.

Step 3 Click **Next**.

The Interface Type window appears.

Step 4 Choose the **Transport Mode** from the drop-down list.

The choices are:

- **VP**—Virtual path mode. This is the default.
- **VC**—Virtual circuit mode.
- **PORT**—Port mode. (Only supported for the IOS XR 3.7 platform.) Usage notes:

- If you choose PORT as the transport mode, the attributes **ATM VCD/Sub-interface #** and **ATM VPI** will be disabled in the Link Attributes window of the service request based on this policy.
- If you choose PORT as the transport mode, three attributes for setting timer values a will appear in the Link Attributes window of the service request based on this policy. These attributes are **Timer1**, **Timer2**, and **Timer3**. They are used to add timer values. The permissible range for these values is 50 to 4095. This feature is supported only for an N-PE as a UNI device.
- If you choose PORT as the transport mode, two attributes for setting cell packing will appear in the Link Attributes window of the service request based on this policy. These attributes are **Maximum no. of cells to be packed** and **Cell packing timer**. This feature is supported only for an N-PE as a UNI device.

Step 5 Choose the **N-PE/U-PE Interface Type** from the drop-down list.

The choices are:

- **ANY**
- **ATM**
- **Switch**

Step 6 Enter an **Interface Format** as the slot number/port number for the PE interface (for example, **1/0** indicates that the interface is located at slot 1, port 0).

This is especially useful to specify here if you know that the link will always go through a particular interface's slot/port location on all or most of the network devices in the service.

Step 7 Choose a **PE Encapsulation**.

The choices are:

- **AAL5SNAP**
- **AAL5MUX**
- **AAL5NLPID**
- **AAL5**
- **AAL0**

Note If the Interface Type is ANY, Prime Fulfillment will not ask for an **Encapsulation** type in the policy.

Step 8 Check the **UNI Shutdown** check box if you want to leave the UNI port shut during service activation, for example, when the service provider wants to deploy a service in the network but wants to activate it at a later time.

Step 9 Check the **Use PseudoWireClass** check box to enable the selection of a pseudowire class.

This attribute is only applicable for IOS XR devices. If the check box is checked, an additional attribute, **PseudoWireClass**, appears in the GUI. Click the **Select** button of PseudoWireClass attribute to choose a pseudowire class previously created in Prime Fulfillment. The pseudowire class name is used for provisioning pw-class commands on IOS XR devices. See Creating and Modifying Pseudowire Classes, page 3-14, for additional information on pseudowire class support for IOS XR devices.

Step 10 Choose an **L2VPN Group Name** from the drop-down list.

The choices are:

- **ISC**
- **VPNSC**

This attribute is used for provisioning the L2VPN group name on IOS XR devices.

Note The choices in the drop-down list are derived from a configurable DCPL property. For information about how to define the L2VPN Group Name choices available in the drop-down list, see Defining L2VPN Group Names for IOS XR Devices, page 3-18.

Step 11 Enter an **E-Line Name** to specify the point-to-point (p2p) E-line name.

This attribute is only applicable for IOS XR devices. If no value is specified for the p2p name, Prime Fulfillment generates a default name consisting of the names of the two PEs forming the pseudowire, separated by hyphens (for example, 6503-A----6503-B). If the default name is more than 32 characters, the device names are truncated.

Step 12 Check the **PW Tunnel Selection** check box if you want to be able to manually select the Traffic Engineering (TE) tunnel for the pseudowire connecting point-to-point N-PEs.

This attribute is unchecked by default

Subsequently, when you create a service request based on this policy, you must specify the TE tunnel ID in a field provided. Prime Fulfillment uses the tunnel information to create and provision a pseudowire class that describes the pseudowire connection between two N-PEs. This pseudowire class can be shared by more than one pseudowire, as long as the pseudowires share the same tunnel ID and remote loopback address. You are responsible to ensure that the tunnel interface and associated ID are configured. During service request creation when you specify the tunnel ID number, Prime Fulfillment does not check the validity of the value. That is, Prime Fulfillment does not verify the existence of the tunnel.

Step 13 Click the **Next** button, if you want to enable template support for the policy.

The Template Association window appears. In this window, you can enable template support and, optionally, associate templates and data files with the policy. For instructions about associating templates with policies and how to use the features in this window, see Using Templates with Service Requests, page 9-24. When you have completed setting up templates and data files for the policy, click **Finish** in the Template Association window to close it and return to the Policy Editor window.

Note An additional window appears the policy workflow before the Template Association window. This window allows you to create user-defined attributes within the policy (and service requests based on the policy). For background information on how to use the additional information feature, see Appendix H, "Adding Additional Information to Services." If you are not using this feature, click **Next** to proceed to the Template Association window, or else click **Finish** to save the policy.

Step 14 Click **Finish**.

Managing an L2VPN Service Request

This section covers the basic steps to provision an ERS (EVPL), EWS (EPL), ATM, or Frame Relay L2VPN service. It contains the following subsections:

- Configuring Device Settings to Support Prime Fulfillment, page 3-6
- Creating an EVC Service Request, page 3-35
- Creating an ERS (EVPL), ATM, or Frame Relay L2VPN Service Request with a CE, page 3-120

Introducing L2VPN Service Requests

An L2VPN service request consists of one or more end-to-end wires, connecting various sites in a point-to-point topology. When you create a service request, you enter several parameters, including the specific interfaces on the CE and PE routers.

You can also associate Prime Fulfillment templates and data files with a service request. See Chapter 9, "Managing Templates and Data Files" for more about using templates and data files in service requests.

It is also possible to create user-defined attributes within a policy (and service requests based on the policy). For background information on how to use the additional information feature, see Appendix H, "Adding Additional Information to Services."

To create a service request, a Service Policy must already be defined, as described in Creating a VPLS Policy, page 3-130.

Based on the predefined L2VPN policy, an operator creates an L2VPN service request, with or without modifications to the L2VPN policy, and deploys the service. Service creation and deployment are normally performed by regular network technicians for daily operation of network provisioning.

Note Not all of the attributes defined in an L2VPN policy might be applicable to a service request. For specific information, see L2VPN policy attribute descriptions in Creating an L2VPN Policy, page 3-90.

The following steps are involved in creating a service request for Layer 2 connectivity between customer sites:

- Choose a CE Topology for ERS (EVPL)/Frame Relay/ATM services.
- Choose the endpoints (CE and PE) that must be connected. For each end-to-end Layer 2 connection, Prime Fulfillment creates an end-to-end wire object in the repository for the service request.
- Choose a CE or PE interface.
- Choose a Named Physical Circuit (NPC) for the CE or PE.
- Edit the end-to-end connection.
- Edit the link attributes.
- (0ptional) Associate templates and data files to devices in the service request.

For sample configlets for L2VPN scenarios, see Sample Configlets, page 3-176.

Creating an L2VPN Service Request

To create an L2VPN service request, perform the following steps.

Step 1 Choose **Operate > Service Requests > Service Request Manager**.

The Service Request Manager window appears.

Step 2 Click **Create**.

The Service Request Editor window appears.

Step 3 From the Policy drop-down list, choose an L2VPN policy from the policies previously created (see Creating an L2VPN Policy, page 3-90).

The L2VPN Service Request editor window appears.

The new service request inherits all the properties of the chosen L2VPN policy, such as all the editable and non-editable features and pre-set parameters.

Step 4 To continue creating an L2VPN service request, go to one of the following sections:

- Creating an ERS (EVPL), ATM, or Frame Relay L2VPN Service Request with a CE, page 3-120.
- Creating an EWS (EPL) L2VPN Service Request with a CE, page 3-122.
- Creating an ERS (EVPL), ATM, or Frame Relay L2VPN Service Request without a CE, page 3-124.
- Creating an EWS (EPL) L2VPN Service Request without a CE, page 3-126.

Creating an ERS (EVPL), ATM, or Frame Relay L2VPN Service Request with a CE

This section includes detailed steps for creating an L2VPN service request with a CE present for ERS (EVPL), ATM, and Frame Relay policies. If you are creating an L2VPN service request for an EWS (EPL) policy, go to Creating an EWS (EPL) L2VPN Service Request with a CE, page 3-122.

After you choose an L2VPN policy, the L2VPN Service Request Editor window appears.

Perform the following steps.

Step 1 Create the L2VPN service request for the policy.

The L2VPN Service Request Editor window appears.

Step 2 Choose a **Topology** from the drop-down list.

If you choose **Full Mesh**, each CE will have direct connections to every other CE.

If you choose **Hub and Spoke**, then only the Hub CE has connection to each Spoke CE and the Spoke CEs do not have direct connection to each other.

Note The full mesh and the hub and spoke topologies make a difference only when you choose more than two endpoints. For example, with four endpoints, Prime Fulfillment automatically creates six links with full mesh topology. With hub and spoke topology, however, Prime Fulfillment creates only three links.

Step 3 Click **Add Link**.

You specify the CE endpoints using the Attachment Tunnel Editor.

Note All the services that deploy point-to-point connections (ERS/EVPL, EWS/EPL, ATMoMPLS, and FRoMPLS) must have at least two CEs specified.

Step 4 Click **Select CE** in the CE column.

The Select CPE Device window appears.This window displays the list of currently defined CEs.

a. From the **Show CPEs with** drop-down list, you can display CEs by Customer Name, by Site, or by Device Name.

b. You can use the **Find** button to either search for a specific CE, or to refresh the display.

c. You can set the **Rows per page** to 5, 10, 20, 30, 40, or All.

Step 5 In the Select column, choose a CE for the L2VPN link.

Step 6 Click **Select**.

The Service Request Editor window appears displaying the name of the selected CE in the CE column.

Step 7 Choose the CE interface from the drop-down list.

Note When you provision an L2VPN ERS (EVPL) service, when you choose a UNI for a particular device, Prime Fulfillment determines if there are other services using the same UNI. If so, a warning message is displayed. If you ignore the message and save the service request, all of the underlying service requests relying on the same UNI are synchronized with the modified shared attributes of the latest service request. In addition, the state of the existing service requests is changed to the Requested state.

Note Prime Fulfillment only displays the available interfaces for the service, based on the configuration of the underlying interfaces, existing service requests that might be using the interface, and the customer associated with the service request. You can click the **Details** button to display a pop-up window with information on the available interfaces, such as interface name, customer name, VPN name and service request ID, service request type, VLAN translation type, and VLAN ID information.

Step 8 If only one NPC exists for the Chosen CE and CE interface, that NPC is autopopulated in the Circuit Selection column and you need not choose it explicitly. If more then one NPC is available, click **Select one circuit** in the Circuit Selection column.

The Select NPC window appears, enabling you to choose the appropriate NPC.

Step 9 Click **OK**.

Each time you choose a CE and its interface, the NPC that was precreated from this CE and interface is automatically displayed under **Circuit Selection**. This means that you do not have to further specify the PE to complete the link.

If you want to review the details of this NPC, click **Circuit Details** in the Circuit Details column. The NPC Details window appears and lists the circuit details for this NPC.

Step 10 Continue to specify additional CEs, as in previous steps.

Prime Fulfillment creates the links between CEs based on the Topology that you chose.

Step 11 Click **OK**.

For ERS (EVPL), ATM, and Frame Relay, the EndToEndWire window appears.

Step 12 The VPN for this service request appears in the **VPN** field.

If there is more than one VPN, click **Select VPN** to choose a VPN. The Select VPN window appears.

Step 13 Choose a **VPN Name** and click **Select**.

The L2VPN Service Request Editor window appears with the VPN name displayed.

Step 14 If necessary, click **Add AC** in the Attachment Circuit2 (AC2) column, and repeat Steps 3 to 10 for AC2.

The EndToEndWire window displays the complete end-to-end wire.

Step 15 Specify remaining items in the EndToEndWire window as necessary for your configuration:

- You can choose any of the blue highlighted values to edit the end-to-end wire.
- You can edit the AC link attributes to change the default policy settings. After you edit these fields, the blue link changes from Default to Changed. For more information, see the section Modifying the EVC Service Request, page 3-53.
- You can enter a description for the service request in the first **Description** field. The description will show up in this window and also in the Description column of the Service Requests window. The maximum length for this field is 256 characters.
- You can enter a description for each end-to-end wire in the **Description** field provided for each wire. The description shows up only in this window. The data in this field is not pushed to the device(s). The maximum length for this field is 256 characters.
- The ID number is system-generated identification number for the circuit.
- The Circuit ID is created automatically, based on the service. For example, for Ethernet, it is based on the VLAN number; for Frame Relay, it is based on the DLCI; for ATM, it is based on the VPI/VCI.
- If the policy was set up for you to define a VC ID manually, enter it into the empty **VC ID** field. If policy was set to “auto pick” the VC ID, Prime Fulfillment will supply a VC ID, and this field will not be editable. In the case where you supply the VC ID manually, if the entered value is in the provider’s range, Prime Fulfillment validates if the entered value is available or allocated. If the entered value has been already allocated, Prime Fulfillment generates an error message saying that the entered value is not available and prompts you to re-enter the value. If the entered value is in the provider’s range, and if it is available, then it is allocated and is removed from the VC ID pool. If the entered value is outside the provider’s range, Prime Fulfillment displays a warning saying that no validation could be performed to verify if it is available or allocated.
- You can also click **Add Link** to add an end-to-end wire.
- You can click **Delete Link** to delete an end-to-end wire.

Step 16 When you are finished editing the end-to-end wires, click **Save**.

The service request is created and saved into Prime Fulfillment.

Creating an EWS (EPL) L2VPN Service Request with a CE

This section includes detailed steps for creating an L2VPN service request with a CE present for EWS (EPL). If you are creating an L2VPN service request for an ERS (EVPL), ATM, or Frame Relay policy, go to Creating an ERS (EVPL), ATM, or Frame Relay L2VPN Service Request with a CE, page 3-120.

Perform the following steps.

Step 1 Create the L2VPN service request for EWS (EPL) with CE.

The L2VPN Service Request Editor window appears.

Step 2 Click **Select VPN** to choose a VPN for use with this CE.

The Select VPN window appears with the VPNs defined in the system.

Step 3 Choose a **VPN Name** in the Select column.

Step 4 Click **Select**.

The L2VPN Service Request Editor window appears with the VPN name displayed.

Step 5 Click **Add Link**.

- You can enter a description for the service request in the first **Description** field. The description will show up in this window and also in the Description column of the Service Request Editor window. The maximum length for this field is 256 characters.
- You can enter a description for each end-to-end wire in the **Description** field provided for each wire. The description shows up only in this window. The data in this field is not pushed to the device(s). The maximum length for this field is 256 characters.
- The ID number is system-generated identification number for the circuit.
- The Circuit ID is created automatically, based on the service. For example, for Ethernet, it is based on the VLAN number; for Frame Relay, it is based on the DLCI; for ATM, it is based on the VPI/VCI.

Step 6 Click **Add AC** in the Attachment Circuit1 (AC1) column.

The Customer and Link Selection window appears.

Step 7 Click **Select CE**.

The Select CPE Device window appears.

This window displays the list of currently defined CEs.

a. From the **Show CPEs with** drop-down list, you can display CEs by Customer Name, by Site, or by Device Name.

b. You can use the **Find** button to either search for a specific CE, or to refresh the display.

c. You can set the **Rows per page** to 5, 10, 20, 30, 40, or All.

Step 8 In the Select column, choose a CE for the L2VPN link.

Step 9 Click **Select**.

Step 10 In the Customer and Link Selection window, choose a CE interface from the drop-down list.

Step 11 If only one NPC exists for the Chosen CE and CE interface, that NPC is autopopulated in the Circuit Selection column and you need not choose it explicitly.

If more then one NPC is available, click **Select one circuit** in the Circuit Selection column. The Select NPC window appears, enabling you to choose the appropriate NPC. Each time you choose a CE and its interface, the NPC that was precreated from this CE and interface is automatically displayed under **Circuit Selection**. This means that you do not have to further specify the PE to complete the link.

Step 12 Click **OK**.

The EndToEndWire window appears displaying the name of the selected CE in the AC1 column.

Step 13 Click the Edit link in the AC1 Attributes column to edit the attributes of the attachment circuit if desired.

The Link Attributes window appears. Edit the attributes as desired. For more information, see the section Modifying the EVC Service Request, page 3-53.

Step 14 Click **OK**.

Step 15 Repeat Steps 6 through 14 for **AC2**.

Step 16 In the L2VPN Service Request Editor, click **Save**.

The EWS (EPL) service request is created and saved in Prime Fulfillment.

Creating an ERS (EVPL), ATM, or Frame Relay L2VPN Service Request without a CE

This section includes detailed steps for creating an L2VPN service request without a CE present for ERS (EVPL), ATM, and Frame Relay policies. If you are creating an L2VPN service request for an EWS (EPL) policy, go to Creating an EWS (EPL) L2VPN Service Request without a CE, page 3-126.

Perform the following steps.

Step 1 Create the L2VPN service request for ERS (EVPL) without a CE.

The L2VPN Service Request Editor window appears.

Step 2 Choose a **Topology** from the drop-down list.

If you choose **Full Mesh**, each CE will have direct connections to every other CE. If you choose **Hub and Spoke**, then only the Hub CE has connection to each Spoke CE and the Spoke CEs do not have direct connection to each other.

Note The full mesh and the hub and spoke topologies make a difference only when you choose more than two endpoints. For example, with four endpoints, Prime Fulfillment automatically creates six links with full mesh topology. With hub and spoke topology, however, Prime Fulfillment creates only three links.

Step 3 Click **Add Link**.

Step 4 Specify the N-PE/PE-AGG/U-PE endpoints, as covered in the following steps.

Step 5 Click **Select U-PE/PE-AGG/N-PE** in the U-PE/PE-AGG/N-PE column.

The Select PE Device window appears.

This window displays the list of currently defined PEs.

a. The **Show PEs with** drop-down list shows PEs by customer name, by site, or by device name.

b. The **Find** button allows a search for a specific PE or a refresh of the window.

c. The **Rows per page** drop-down list allows the page to be set to 5, 10, 20, 30, 40, or All.

Step 6 In the **Select** column, choose the PE device name for the L2VPN link.

Step 7 Click **Select**.

The L2VPN Service Request Editor window appears displaying the name of the selected PE in the N-PE/PE-AGG/U-PE column.

Step 8 Choose the UNI interface from the drop-down list.

Note When you provision an L2VPN ERS (EVPL) service, when you choose a UNI for a particular device, Prime Fulfillment determines if there are other services using the same UNI. If so, a warning message is displayed. If you ignore the message and save the service request, all of the underlying service requests lying on the same UNI are synchronized with the modified shared attributes of the latest service request. In addition, the state of the existing service requests is changed to the Requested state.

Note Prime Fulfillment only displays the available interfaces for the service, based on the configuration of the underlying interfaces, existing service requests that might be using the interface, and the customer associated with the service request. You can click the **Details** button to display a pop-up window with information on the available interfaces, such as interface name, customer name, VPN name and service request ID, service request type, VLAN translation type, and VLAN ID information.

Step 9 If the PE role type is U-PE, click **Select one circuit** in the Circuit Selection column.

The Select NPC window appears.

If only one NPC exists for the Chosen PE and PE interface, that NPC is auto populated in the Circuit Selection column and you need not choose it explicitly.

Note If the PE role type is N-PE, the columns Circuit Selection and Circuit Details are disabled.

Step 10 Choose the name of the NPC from the **Select** column.

Step 11 Click **OK**.

Each time you choose a PE and its interface, the NPC that was precreated from this PE and interface is automatically displayed under **Circuit Selection**. This means that you do not have to further specify the PE to complete the link.

Step 12 If you want to review the details of this NPC, click **Circuit Details** in the Circuit Details column.

The Select NPC Details window appears and lists the circuit details for this NPC.

Step 13 After you specify all the PEs, Prime Fulfillment creates the links between PEs based on the Topology that you chose.

Step 14 Click **OK**.

For ERS (EVPL), ATM, and Frame Relay, the EndToEndWire window appears.

Step 15 The VPN for this service request appears in the Select VPN field.

If there is more than one VPN, click **Select VPN** to choose a VPN.

Step 16 Specify remaining items in the EndToEnd Wire window, as necessary for your configuration:

- You can choose any of the blue highlighted values to edit the end-to-end wire.
- You can edit the AC link attributes to change the default policy settings. After you edit these fields, the blue link changes from Default to Changed. For more information, see the section Modifying the EVC Service Request, page 3-53.
- You can also click **Add Link** to add an end-to-end wire.
- You can click **Delete Link** to delete an end-to-end wire.

Note If you are attempting to decommission a service request to which a template has been added, see Decommissioning Service Requests, page 8-12, for information on the proper way to do this.

- You can enter a description for the service request in the first **Description** field. The description will show up in this window and also in the Description column of the Service Requests window. The maximum length for this field is 256 characters.

- You can enter a description for each end-to-end wire in the **Description** field provided for each wire. The description shows up only in this window. The data in this field is not pushed to the device(s). The maximum length for this field is 256 characters.
- The ID number is system-generated identification number for the circuit.
- The Circuit ID is created automatically, based on the service. For example, for Ethernet, it is based on the VLAN number; for Frame Relay, it is based on the DLCI; for ATM, it is based on the VPI/VCI.

Step 17 When you are finished editing the end-to-end wires, click **Save**.

The service request is created and saved into Prime Fulfillment.

Creating an EWS (EPL) L2VPN Service Request without a CE

This section includes detailed steps for creating an L2VPN service request without a CE present for EWS (EPL). If you are creating an L2VPN service request for an ERS (EVPL), ATM, or Frame Relay policy, see Creating an ERS (EVPL), ATM, or Frame Relay L2VPN Service Request without a CE, page 3-124.

Step 1 Create the L2VPN service request for EWS (EPL) without a CE.

The L2VPN Service Request Editor window appears.

Step 2 Click **Select VPN** to choose a VPN for use with this PE.

The Select VPN window appears with the VPNs defined in the system.

Step 3 Choose a **VPN Name** in the Select column.

Step 4 Click **Select**.

The EndToEndWire window appears with the VPN name displayed.

Step 5 Click **Add AC** in the Attachment Circuit 1(AC1) column.

The Customer and Link Selection window appears.

Step 6 Click **Select N-PE/PE-AGG/U-PE**. in the N-PE/PE-AGG/U-PE column.

The Select PE Device window appears.

This window displays the list of currently defined PEs.

a. From the **Show PEs with** drop-down list, you can display PEs by Customer Name, by Site, or by Device Name.

b. You can use the **Find** button to either search for a specific PE, or to refresh the display.

c. You can set the **Rows per page** to 5, 10, 20, 30, 40, or All.

Step 7 In the Select column, choose a PE for the L2VPN link.

Step 8 Click **Select**.

The Customer and Link Selection window appears.

Step 9 Choose a PE interface from the drop-down list.

Note Prime Fulfillment only displays the available interfaces for the service, based on the configuration of the underlying interfaces, existing service requests that might be using the interface, and the customer associated with the service request. You can click the **Details** button to display a pop-up window with information on the available interfaces, such as interface name, customer name, VPN name and service request ID, service request type, VLAN translation type, and VLAN ID information.

Step 10 If the PE role type is N-PE, the columns Circuit Selection and Circuit Details are disabled. In this case, skip to Step 13.

Step 11 If the PE role type is U-PE, click **Select one circuit** in the Circuit Selection column.

The Select NPC window appears.

Note If only one NPC exists for the Chosen PE and PE interface, that NPC is auto populated in the Circuit Selection column and you need not choose it explicitly.

Step 12 If applicable, choose the name of the NPC from the Select column.

Step 13 Click **OK**.

Note Each time you choose a PE and its interface, the NPC that was precreated from this PE and interface is automatically displayed under **Circuit Selection**. This means that you do not have to further specify the PE to complete the link.

Step 14 Click **OK**.

The l2VPN Service Request window appears displaying the name of the selected PE in the Attachment Circuit1 (AC1) column.

Step 15 Click the **Edit** link in the AC1 Attributes and edit the attributes, if desired.

For more information, see the section Modifying the EVC Service Request, page 3-53.

Step 16 Repeat Steps 5 through 14 for Attachment Circuit2.

Step 17 Specify remaining items in the EndToEndWire window, as necessary for your configuration.

- You can enter a description for the service request in the first **Description** field. The description will show up in this window and also in the Description column of the Service Requests window. The maximum length for this field is 256 characters.
- You can enter a description for each end-to-end wire in the **Description** field provided for each wire. The description shows up only in this window. The data in this field is not pushed to the device(s). The maximum length for this field is 256 characters.
- The ID number is system-generated identification number for the circuit.
- The Circuit ID is created automatically, based on the service. For example, for Ethernet, it is based on the VLAN number; for Frame Relay, it is based on the DLCI; for ATM, it is based on the VPI/VCI.

Step 18 Click **Save**.

The EWS (EPL) service request is created and saved in Prime Fulfillment.

Modifying the L2VPN Service Request

This section describes how to edit the L2VPN service request attributes. This is also where you can associate templates and data files to devices that are part of theattachment circuits.

Perform the following steps.

Step 1 Choose **Operate > Service Request > Service Request Manager**.

The L2VPN Service Request window appears.

Step 2 Check a check box for a service request.

Step 3 Click **Edit**.

The EndToEndWire window appears.

Step 4 Modify any of the attributes, as desired:

- The VPN for this service request appears in the Select VPN field. If this request has more than one VPN, click **Select VPN** to choose a VPN.
- You can choose any of the blue highlighted values to edit the end-to-end wire.
- You can edit the AC link attributes to change the default policy settings. After you edit these fields, the blue link changes from Default to Changed.
- You can enter a description for the service request in the first **Description** field. The description will show up in this window and also in the Description column of the Service Requests window. The maximum length for this field is 256 characters.
- You can enter a description for each end-to-end wire in the **Description** field provided for each wire. The description shows up only in this window. The data in this field is not pushed to the device(s). The maximum length for this field is 256 characters.
- The Circuit ID is created automatically, based on the VLAN data for the circuit.
- If the policy was set up for you to define a VC ID manually, enter it into the empty **VC ID** field. If policy was set to “auto pick” the VC ID, Prime Fulfillment will supply a VC ID, and this field will not be editable. In the case where you supply the VC ID manually, if the entered value is in the provider’s range, Prime Fulfillment validates if the entered value is available or allocated. If the entered value has been already allocated, Prime Fulfillment generates an error message saying that the entered value is not available and prompts you to re-enter the value. If the entered value is in the provider’s range, and if it is available, then it is allocated and is removed from the VC ID pool. If the entered value is outside the provider’s range, Prime Fulfillment displays a warning saying that no validation could be performed to verify if it is available or allocated.
- You can also click **Add Link** to add an end-to-end wire.
- You can click **Delete Link** to delete an end-to-end wire.

Note If you are attempting to decommission a service request to which a template has been added, see Decommissioning Service Requests, page 8-12 for information on the proper way to do this.

- The ID number is system-generated identification number for the circuit.
- The Circuit ID is created automatically, based on the service. For example, for Ethernet, it is based on the VLAN number; for Frame Relay, it is based on the DLCI; for ATM, it is based on the VPI/VCI.

Step 5 To edit AC attributes, click the **Default** link in the appropriate AC Attributes column.

The Link Attributes window appears.

Step 6 Edit any of the link attributes, as desired.

Step 7 To add a template and data file to an attachment circuit, choose a Device Name, and click **Add** under Templates.

The Add/Remove Templates window appears.

Note To add a template to an attachment circuit, you must have already created the template. For detailed steps to create templates, see Overview, page 9-1. For more information on how to use templates and data files in service requests, see Chapter 9, "Managing Templates and Data Files."

Step 8 Click **Add**.

The Template Data File Chooser window appears.

Step 9 In the left pane, navigate to and select a template.

The associated data files are listed in rows in the main window.

Step 10 Check the data file that you want to add and click **Accept**.

The Add/Remove Templates window appears with the template displayed.

Step 11 Choose a Template name.

Step 12 Under Action, use the drop-down list and choose **APPEND** or **PREPEND.**

Append tells Prime Fulfillment to append the template generated CLI to the regular Prime Fulfillment (non-template) CLI. Prepend is the reverse and does not append the template to the Prime Fulfillment CLI.

Step 13 Choose **Active** to use this template for this service request.

If you do not choose Active, the template is not used.

Step 14 Click **OK**.

The Link Attributes with the template added appears.

Note For more information about using templates and data files in service requests, see Chapter 9, "Managing Templates and Data Files."

Step 15 Click **OK**.

The L2VPN Service Request window appears showing the link in the AC Attachment Circuit column has changed from Default to Changed.

Step 16 When you are finished editing the end-to-end wires, click **Save**.

Saving the L2VPN Service Request

To save an L2VPN service request, perform the following steps.

Step 1 When you are finished specifying the link attributes for all the attachment circuits, click **Save** to finish the L2VPN service request creation.

If the L2VPN service request is successfully created, you will see it listed in the Service Request Manager window. The newly created L2VPN service request is added with the state of REQUESTED.

Step 2 If, however, the L2VPN service request creation failed for some reason (for example, a value chosen is out of bounds), you are warned with an error message. In such a case, you should correct the error and save the service request again.

For information on deploying L2VPN service requests, see Deploying Service Requests, page 8-10.

Creating a VPLS Policy

This section contains the basic steps to create a VPLS policy. It contains the following subsections:

Defining a VPLS Policy

You must define a VPLS policy before you can provision a service. A VPLS policy defines the common characteristics shared by the Attachment Circuit (AC) attributes.

A policy can be shared by one or more service requests that have similar service requirements. The Editable check box gives the network operator the option of making a field editable. If the value is set to editable, the service request creator can change to other valid values for the particular policy item. If the value is *not* set to editable, the service request creator cannot change the policy item.

You can also associate Prime Fulfillment templates and data files with a policy. See Chapter 9, "Managing Templates and Data Files" for more about using templates and data files in policies.

It is also possible to create user-defined attributes within a policy (and service requests based on the policy). For background information on how to use the additional information feature, see Appendix H, "Adding Additional Information to Services."

VPLS policies correspond to the one of the core types that VPLS provides:

- MPLS core type—provider core network is MPLS enabled
- Ethernet core type—provider core network uses Ethernet switches

and to one of the service types that VPLS provides:

- Ethernet Relay Multipoint Service (ERMS). The Metro Ethernet Forum name for ERMS is Ethernet Virtual Private LAN (EVP-LAN). For more information about terms used to denote VPLS services in this guide, see the section "Layer 2 Terminology Conventions" in the L2VPN Concepts chapter in the *Cisco Prime Fulfillment Theory of Operations Guide 6.2*.
- Ethernet Multipoint Service (EMS). The MEF name for EMS is Ethernet Private LAN (EP-LAN).

A policy is a template of most of the parameters needed to define a VPLS service request. After you define it, a VPLS policy can be used by all the VPLS service requests that share a common set of characteristics.

You create a new VPLS policy whenever you create a new type of service or a service with different parameters. VPLS policy creation is normally performed by experienced network engineers.

To define a VPLS policy in the Prime Fulfillment, perform the following steps.

Step 1 Choose **Service Design > Policies > Policy Manager**.

The Policy Manager window appears.

Step 2 Click **Create**.

The Policy Editor window appears.

Step 3 Choose **VPLS** from the Policy Type drop-down list.

The Policy Editor window appears.

Step 4 Enter a **Policy Name** for the VPLS policy.

Step 5 Choose the **Policy Owner** for the VPLS policy.

There are three types of VPLS policy ownership:

- Customer ownership
- Provider ownership
- Global ownership—Any service operator can make use of this VPLS policy.

This ownership has relevance when the Prime Fulfillment Role-Based Access Control (RBAC) comes into play. For example, a VPLS policy that is customer owned can only be seen by operators who are allowed to work on this customer-owned policy.

Similarly, operators who are allowed to work on a provider's network can view, use, and deploy a particular provider-owned policy.

Step 6 Click **Select** to choose the owner of the VPLS policy.

The policy owner was established when you created customers or providers during Prime Fulfillment setup. If the ownership is global, the Select function does not appear.

Step 7 Choose the **Core Type** of the VPLS policy.

There are two core types for VPLS policies:

- MPLS—running on an IP network
- Ethernet—all PEs are on an Ethernet provider network

Step 8 Choose the **Service Type** of the VPLS policy.

There are two service types for VPLS policies:

- Ethernet Relay Multipoint Service (ERMS). (The MEF name for ERMS is EVP-LAN.)
- Ethernet Multipoint Service (EMS). (The MEF name for EMS is EP-LAN.)

Step 9 Check the **CE Present** check box if you want Prime Fulfillment to ask the service operator who uses this VPLS policy to provide a CE router and interface during service activation.

The default is CE present in the service.

If you do not check the **CE Present** check box, Prime Fulfillment asks the service operator, during service activation, only for the PE router and customer-facing interface.

Defining an MPLS/ERMS (EVP-LAN) Policy with a CE

This section describes how to define a VPLS policy with an MPLS core type and an ERMS (EVP-LAN) service type with CE present.

Perform the following steps.

Step 1 In the Service Information window of the Policy Editor, choose **VPLS** for the Policy Type.

Step 2 For Core Type, choose **MPLS**.

Step 3 For Service Type, choose **Ethernet Relay Multipoint Service (ERMS)**.

Step 4 Check the **CE Present** check box.

Step 5 Click **Next**.

The Interface Type window appears.

Step 6 Choose an **Interface Type** from the drop-down list.

You can choose a particular interface on a CE, N-PE, PE-AGG, or U-PE interface based on the service provider's POP design. The interfaces are:

- **ANY** (Any interface can be chosen.)
- **Port-Channel** (A bundle of ports that share the same characteristics—this gives the service provider the ability to aggregate bandwidth and protection.)
- **Ethernet**
- **FastEthernet**
- **GE-WAN**
- **GigabitEthernet**
- **TenGigabitEthernet**
- **TenGigE**

The value defined here functions as a filter to restrict the interface types an operator can see during VPLS service request creation. If defined as ANY, the operator can see all interface types.

Step 7 Enter an **Interface Format** as the slot number/port number for the CE interface (for example, **1/0** indicates that the interface is located at slot 1, port 0).

This is especially useful to specify here if you know that the link will always go through a particular interface's slot/port location on all or most of the network devices in the service.

Step 8 Choose a CE **Encapsulation** type.

The choices are:

- **DOT1Q**

- **DEFAULT**

If **DEFAULT** is the CE encapsulation type, Prime Fulfillment shows another field for the UNI port type.

Step 9 Check the **Standard UNI Port** check box to enable port security.

This is the default. When you uncheck the check box, the port is treated as an uplink with no security features, and the window dynamically changes to eliminate items related to port security.

Step 10 Check **UNI Shutdown** box if you want to leave the UNI port shut during service activation, for example, when the service provider wants to deploy a service in the network but wants to activate it at a later time.

Step 11 Check the **Keep Alive** check box to configure keepalives on the UNI port.

By default, this check box is unchecked, which causes the command **no keepalive** to be provisioned on the UNI port. This prevents a CPE from sending keepalive packets to the U-PE, for security purposes. This attribute is editable to support modification on a per-service request basis.

Step 12 Check the **ANY** check box to display all interface types as choices for the UNI interface (when creating service requests based on this policy).

This check box is checked by default.

Step 13 Check the **UNI** check box to display all interfaces defined as type UNI as choices for the UNI interface (when creating service requests based on this policy).

This check box is checked by default.

Step 14 Enter one or more Ethernet MAC addresses in **UNI MAC addresses**.

This selection is present only if you uncheck the **Use Existing ACL Name** check box. Click the **Edit** button to bring up a pop-up window in which you enter MAC addresses to be allowed or denied on the port. You can also specify a range of addresses by setting a base MAC address and a filtered MAC address.

Step 15 Choose a **Port Type**.

The choices are:

- **Access Port**
- **Trunk with Native VLAN**

Step 16 Enter a **Link Speed** (optional) of None, 10, 100, 1000, Auto, or nonegotiate.

Step 17 Enter a **Link Duplex** (optional) of None, Full, Half, or Auto.

Step 18 In the **PE/UNI Interface Description** field, enter an optional description, for example *Customer-B ERMS (EVP-LAN) Service*.

Step 19 Check the **VLAN ID AutoPick** check box if you want Prime Fulfillment to choose a VLAN ID.

If you do not check this check box, you will be prompted to provide the VLAN in a Provider VLAN ID field during service activation.

Step 20 Enter a **VLAN NAME** (optional) to specify a name to describe the VLAN.

The name must be one token (no spaces allowed.) The limit for the VLAN name is 32 characters. The name has to be unique. Two VLANs cannot share the same name.

Step 21 Check the **Use Existing ACL Name** check box if you want assign your own named access list to the port.

By default, this check box is not checked and Prime Fulfillment automatically assigns a MAC-based ACL on the customer facing UNI port, based on values you enter in **UNI MAC addresses** (below).

Step 22 Enter a **Port-Based ACL Name** (if you checked the **Use Existing ACL Name** check box, as mentioned in the previous step).

Note Prime Fulfillment does not create this ACL automatically. The ACL must already exist on the device, or be added as part of a template, before the service request is deployed. Otherwise, deployment will fail.

Step 23 Check the **Disable CDP** check box if you want to disable the Cisco Discover Protocol (CDP) on the UNI port.

Step 24 Check the **Filter BPDU** check box to specify that the UNI port should not process Layer 2 Bridge Protocol Data Units (BPDUs).

Step 25 Check the **UNI Port Security** check box if you to want to provision port security-related CLIs to the UNI port by controlling the MAC addresses that are allowed to go through the interface.

a. For **Maximum Number of MAC address**, enter the number of MAC addresses allowed for port security.

b. For **Aging,** enter the length of time the MAC address can stay on the port security table.

c. For **Violation Action**, choose what action will occur when a port security violation is detected:

- **PROTECT**—Drops packets with unknown source addresses until a sufficient number of secure MAC addresses are removed to drop below the maximum value.
- **RESTRICT**—Drops packets with unknown source addresses until a sufficient number of secure MAC addresses are removed to drop below the maximum value and causes the Security Violation counter to increment.
- **SHUTDOWN**—Puts the interface into the error-disabled state immediately and sends an SNMP trap notification.

d. In the **Secure MAC Addresses** field, enter one or more Ethernet MAC addresses. Click the **Edit** button to enter the addresses.

Step 26 Check the **Enable Storm Control** check box to help prevent the UNI port from being disrupted by a broadcast, multicast, or unicast storm.

Enter a threshold value for each type of traffic. The value, which can be specified to two significant digits, represents the percentage of the total available bandwidth of the port. If the threshold of a traffic type is reached, further traffic of that type is suppressed until the incoming traffic falls below the threshold level.

Step 27 Click the **Next** button, if you want to enable template support for the policy.

The Template Association window appears. In this window, you can enable template support and, optionally, associate templates and data files with the policy. For instructions about associating templates with policies and how to use the features in this window, see Using Templates with Service Requests, page 9-24. When you have completed setting up templates and data files for the policy, click **Finish** in the Template Association window to close it and return to the Policy Editor window.

Note An additional window appears the policy workflow before the Template Association window. This window allows you to create user-defined attributes within the policy (and service requests based on the policy). For background information on how to use the additional information feature, see Appendix H, "Adding Additional Information to Services." If you are not using this feature, click **Next** to proceed to the Template Association window, or else click **Finish** to save the policy.

Step 28 Click **Finish**.

Note The VC ID is mapped from the VPN ID. By default, Prime Fulfillment will "auto pick" this value. However, you can set this manually, if desired. This is done by editing the associated VPN configuration. The Edit VPN window has an **Enable VPLS** check box. When you check this box, you can manually enter a VPN ID in a field provided. For more information on creating and modifying VPNs, see Setting Up Logical Inventory, page 2-53.

Defining an MPLS/ERMS (EVP-LAN) Policy without a CE

This section describes defining a VPLS policy with an MPLS core type and an ERMS (EVP-LAN) service type without a CE present.

Perform the following steps.

Step 1 In the Service Information window of the Policy Editor, choose **VPLS** for the Policy Type.

Step 2 For Core Type, choose **MPLS**.

Step 3 For Service Type, choose **Ethernet Relay Multipoint Service (ERMS)**.

Step 4 Uncheck the **CE Present** check box.

Step 5 Click **Next**.

The Interface Type window appears.

Step 6 Choose an **Interface Type** from the drop-down list.

You can choose a particular interface on a N-PE, U-PE, or PE-AGG interface based on the service provider's POP design. The interfaces are:

- **ANY** (Any interface can be chosen.)
- **Port-Channel** (A bundle of ports that share the same characteristics—this gives the service provider the ability to aggregate bandwidth and protection.)
- **Ethernet**
- **FastEthernet**
- **GE-WAN**
- **GigabitEthernet**
- **TenGigabitEthernet**
- **TenGigE**

The value defined here functions as a filter to restrict the interface types an operator can see during VPLS service request creation. If defined as ANY, the operator can see all interface types.

Step 7 Check the **Standard UNI Port** check box to enable port security.

This is the default. When you uncheck the check box, the port is treated as an uplink with no security features, and the window dynamically changes to eliminate items related to port security.

Step 8 Enter an **Interface Format** as the slot number/port number for the CE interface (for example, **1/0** indicates that the interface is located at slot 1, port 0).

This is especially useful to specify here if you know that the link will always go through a particular interface's slot/port location on all or most of the network devices in the service.

Step 9 Choose a CE **Encapsulation** type.

The choices are:

- **DOT1Q**
- **DEFAULT**

If **DEFAULT** is the CE encapsulation type, Prime Fulfillment shows another field for the UNI port type.

Step 10 Check **UNI Shutdown** box if you want to leave the UNI port shut during service activation, for example, when the service provider wants to deploy a service in the network but wants to activate it at a later time.

Step 11 Check the **Keep Alive** check box to configure keepalives on the UNI port.

By default, this check box is unchecked, which causes the command **no keepalive** to be provisioned on the UNI port. This prevents a CPE from sending keepalive packets to the U-PE, for security purposes. This attribute is editable to support modification on a per-service request basis.

Step 12 Check the **ANY** check box to display all interface types as choices for the UNI interface (when creating service requests based on this policy).

This check box is checked by default.

Step 13 Check the **UNI** check box to display all interfaces defined as type UNI as choices for the UNI interface (when creating service requests based on this policy). This check box is checked by default.

Step 14 Enter one or more Ethernet MAC addresses in **UNI MAC addresses**.

This selection is present only if you uncheck the **Use Existing ACL Name** check box. Click the **Edit** button to bring up a pop-up window in which you enter MAC addresses to be allowed or denied on the port. You can also specify a range of addresses by setting a base MAC address and a filtered MAC address.

Step 15 Choose a **Port Type**.

The choices are:

- **Access Port**
- **Trunk with Native VLAN**

Step 16 Enter a **Link Speed** (optional) of None, 10, 100, 1000, Auto, or nonegotiate.

Step 17 Enter a **Link Duplex** (optional) of None, Full, Half, or Auto.

Step 18 In the **PE/UNI Interface Description** field, enter an optional description, for example *Customer-B ERMS (EVP-LAN) Service*.

Step 19 Check the **VLAN ID AutoPick** check box if you want Prime Fulfillment to choose a VLAN ID.

If you do not check this check box, you will be prompted to provide the VLAN in a Provider VLAN ID field during service activation.

Step 20 Enter a **VLAN NAME** (optional) to specify a name to describe the VLAN.

The name must be one token (no spaces allowed.) The limit for the VLAN name is 32 characters. The name has to be unique. Two VLANs cannot share the same name.

Step 21 Check the **Use Existing ACL Name** check box if you want assign your own named access list to the port.

By default, this check box is not checked and Prime Fulfillment automatically assigns a MAC-based ACL on the customer facing UNI port, based on values you enter in **UNI MAC addresses** (below).

Step 22 Enter a **Port-Based ACL Name** (if you checked the **Use Existing ACL Name** check box, as mentioned in the previous step).

Note Prime Fulfillment does not create this ACL automatically. The ACL must already exist on the device, or be added as part of a template, before the service request is deployed. Otherwise, deployment will fail.

Step 23 Check the **Disable CDP** check box if you want to disable the Cisco Discover Protocol (CDP) on the UNI port.

Step 24 Check the **Filter BPDU** check box to specify that the UNI port should not process Layer 2 Bridge Protocol Data Units (BPDUs).

Step 25 Check the **UNI Port Security** check box if you to want to provision port security-related CLIs to the UNI port by controlling the MAC addresses that are allowed to go through the interface.

a. For **Maximum Number of MAC address**, enter the number of MAC addresses allowed for port security.

b. For **Aging,** enter the length of time the MAC address can stay on the port security table.

c. For **Violation Action**, choose what action will occur when a port security violation is detected:

- **PROTECT**—Drops packets with unknown source addresses until a sufficient number of secure MAC addresses are removed to drop below the maximum value.
- **RESTRICT**—Drops packets with unknown source addresses until a sufficient number of secure MAC addresses are removed to drop below the maximum value and causes the Security Violation counter to increment.
- **SHUTDOWN**—Puts the interface into the error-disabled state immediately and sends an SNMP trap notification.

d. In the **Secure MAC Addresses** field, enter one or more Ethernet MAC addresses.

Step 26 Check the **Enable Storm Control** check box to help prevent the UNI port from being disrupted by a broadcast, multicast, or unicast storm.

Enter a threshold value for each type of traffic. The value, which can be specified to two significant digits, represents the percentage of the total available bandwidth of the port. If the threshold of a traffic type is reached, further traffic of that type is suppressed until the incoming traffic falls below the threshold level.

Step 27 Click the **Next** button, if you want to enable template support for the policy.

The Template Association window appears. In this window, you can enable template support and, optionally, associate templates and data files with the policy. For instructions about associating templates with policies and how to use the features in this window, see Using Templates with Service Requests, page 9-24. When you have completed setting up templates and data files for the policy, click **Finish** in the Template Association window to close it and return to the Policy Editor window.

Note An additional window appears the policy workflow before the Template Association window. This window allows you to create user-defined attributes within the policy (and service requests based on the policy). For background information on how to use the additional information feature, see Appendix H, "Adding Additional Information to Services." If you are not using this feature, click **Next** to proceed to the Template Association window, or else click **Finish** to save the policy.

Step 28 Click **Finish**.

Note The VC ID is mapped from the VPN ID. By default, Prime Fulfillment will "auto pick" this value. However, you can set this manually, if desired. This is done by editing the associated VPN configuration. The Edit VPN window has an **Enable VPLS** check box. When you check this box, you can manually enter a VPN ID in a field provided. For more information on creating and modifying VPNs, see Setting Up Logical Inventory, page 2-53.

Defining an MPLS/EMS (EP-LAN) Policy with a CE

This section describes defining a VPLS policy with an MPLS core type and an EMS (EP-LAN) service type with CE present.

Perform the following steps.

Step 1 In the Service Information window of the Policy Editor, choose **VPLS** for the Policy Type.

Step 2 For Core Type, choose **MPLS**.

Step 3 For Service Type, choose **Ethernet Multipoint Service (EMS)**.

Step 4 Check the **CE Present** check box.

Step 5 Click **Next**.

The Interface Type window appears.

Step 6 Choose an **Interface Type** from the drop-down list.

You can choose a particular interface on a CE, N-PE, U-PE, or PE-AGG interface based on the service provider's POP design. The interfaces are:

- **ANY** (Any interface can be chosen.)
- **Port-Channel** (A bundle of ports that share the same characteristics—this gives the service provider the ability to aggregate bandwidth and protection.)
- **Ethernet**
- **FastEthernet**
- **GE-WAN**
- **GigabitEthernet**
- **TenGigabitEthernet**
- **TenGigE**

The value defined here functions as a filter to restrict the interface types an operator can see during VPLS service request creation. If defined as ANY, the operator can see all interface types.

Step 7 Enter an **Interface Format** as the slot number/port number for the CE interface (for example, **1/0** indicates that the interface is located at slot 1, port 0).

This is especially useful to specify here if you know that the link will always go through a particular interface's slot/port location on all or most of the network devices in the service.

Step 8 Choose a CE **Encapsulation** type.

The choices are:

- **DOT1Q**
- **DEFAULT**

Note When creating a service request based on the MPLS/EMS (EP-LAN) with CE policy, the Encapsulation attribute is ignored. Therefore, setting this value has no effect.

Step 9 Check the **Standard UNI Port** check box to enable port security.

This is the default. When you uncheck the check box, the port is treated as an uplink with no security features, and the window dynamically changes to eliminate items related to port security.

Step 10 Check the **UNI Shutdown** check box if you want to leave the UNI port shut during service activation, for example, when the service provider wants to deploy a service in the network but wants to activate it at a later time.

Step 11 Check the **Keep Alive** check box to configure keepalives on the UNI port.

By default, this check box is unchecked, which causes the command **no keepalive** to be provisioned on the UNI port. This prevents a CPE from sending keepalive packets to the U-PE, for security purposes. This attribute is editable to support modification on a per-service request basis.

Step 12 Check the **ANY** check box to display all interface types as choices for the UNI interface (when creating service requests based on this policy). This check box is checked by default.

Step 13 Enter one or more Ethernet MAC addresses in **UNI MAC addresses**.

This selection is present only if you uncheck the **Use Existing ACL Name** check box. Click the **Edit** button to bring up a pop-up window in which you enter MAC addresses to be allowed or denied on the port. You can also specify a range of addresses by setting a base MAC address and a filtered MAC address.

Step 14 Check the **UNI** check box to display all interfaces defined as type UNI as choices for the UNI interface (when creating service requests based on this policy).

This check box is checked by default.

Step 15 Enter a **Link Speed** (optional) of None, 10, 100, 1000, Auto, or nonegotiate.

Step 16 Enter a **Link Duplex** (optional) of None, Full, Half, or Auto.

Step 17 In the **PE/UNI Interface Description** field, enter an optional description, for example *Customer-B EMS (EP-LAN) Service*.

Step 18 Check the **VLAN ID AutoPick** check box if you want Prime Fulfillment to choose a VLAN ID.

If you do not check this check box, you will be prompted to provide the VLAN in a Provider VLAN ID field during service activation.

Step 19 Enter a **VLAN NAME** (optional) to specify a name to describe the VLAN.

The name must be one token (no spaces allowed.) The limit for the VLAN name is 32 characters. The name has to be unique. Two VLANs cannot share the same name.

Step 20 Enter the **System MTU** in bytes.

The maximum transmission unit (MTU) size is configurable and optional. Prime Fulfillment does not perform an integrity check for this customized value. If a service request goes to the **Failed Deploy** state because this size is not accepted, you must adjust the size until the service request is deployed. Prime Fulfillment supports, ranges for different platforms, as specified below. The range is 1500 to 9216.

- For the 3750 and 3550 platforms, the MTU range is 1500-1546.
- For the 7600 ethernet port, the MTU size is always 9216. Even with the same platform and same IOS release, different line cards support the MTU differently. For example, older line cards only take an MTU size of 9216 and newer cards support 1500-9216. However, Prime Fulfillment uses 9216 in both cases.
- For the 7600 SVI (interface VLAN), the MTU size is 1500-9216.

Step 21 Check the **Use Existing ACL Name** check box if you want assign your own named access list to the port.

By default, this check box is not checked and Prime Fulfillment automatically assigns a MAC-based ACL on the customer facing UNI port, based on values you enter in **UNI MAC addresses** (below).

Step 22 Enter a **Port-Based ACL Name** (if you checked the **Use Existing ACL Name** check box, as mentioned in the previous step).

Note Prime Fulfillment does not create this ACL automatically. The ACL must already exist on the device, or be added as part of a template, before the service request is deployed. Otherwise, deployment will fail.

Step 23 Check the **Disable CDP** check box if you want to disable the Cisco Discover Protocol (CDP) on the UNI port.

Step 24 Check the **UNI Port Security** check box if you to want to provision port security-related CLIs to the UNI port by controlling the MAC addresses that are allowed to go through the interface.

a. For **Maximum Number of MAC address**, enter the number of MAC addresses allowed for port security.

b. For **Aging,** enter the length of time the MAC address can stay on the port security table.

c. For **Violation Action**, choose what action will occur when a port security violation is detected:

- **PROTECT**—Drops packets with unknown source addresses until a sufficient number of secure MAC addresses are removed to drop below the maximum value.
- **RESTRICT**—Drops packets with unknown source addresses until a sufficient number of secure MAC addresses are removed to drop below the maximum value and causes the Security Violation counter to increment.
- **SHUTDOWN**—Puts the interface into the error-disabled state immediately and sends an SNMP trap notification.

d. In the **Secure MAC Addresses** field, enter one or more Ethernet MAC addresses.

Step 25 Check the **Enable Storm Control** check box to help prevent the UNI port from being disrupted by a broadcast, multicast, or unicast storm.

Enter a threshold value for each type of traffic. The value, which can be specified to two significant digits, represents the percentage of the total available bandwidth of the port. If the threshold of a traffic type is reached, further traffic of that type is suppressed until the incoming traffic falls below the threshold level.

Step 26 Check the **Protocol Tunnelling** check box if you want to define the Layer 2 Bridge Protocol Data Unit (BPDU) frames that can be tunneled over the core to the other end.

For each protocol that you check, enter the shutdown threshold and drop threshold for that protocol:

a. **Tunnel CDP**—Enable Layer 2 tunnelling on Cisco Discover Protocol (CDP).

b. **CDP Threshold**—Enter the number of packets per second to be received before the interface is shut down.

c. **cdp drop threshold**—Enter the number of packets per second to be received at which point the interface will start dropping CDP packets.

d. **Tunnel VTP**—Enable Layer 2 tunnelling on VLAN Trunk Protocol (VTP).

e. **VTP threshold**—Enter the number of packets per second to be received before the interface is shut down.

f. **vtp drop threshold**—Enter the number of packets per second to be received at which point the interface will start dropping VTP packets.

g. **Tunnel STP**—Enable Layer 2 tunnelling on Spanning Tree Protocol (STP).

h. **STP Threshold**—Enter the number of packets per second to be received before the interface is shut down.

i. **stp drop threshold**—Enter the number of packets per second to be received at which point the interface will start dropping STP packets.

j. **Recovery Interval**—Enter the amount of time, in seconds, to wait before recovering a UNI port.

Step 27 Click the **Next** button, if you want to enable template support for the policy.

The Template Association window appears. In this window, you can enable template support and, optionally, associate templates and data files with the policy. For instructions about associating templates with policies and how to use the features in this window, see Using Templates with Service Requests, page 9-24. When you have completed setting up templates and data files for the policy, click **Finish** in the Template Association window to close it and return to the Policy Editor window.

Note An additional window appears the policy workflow before the Template Association window. This window allows you to create user-defined attributes within the policy (and service requests based on the policy). For background information on how to use the additional information feature, see Appendix H, "Adding Additional Information to Services." If you are not using this feature, click **Next** to proceed to the Template Association window, or else click **Finish** to save the policy.

Step 28 Click **Finish**.

Note The VC ID is mapped from the VPN ID. By default, Prime Fulfillment will "auto pick" this value. However, you can set this manually, if desired. This is done by editing the associated VPN configuration. The Edit VPN window has an **Enable VPLS** check box. When you check this box, you can manually enter a VPN ID in a field provided. For more information on creating and modifying VPNs, seeSetting Up Logical Inventory, page 2-53.

Defining an MPLS/EMS (EP-LAN) Policy without a CE

This section describes defining a VPLS policy with an MPLS core type and an EMS (EP-LAN) service type without a CE present.

Perform the following steps.

Step 1 In the Service Information window of the Policy Editor, choose **VPLS** for the Policy Type.

Step 2 For Core Type, choose **MPLS**.

Step 3 For Service Type, choose **Ethernet Multipoint Service (EMS)**.

Step 4 Uncheck the **CE Present** check box.

Step 5 Click **Next**.

The Interface Type window appears.

Step 6 Choose an **Interface Type** from the drop-down list.

You can choose a particular interface on a N-PE, U-PE, or PE-AGG interface based on the service provider's POP design. The interfaces are:

- **ANY** (Any interface can be chosen.)
- **Port-Channel** (A bundle of ports that share the same characteristics—this gives the service provider the ability to aggregate bandwidth and protection.)
- **Ethernet**
- **FastEthernet**
- **GE-WAN**
- **GigabitEthernet**
- **TenGigabitEthernet**
- **TenGigE**

The value defined here functions as a filter to restrict the interface types an operator can see during VPLS service request creation. If defined as ANY, the operator can see all interface types.

Step 7 Check the **Standard UNI Port** check box to enable port security.

This is the default. When you uncheck the check box, the port is treated as an uplink with no security features, and the window dynamically changes to eliminate items related to port security.

Step 8 Enter an **Interface Format** as the slot number/port number for the PE interface (for example, **1/0** indicates that the interface is located at slot 1, port 0).

This is especially useful to specify here if you know that the link will always go through a particular interface's slot/port location on all or most of the network devices in the service.

Step 9 Choose a N-PE/U-PE **Encapsulation** type.

The choices are:

- **DOT1Q**
- **DEFAULT**

Note When creating a service request based on the MPLS/EMS (EP-LAN) without CE policy, the Encapsulation attribute is ignored. Therefore, setting this value has no effect.

Step 10 Check the **UNI Shutdown** check box if you want to leave the UNI port shut during service activation, for example, when the service provider wants to deploy a service in the network but wants to activate it at a later time.

Step 11 Check the **Keep Alive** check box to configure keepalives on the UNI port.

By default, this check box is unchecked, which causes the command **no keepalive** to be provisioned on the UNI port. This prevents a CPE from sending keepalive packets to the U-PE, for security purposes. This attribute is editable to support modification on a per-service request basis.

Step 12 Check the **ANY** check box to display all interface types as choices for the UNI interface (when creating service requests based on this policy).

This check box is checked by default.

Step 13 Check the **UNI** check box to display all interfaces defined as type UNI as choices for the UNI interface (when creating service requests based on this policy). This check box is checked by default.

Step 14 Enter one or more Ethernet MAC addresses in **UNI MAC addresses**.

This selection is present only if you uncheck the **Use Existing ACL Name** check box. Click the **Edit** button to bring up a pop-up window in which you enter MAC addresses to be allowed or denied on the port. You can also specify a range of addresses by setting a base MAC address and a filtered MAC address.

Step 15 Enter a **Link Speed** (optional) of None, 10, 100, 1000, Auto, or nonegotiate.

Step 16 Enter a **Link Duplex** (optional) of None, Full, Half, or Auto.

Step 17 In the **PE/UNI Interface Description** field, enter an optional description, for example *Customer-B EMS (EP-LAN) Service*.

Step 18 Check the **VLAN ID AutoPick** check box if you want Prime Fulfillment to choose a VLAN ID.

If you do not check this check box, you will be prompted to provide the VLAN in a Provider VLAN ID field during service activation.

Step 19 Enter a **VLAN NAME** (optional) to specify a name to describe the VLAN.

The name must be one token (no spaces allowed.) The limit for the VLAN name is 32 characters. The name has to be unique. Two VLANs cannot share the same name.

Step 20 Enter the **System MTU** in bytes.

The maximum transmission unit (MTU) size is configurable and optional. Prime Fulfillment does not perform an integrity check for this customized value. If a service request goes to the **Failed Deploy** state because this size is not accepted, you must adjust the size until the service request is deployed. Prime Fulfillment supports, ranges for different platforms, as specified below. The range is 1500 to 9216.

- For the 3750 and 3550 platforms, the MTU range is 1500-1546.
- For the 7600 ethernet port, the MTU size is always 9216. Even with the same platform and same IOS release, different line cards support the MTU differently. For example, older line cards only take an MTU size of 9216 and newer cards support 1500-9216. However, Prime Fulfillment uses 9216 in both cases.
- For the 7600 SVI (interface VLAN), the MTU size is 1500-9216.

Step 21 Check the **Use Existing ACL Name** check box if you want assign your own named access list to the port.

By default, this check box is not checked and Prime Fulfillment automatically assigns a MAC-based ACL on the customer facing UNI port, based on values you enter in **UNI MAC addresses** (below).

Step 22 Enter a **Port-Based ACL Name** (if you checked the **Use Existing ACL Name** check box, as mentioned in the previous step).

Note Prime Fulfillment does not create this ACL automatically. The ACL must already exist on the device, or be added as part of a template, before the service request is deployed. Otherwise, deployment will fail.

Step 23 Check the **Disable CDP** check box if you want to disable the Cisco Discover Protocol (CDP) on the UNI port.

Step 24 Check the **UNI Port Security** check box if you to want to provision port security-related CLIs to the UNI port by controlling the MAC addresses that are allowed to go through the interface.

a. For **Maximum Number of MAC address**, enter the number of MAC addresses allowed for port security.

b. For **Aging,** enter the length of time the MAC address can stay on the port security table.

c. For **Violation Action**, choose what action will occur when a port security violation is detected:

- **PROTECT**—Drops packets with unknown source addresses until a sufficient number of secure MAC addresses are removed to drop below the maximum value.
- **RESTRICT**—Drops packets with unknown source addresses until a sufficient number of secure MAC addresses are removed to drop below the maximum value and causes the Security Violation counter to increment.
- **SHUTDOWN**—Puts the interface into the error-disabled state immediately and sends an SNMP trap notification.

d. In the **Secure MAC Addresses** field, enter one or more Ethernet MAC addresses.

Step 25 Check the **Enable Storm Control** check box to help prevent the UNI port from being disrupted by a broadcast, multicast, or unicast storm.

Enter a threshold value for each type of traffic. The value, which can be specified to two significant digits, represents the percentage of the total available bandwidth of the port. If the threshold of a traffic type is reached, further traffic of that type is suppressed until the incoming traffic falls below the threshold level.

Step 26 Check the **Protocol Tunnelling** check box if you want to define the Layer 2 Bridge Protocol Data Unit (BPDU) frames that can be tunneled over the core to the other end.

For each protocol that you check, enter the shutdown threshold and drop threshold for that protocol:

a. **Tunnel CDP**—Enable Layer 2 tunnelling on Cisco Discover Protocol (CDP).

b. **CDP Threshold**—Enter the number of packets per second to be received before the interface is shut down.

c. **cdp drop threshold**—Enter the number of packets per second to be received at which point the interface will start dropping CDP packets.

d. **Tunnel VTP**—Enable Layer 2 tunnelling on VLAN Trunk Protocol (VTP).

e. **VTP threshold**—Enter the number of packets per second to be received before the interface is shut down.

f. **vtp drop threshold**—Enter the number of packets per second to be received at which point the interface will start dropping VTP packets.

g. **Tunnel STP**—Enable Layer 2 tunnelling on Spanning Tree Protocol (STP).

h. **STP Threshold**—Enter the number of packets per second to be received before the interface is shut down.

i. **stp drop threshold**—Enter the number of packets per second to be received at which point the interface will start dropping STP packets.

j. **Recovery Interval**—Enter the amount of time, in seconds, to wait before recovering a UNI port.

Step 27 Click the **Next** button, if you want to enable template support for the policy.

The Template Association window appears. In this window, you can enable template support and, optionally, associate templates and data files with the policy. For instructions about associating templates with policies and how to use the features in this window, see Using Templates with Service Requests, page 9-24. When you have completed setting up templates and data files for the policy, click **Finish** in the Template Association window to close it and return to the Policy Editor window.

Note An additional window appears the policy workflow before the Template Association window. This window allows you to create user-defined attributes within the policy (and service requests based on the policy). For background information on how to use the additional information feature, see Appendix H, "Adding Additional Information to Services." If you are not using this feature, click **Next** to proceed to the Template Association window, or else click **Finish** to save the policy.

Step 28 Click **Finish**.

Note The VC ID is mapped from the VPN ID. By default, Prime Fulfillment will "auto pick" this value. However, you can set this manually, if desired. This is done by editing the associated VPN configuration. The Edit VPN window has an **Enable VPLS** check box. When you check this box, you can manually enter a VPN ID in a field provided. For more information on creating and modifying VPNs, see Setting Up Logical Inventory, page 2-53.

Defining an Ethernet/ERMS (EVP-LAN) Policy with a CE

This section describes defining a VPLS policy with an Ethernet core type and an ERMS (EVP-LAN) service type with CE present.

Perform the following steps.

Step 1 In the Service Information window of the Policy Editor, choose **VPLS** for the Policy Type.

Step 2 For Core Type, choose **Ethernet**.

Step 3 For Service Type, choose **Ethernet Relay Multipoint Service (ERMS)**.

Step 4 Check the **CE Present** check box.

Step 5 Click **Next**.

The Interface Type window appears.

Step 6 Choose an **Interface Type** from the drop-down list.

You can choose a particular interface on a CE, N-PE, U-PE, or PE-AGG interface based on the service provider's POP design. The interfaces are:

- **ANY** (Any interface can be chosen.)
- **Port-Channel** (A bundle of ports that share the same characteristics—this gives the service provider the ability to aggregate bandwidth and protection.)
- **Ethernet**
- **FastEthernet**
- **GE-WAN**

- **GigabitEthernet**
- **TenGigabitEthernet**
- **TenGigE**

The value defined here functions as a filter to restrict the interface types an operator can see during VPLS service request creation. If defined as ANY, the operator can see all interface types.

Step 7 Enter an **Interface Format** as the slot number/port number for the CE interface (for example, **1/0** indicates that the interface is located at slot 1, port 0).

This is especially useful to specify here if you know that the link will always go through a particular interface's slot/port location on all or most of the network devices in the service.

Step 8 Choose a CE **Encapsulation** type.

The choices are:

- **DOT1Q**
- **DEFAULT**

If **DEFAULT** is the CE encapsulation type, Prime Fulfillment shows another field for the UNI port type.

Step 9 Check the **Standard UNI Port** check box to enable port security.

This is the default. When you uncheck the check box, the port is treated as an uplink with no security features, and the window dynamically changes to eliminate items related to port security.

Step 10 Check the **UNI Shutdown** check box if you want to leave the UNI port shut during service activation, for example, when the service provider wants to deploy a service in the network but wants to activate it at a later time.

Step 11 Check the **Keep Alive** check box to configure keepalives on the UNI port.

By default, this check box is unchecked, which causes the command **no keepalive** to be provisioned on the UNI port. This prevents a CPE from sending keepalive packets to the U-PE, for security purposes. This attribute is editable to support modification on a per-service request basis.

Step 12 Check the **ANY** check box to display all interface types as choices for the UNI interface (when creating service requests based on this policy).

This check box is checked by default.

Step 13 Check the **UNI** check box to display all interfaces defined as type UNI as choices for the UNI interface (when creating service requests based on this policy).

This check box is checked by default.

Step 14 Enter one or more Ethernet MAC addresses in **UNI MAC addresses**.

This selection is present only if you uncheck the **Use Existing ACL Name** check box. Click the **Edit** button to bring up a pop-up window in which you enter MAC addresses to be allowed or denied on the port. You can also specify a range of addresses by setting a base MAC address and a filtered MAC address.

Step 15 Choose a **Port Type**.

The choices are:

- **Access Port**
- **Trunk with Native VLAN**

Step 16 Enter a **Link Speed** (optional) of None, 10, 100, 1000, Auto, or nonegotiate.

Step 17 Enter a **Link Duplex** (optional) of None, Full, Half, or Auto.

Step 18 In the **PE/UNI Interface Description** field, enter an optional description, for example *Customer-B ERMS (EVP-LAN) Service*.

Step 19 Check the **VLAN ID AutoPick** check box if you want Prime Fulfillment to choose a VLAN ID.

If you do not check this check box, you will be prompted to provide the VLAN in a Provider VLAN ID field during service activation.

Step 20 Enter a **VLAN NAME** (optional) to specify a name to describe the VLAN.

The name must be one token (no spaces allowed.) The limit for the VLAN name is 32 characters. The name has to be unique. Two VLANs cannot share the same name.

Step 21 Check the **Use Existing ACL Name** check box if you want assign your own named access list to the port.

By default, this check box is not checked and Prime Fulfillment automatically assigns a MAC-based ACL on the customer facing UNI port, based on values you enter in **UNI MAC addresses** (below).

Step 22 Enter a **Port-Based ACL Name** (if you checked the **Use Existing ACL Name** check box, as mentioned in the previous step).

Note Prime Fulfillment does not create this ACL automatically. The ACL must already exist on the device, or be added as part of a template, before the service request is deployed. Otherwise, deployment will fail.

Step 23 Check the **Disable CDP** check box if you want to disable the Cisco Discover Protocol (CDP) on the UNI port.

Step 24 Check the **Filter BPDU** check box to specify that the UNI port should not process Layer 2 Bridge Protocol Data Units (BPDUs).

Step 25 Check the **UNI Port Security** check box if you to want to provision port security-related CLIs to the UNI port by controlling the MAC addresses that are allowed to go through the interface.

a. For **Maximum Number of MAC address**, enter the number of MAC addresses allowed for port security.

b. For **Aging,** enter the length of time the MAC address can stay on the port security table.

c. For **Violation Action**, choose what action will occur when a port security violation is detected:

- **PROTECT**—Drops packets with unknown source addresses until a sufficient number of secure MAC addresses are removed to drop below the maximum value.
- **RESTRICT**—Drops packets with unknown source addresses until a sufficient number of secure MAC addresses are removed to drop below the maximum value and causes the Security Violation counter to increment.
- **SHUTDOWN**—Puts the interface into the error-disabled state immediately and sends an SNMP trap notification.

d. In the **Secure MAC Addresses** field, enter one or more Ethernet MAC addresses.

Step 26 Check the **Enable Storm Control** check box to help prevent the UNI port from being disrupted by a broadcast, multicast, or unicast storm.

Enter a threshold value for each type of traffic. The value, which can be specified to two significant digits, represents the percentage of the total available bandwidth of the port. If the threshold of a traffic type is reached, further traffic of that type is suppressed until the incoming traffic falls below the threshold level.

Step 27 Click the **Next** button, if you want to enable template support for the policy.

The Template Association window appears. In this window, you can enable template support and, optionally, associate templates and data files with the policy. For instructions about associating templates with policies and how to use the features in this window, see Using Templates with Service Requests, page 9-24. When you have completed setting up templates and data files for the policy, click **Finish** in the Template Association window to close it and return to the Policy Editor window.

Note An additional window appears the policy workflow before the Template Association window. This window allows you to create user-defined attributes within the policy (and service requests based on the policy). For background information on how to use the additional information feature, see Appendix H, "Adding Additional Information to Services." If you are not using this feature, click **Next** to proceed to the Template Association window, or else click **Finish** to save the policy.

Step 28 Click **Finish**.

Note The VC ID is mapped from the VPN ID. By default, Prime Fulfillment will "auto pick" this value. However, you can set this manually, if desired. This is done by editing the associated VPN configuration. The Edit VPN window has an **Enable VPLS** check box. When you check this box, you can manually enter a VPN ID in a field provided. For more information on creating and modifying VPNs, see Setting Up Logical Inventory, page 2-53.

Defining an Ethernet/ERMS (EVP-LAN) Policy without a CE

This section describes defining a VPLS policy with an Ethernet core type and an ERMS (EVP-LAN) service type without a CE present.

Perform the following steps.

Step 1 In the Service Information window of the Policy Editor, choose **VPLS** for the Policy Type.

Step 2 For Core Type, choose **Ethernet**.

Step 3 For Service Type, choose **Ethernet Relay Multipoint Service (ERMS)**.

Step 4 Uncheck the **CE Present** check box.

Step 5 Click **Next**.

The Interface Type window appears.

Step 6 Choose an **Interface Type** from the drop-down list.

You can choose a particular interface on a CE, N-PE, U-PE, or PE-AGG interface based on the service provider's POP design. The interfaces are:

- **ANY** (Any interface can be chosen.)
- **Port-Channel** (A bundle of ports that share the same characteristics—this gives the service provider the ability to aggregate bandwidth and protection.)
- **Ethernet**
- **FastEthernet**
- **GE-WAN**

- **GigabitEthernet**
- **TenGigabitEthernet**
- **TenGigE**

The value defined here functions as a filter to restrict the interface types an operator can see during VPLS service request creation. If defined as ANY, the operator can see all interface types.

Step 7 Check the **Standard UNI Port** check box to enable port security.

This is the default. When you uncheck the check box, the port is treated as an uplink with no security features, and the window dynamically changes to eliminate items related to port security.

Step 8 Enter an **Interface Format** as the slot number/port number for the CE interface (for example, **1/0** indicates that the interface is located at slot 1, port 0).

This is especially useful to specify here if you know that the link will always go through a particular interface's slot/port location on all or most of the network devices in the service.

Step 9 Choose a CE **Encapsulation** type.

The choices are:

- **DOT1Q**
- **DEFAULT**

If **DEFAULT** is the CE encapsulation type, Prime Fulfillment shows another field for the UNI port type.

Step 10 Check the **UNI Shutdown** check box if you want to leave the UNI port shut during service activation, for example, when the service provider wants to deploy a service in the network but wants to activate it at a later time.

Step 11 Check the **Keep Alive** check box to configure keepalives on the UNI port.

By default, this check box is unchecked, which causes the command **no keepalive** to be provisioned on the UNI port. This prevents a CPE from sending keepalive packets to the U-PE, for security purposes. This attribute is editable to support modification on a per-service request basis.

Step 12 Check the **ANY** check box to display all interface types as choices for the UNI interface (when creating service requests based on this policy).

This check box is checked by default.

Step 13 Check the **UNI** check box to display all interfaces defined as type UNI as choices for the UNI interface (when creating service requests based on this policy). This check box is checked by default.

Step 14 Enter one or more Ethernet MAC addresses in **UNI MAC addresses**.

This selection is present only if you uncheck the **Use Existing ACL Name** check box. Click the **Edit** button to bring up a pop-up window in which you enter MAC addresses to be allowed or denied on the port. You can also specify a range of addresses by setting a base MAC address and a filtered MAC address.

Step 15 Choose a **Port Type**.

The choices are:

- **Access Port**
- **Trunk with Native VLAN**

Step 16 Enter a **Link Speed** (optional) of None, 10, 100, 1000, Auto, or nonegotiate.

Step 17 Enter a **Link Duplex** (optional) of None, Full, Half, or Auto.

Step 18 In the **PE/UNI Interface Description** field, enter an optional description, for example *Customer-B ERMS (EVP-LAN) Service*.

Step 19 Check the **VLAN ID AutoPick** check box if you want Prime Fulfillment to choose a VLAN ID.

If you do not check this check box, you will be prompted to provide the VLAN in a Provider VLAN ID field during service activation.

Step 20 Enter a **VLAN NAME** (optional) to specify a name to describe the VLAN.

The name must be one token (no spaces allowed.) The limit for the VLAN name is 32 characters. The name has to be unique. Two VLANs cannot share the same name.

Step 21 Check the **Use Existing ACL Name** check box if you want assign your own named access list to the port.

By default, this check box is not checked and Prime Fulfillment automatically assigns a MAC-based ACL on the customer facing UNI port, based on values you enter in **UNI MAC addresses** (below).

Step 22 Enter a **Port-Based ACL Name** (if you checked the **Use Existing ACL Name** check box, as mentioned in the previous step).

Note Prime Fulfillment does not create this ACL automatically. The ACL must already exist on the device, or be added as part of a template, before the service request is deployed. Otherwise, deployment will fail.

Step 23 Check the **Disable CDP** check box if you want to disable the Cisco Discover Protocol (CDP) on the UNI port.

Step 24 Check the **Filter BPDU** check box to specify that the UNI port should not process Layer 2 Bridge Protocol Data Units (BPDUs).

Step 25 Check the **UNI Port Security** check box if you to want to provision port security-related CLIs to the UNI port by controlling the MAC addresses that are allowed to go through the interface.

a. For **Maximum Number of MAC address**, enter the number of MAC addresses allowed for port security.

b. For **Aging,** enter the length of time the MAC address can stay on the port security table.

c. For **Violation Action**, choose what action will occur when a port security violation is detected:

- **PROTECT**—Drops packets with unknown source addresses until a sufficient number of secure MAC addresses are removed to drop below the maximum value.
- **RESTRICT**—Drops packets with unknown source addresses until a sufficient number of secure MAC addresses are removed to drop below the maximum value and causes the Security Violation counter to increment.
- **SHUTDOWN**—Puts the interface into the error-disabled state immediately and sends an SNMP trap notification.

d. In the **Secure MAC Addresses** field, enter one or more Ethernet MAC addresses.

Step 26 Check the **Enable Storm Control** check box to help prevent the UNI port from being disrupted by a broadcast, multicast, or unicast storm.

Enter a threshold value for each type of traffic. The value, which can be specified to two significant digits, represents the percentage of the total available bandwidth of the port. If the threshold of a traffic type is reached, further traffic of that type is suppressed until the incoming traffic falls below the threshold level.

Step 27 Click the **Next** button, if you want to enable template support for the policy.

The Template Association window appears. In this window, you can enable template support and, optionally, associate templates and data files with the policy. For instructions about associating templates with policies and how to use the features in this window, see Using Templates with Service Requests, page 9-24. When you have completed setting up templates and data files for the policy, click **Finish** in the Template Association window to close it and return to the Policy Editor window.

Note An additional window appears the policy workflow before the Template Association window. This window allows you to create user-defined attributes within the policy (and service requests based on the policy). For background information on how to use the additional information feature, see Appendix H, "Adding Additional Information to Services." If you are not using this feature, click **Next** to proceed to the Template Association window, or else click **Finish** to save the policy.

Step 28 Click **Finish**.

Note The VC ID is mapped from the VPN ID. By default, Prime Fulfillment will "auto pick" this value. However, you can set this manually, if desired. This is done by editing the associated VPN configuration. The Edit VPN window has an **Enable VPLS** check box. When you check this box, you can manually enter a VPN ID in a field provided. For more information on creating and modifying VPNs, see Setting Up Logical Inventory, page 2-53.

Defining an Ethernet/EMS (EP-LAN) Policy with a CE

This section describes defining a VPLS policy with an Ethernet core type and an EMS (EP-LAN) service type with a CE present.

Perform the following steps.

Step 1 In the Service Information window of the Policy Editor, choose **VPLS** for the Policy Type.

Step 2 For Core Type, choose **Ethernet**.

Step 3 For Service Type, choose **Ethernet Multipoint Service (EMS)**.

Step 4 Check the **CE Present** check box.

Step 5 Click **Next**.

The Interface Type window appears.

Step 6 Choose an **Interface Type** from the drop-down list.

You can choose a particular interface on a CE, N-PE, U-PE, or PE-AGG interface based on the service provider's POP design.

The interfaces are:

- **ANY** (Any interface can be chosen.)
- **Port-Channel** (A bundle of ports that share the same characteristics—this gives the service provider the ability to aggregate bandwidth and protection.)
- **Ethernet**
- **FastEthernet**

- **GE-WAN**
- **GigabitEthernet**
- **TenGigabitEthernet**

The value defined here functions as a filter to restrict the interface types an operator can see during VPLS service request creation. If defined as ANY, the operator can see all interface types.

Step 7 Enter an **Interface Format** as the slot number/port number for the CE interface (for example, **1/0** indicates that the interface is located at slot 1, port 0).

This is especially useful to specify here if you know that the link will always go through a particular interface's slot/port location on all or most of the network devices in the service.

Step 8 Choose a CE **Encapsulation** type.

The choices are:

- **DOT1Q**
- **DEFAULT**

Note When creating a service request based on the Ethernet/EMS (EP-LAN) with CE policy, the Encapsulation attribute is ignored. Therefore, setting this value has no effect.

Step 9 Check the **Standard UNI Port** check box to enable port security.

This is the default. When you uncheck the check box, the port is treated as an uplink with no security features, and the window dynamically changes to eliminate items related to port security.

Step 10 Check the **UNI Shutdown** check box if you want to leave the UNI port shut during service activation, for example, when the service provider wants to deploy a service in the network but wants to activate it at a later time.

Step 11 Check the **Keep Alive** check box to configure keepalives on the UNI port.

By default, this check box is unchecked, which causes the command **no keepalive** to be provisioned on the UNI port. This prevents a CPE from sending keepalive packets to the U-PE, for security purposes. This attribute is editable to support modification on a per-service request basis.

Step 12 Check the **ANY** check box to display all interface types as choices for the UNI interface (when creating service requests based on this policy).

This check box is checked by default.

Step 13 Check the **UNI** check box to display all interfaces defined as type UNI as choices for the UNI interface (when creating service requests based on this policy).

This check box is checked by default.

Step 14 Enter one or more Ethernet MAC addresses in **UNI MAC addresses**.

This selection is present only if you uncheck the **Use Existing ACL Name** check box. Click the **Edit** button to bring up a pop-up window in which you enter MAC addresses to be allowed or denied on the port. You can also specify a range of addresses by setting a base MAC address and a filtered MAC address.

Step 15 Enter a **Link Speed** (optional) of None, 10, 100, 1000, Auto, or nonegotiate.

Step 16 Enter a **Link Duplex** (optional) of None, Full, Half, or Auto.

Step 17 In the **PE/UNI Interface Description** field, enter an optional description, for example *Customer-B EMS (EP-LAN) Service*.

Step 18 Check the **VLAN ID AutoPick** check box if you want Prime Fulfillment to choose a VLAN ID.

If you do not check this check box, you will be prompted to provide the VLAN in a Provider VLAN ID field during service activation.

Step 19 Enter a **VLAN NAME** (optional) to specify a name to describe the VLAN.

The name must be one token (no spaces allowed.) The limit for the VLAN name is 32 characters. The name has to be unique. Two VLANs cannot share the same name.

Step 20 Enter the **System MTU** in bytes.

The maximum transmission unit (MTU) size is configurable and optional. The default size is 9216, and the range is 1500 to 9216. Prime Fulfillment does not perform an integrity check for this customized value. If a service request goes to the Failed Deploy state because this size is not accepted, you must adjust the size until the Service Request is deployed.

In Cisco Prime Fulfillment 1.0, different platforms support different ranges.

- For the 3750 and 3550 platforms, the MTU range is 1500-1546.
- For the 7600 ethernet port, the MTU size is always 9216. Even with the same platform and same IOS release, different line cards support the MTU differently. For example, older line cards only take an MTU size of 9216 and newer cards support 1500-9216. However, Cisco Prime Fulfillment 1.0 uses 9216 in both cases.
- For the 7600 SVI (interface VLAN), the MTU size is 1500-9216.

Step 21 Check the **Use Existing ACL Name** check box if you want assign your own named access list to the port.

By default, this is not checked and Prime Fulfillment automatically assigns a MAC-based ACL on the customer facing UNI port, based on values you enter in **UNI MAC addresses** (below).

Step 22 Enter a **Port-Based ACL Name** (if you checked the **Use Existing ACL Name** check box, as mentioned in the previous step).

Note Prime Fulfillment does not create this ACL automatically. The ACL must already exist on the device, or be added as part of a template, before the service request is deployed. Otherwise, deployment will fail.

Step 23 Check the **Disable CDP** check box if you want to disable the Cisco Discover Protocol (CDP) on the UNI port.

Step 24 Check the **UNI Port Security** check box if you to want to provision port security-related CLIs to the UNI port by controlling the MAC addresses that are allowed to go through the interface.

a. For **Maximum Number of MAC address**, enter the number of MAC addresses allowed for port security.

b. For **Aging,** enter the length of time the MAC address can stay on the port security table.

c. For **Violation Action**, choose what action will occur when a port security violation is detected:

- **PROTECT**—Drops packets with unknown source addresses until a sufficient number of secure MAC addresses are removed to drop below the maximum value.
- **RESTRICT**—Drops packets with unknown source addresses until a sufficient number of secure MAC addresses are removed to drop below the maximum value and causes the Security Violation counter to increment.
- **SHUTDOWN**—Puts the interface into the error-disabled state immediately and sends an SNMP trap notification.

d. In the **Secure MAC Addresses** field, enter one or more Ethernet MAC addresses.

Step 25 Check the **Enable Storm Control** check box to help prevent the UNI port from being disrupted by a broadcast, multicast, or unicast storm.

Enter a threshold value for each type of traffic. The value, which can be specified to two significant digits, represents the percentage of the total available bandwidth of the port. If the threshold of a traffic type is reached, further traffic of that type is suppressed until the incoming traffic falls below the threshold level.

Step 26 Check the **Protocol Tunnelling** check box if you want to define the Layer 2 Bridge Protocol Data Unit (BPDU) frames that can be tunneled over the core to the other end.

For each protocol that you check, enter the shutdown threshold and drop threshold for that protocol:

a. **Tunnel CDP**—Enable Layer 2 tunnelling on Cisco Discover Protocol (CDP).

b. **CDP Threshold**—Enter the number of packets per second to be received before the interface is shut down.

c. **cdp drop threshold**—Enter the number of packets per second to be received at which point the interface will start dropping CDP packets.

d. **Tunnel VTP**—Enable Layer 2 tunnelling on VLAN Trunk Protocol (VTP).

e. **VTP threshold**—Enter the number of packets per second to be received before the interface is shut down.

f. **vtp drop threshold**—Enter the number of packets per second to be received at which point the interface will start dropping VTP packets.

g. **Tunnel STP**—Enable Layer 2 tunnelling on Spanning Tree Protocol (STP).

h. **STP Threshold**—Enter the number of packets per second to be received before the interface is shut down.

i. **stp drop threshold**—Enter the number of packets per second to be received at which point the interface will start dropping STP packets.

j. **Recovery Interval**—Enter the amount of time, in seconds, to wait before recovering a UNI port.

Step 27 Click the **Next** button, if you want to enable template support for the policy.

The Template Association window appears. In this window, you can enable template support and, optionally, associate templates and data files with the policy. For instructions about associating templates with policies and how to use the features in this window, see Using Templates with Service Requests, page 9-24. When you have completed setting up templates and data files for the policy, click **Finish** in the Template Association window to close it and return to the Policy Editor window.

Note An additional window appears the policy workflow before the Template Association window. This window allows you to create user-defined attributes within the policy (and service requests based on the policy). For background information on how to use the additional information feature, see Appendix H, "Adding Additional Information to Services." If you are not using this feature, click **Next** to proceed to the Template Association window, or else click **Finish** to save the policy.

Step 28 Click **Finish**.

Note The VC ID is mapped from the VPN ID. By default, Prime Fulfillment will "auto pick" this value. However, you can set this manually, if desired. This is done by editing the associated VPN configuration. The Edit VPN window has an **Enable VPLS** check box. When you check this box, you can manually enter a VPN ID in a field provided. For more information on creating and modifying VPNs, see Setting Up Logical Inventory, page 2-53.

Defining an Ethernet/EMS (EP-LAN) Policy without a CE

This section describes defining a VPLS policy with an Ethernet core type and an EMS (EP-LAN) service type without a CE present. Perform the following steps.

Step 1 In the Service Information window of the Policy Editor, choose **VPLS** for the Policy Type.

Step 2 For Core Type, choose **Ethernet**.

Step 3 For Service Type, choose **Ethernet Multipoint Service (EMS)**.

Step 4 Uncheck the **CE Present** check box.

Step 5 Click **Next**.

The Interface Type window appears.

Step 6 Choose an **Interface Type** from the drop-down list.

You can choose a particular interface on a CE, N-PE, U-PE, or PE-AGG interface based on the service provider's POP design. The interfaces are:

- **ANY** (Any interface can be chosen.)
- **Port-Channel** (A bundle of ports that share the same characteristics—this gives the service provider the ability to aggregate bandwidth and protection.)
- **Ethernet**
- **FastEthernet**
- **GE-WAN**
- **GigabitEthernet**
- **TenGigabitEthernet**
- **TenGigE**

The value defined here functions as a filter to restrict the interface types an operator can see during VPLS service request creation. If defined as ANY, the operator can see all interface types.

Step 7 Check the **Standard UNI Port** check box to enable port security.

This is the default. When you uncheck the check box, the port is treated as an uplink with no security features, and the window dynamically changes to eliminate items related to port security.

Step 8 Enter an **Interface Format** as the slot number/port number for the CE interface (for example, **1/0** indicates that the interface is located at slot 1, port 0).

This is especially useful to specify here if you know that the link will always go through a particular interface's slot/port location on all or most of the network devices in the service.

Step 9 Choose a N-PE/U-PE **Encapsulation** type.

The choices are:

- **DOT1Q**
- **DEFAULT**

Note When creating a service request based on the Ethernet/EMS (EP-LAN) without CE policy, the Encapsulation attribute is ignored. Therefore, setting this value has no effect.

Step 10 Check the **UNI Shutdown** check box if you want to leave the UNI port shut during service activation, for example, when the service provider wants to deploy a service in the network but wants to activate it at a later time.

Step 11 Check the **Keep Alive** check box to configure keepalives on the UNI port.

By default, this check box is unchecked, which causes the command **no keepalive** to be provisioned on the UNI port. This prevents a CPE from sending keepalive packets to the U-PE, for security purposes. This attribute is editable to support modification on a per-service request basis.

Step 12 Check the **ANY** check box to display all interface types as choices for the UNI interface (when creating service requests based on this policy).

This check box is checked by default.

Step 13 Check the **UNI** check box to display all interfaces defined as type UNI as choices for the UNI interface (when creating service requests based on this policy).

This check box is checked by default.

Step 14 Enter one or more Ethernet MAC addresses in **UNI MAC addresses**.

This selection is present only if you uncheck the **Use Existing ACL Name** check box. Click the **Edit** button to bring up a pop-up window in which you enter MAC addresses to be allowed or denied on the port. You can also specify a range of addresses by setting a base MAC address and a filtered MAC address.

Step 15 Enter a **Link Speed** (optional) of None, 10, 100, 1000, Auto, or nonegotiate.

Step 16 Enter a **Link Duplex** (optional) of None, Full, Half, or Auto.

Step 17 In the **PE/UNI Interface Description** field, enter an optional description, for example *Customer-B EMS (EP-LAN) Service*.

Step 18 Check the **VLAN ID AutoPick** check box if you want Prime Fulfillment to choose a VLAN ID. If you do not check this check box, you will be prompted to provide the VLAN in a Provider VLAN ID field during service activation.

Step 19 Enter a **VLAN NAME** (optional) to specify a name to describe the VLAN.

The name must be one token (no spaces allowed.) The limit for the VLAN name is 32 characters. The name has to be unique. Two VLANs cannot share the same name.

Step 20 Enter the **System MTU** in bytes.

The maximum transmission unit (MTU) size is configurable and optional. Prime Fulfillment does not perform an integrity check for this customized value. If a service request goes to the **Failed Deploy** state because this size is not accepted, you must adjust the size until the service request is deployed. Prime Fulfillment supports, ranges for different platforms, as specified below. The range is 1500 to 9216.

- For the 3750 and 3550 platforms, the MTU range is 1500-1546.

- For the 7600 ethernet port, the MTU size is always 9216. Even with the same platform and same IOS release, different line cards support the MTU differently. For example, older line cards only take an MTU size of 9216 and newer cards support 1500-9216. However, Prime Fulfillment uses 9216 in both cases.
- For the 7600 SVI (interface VLAN), the MTU size is 1500-9216.

Step 21 Check the **Use Existing ACL Name** check box if you want assign your own named access list to the port.

By default, this check box is not checked and Prime Fulfillment automatically assigns a MAC-based ACL on the customer facing UNI port, based on values you enter in **UNI MAC addresses** (below).

Step 22 Enter a **Port-Based ACL Name** (if you checked the **Use Existing ACL Name** check box, as mentioned in the previous step).

Note Prime Fulfillment does not create this ACL automatically. The ACL must already exist on the device, or be added as part of a template, before the service request is deployed. Otherwise, deployment will fail.

Step 23 Check the **Disable CDP** check box if you want to disable the Cisco Discover Protocol (CDP) on the UNI port.

Step 24 Check the **UNI Port Security** check box if you to want to provision port security-related CLIs to the UNI port by controlling the MAC addresses that are allowed to go through the interface.

a. For **Maximum Number of MAC address**, enter the number of MAC addresses allowed for port security.

b. For **Aging,** enter the length of time the MAC address can stay on the port security table.

c. For **Violation Action**, choose what action will occur when a port security violation is detected:

- **PROTECT**—Drops packets with unknown source addresses until a sufficient number of secure MAC addresses are removed to drop below the maximum value.
- **RESTRICT**—Drops packets with unknown source addresses until a sufficient number of secure MAC addresses are removed to drop below the maximum value and causes the Security Violation counter to increment.
- **SHUTDOWN**—Puts the interface into the error-disabled state immediately and sends an SNMP trap notification.

d. In the **Secure MAC Addresses** field, enter one or more Ethernet MAC addresses.

Step 25 Check the **Enable Storm Control** check box to help prevent the UNI port from being disrupted by a broadcast, multicast, or unicast storm.

Enter a threshold value for each type of traffic. The value, which can be specified to two significant digits, represents the percentage of the total available bandwidth of the port. If the threshold of a traffic type is reached, further traffic of that type is suppressed until the incoming traffic falls below the threshold level.

Step 26 Check the **Protocol Tunnelling** check box if you want to define the Layer 2 Bridge Protocol Data Unit (BPDU) frames that can be tunneled over the core to the other end.

For each protocol that you check, enter the shutdown threshold and drop threshold for that protocol:

a. **Tunnel CDP**—Enable Layer 2 tunnelling on Cisco Discover Protocol (CDP).

b. **CDP Threshold—**Enter the number of packets per second to be received before the interface is shut down.

c. **cdp drop threshold**—Enter the number of packets per second to be received at which point the interface will start dropping CDP packets.

d. **Tunnel VTP**—Enable Layer 2 tunnelling on VLAN Trunk Protocol (VTP).

e. **VTP threshold**—Enter the number of packets per second to be received before the interface is shut down.

f. **vtp drop threshold**—Enter the number of packets per second to be received at which point the interface will start dropping VTP packets.

g. **Tunnel STP—**Enable Layer 2 tunnelling on Spanning Tree Protocol (STP).

h. **STP Threshold—**Enter the number of packets per second to be received before the interface is shut down.

i. **stp drop threshold**—Enter the number of packets per second to be received at which point the interface will start dropping STP packets.

j. **Recovery Interval**—Enter the amount of time, in seconds, to wait before recovering a UNI port.

Step 27 Click the **Next** button, if you want to enable template support for the policy.

The Template Association window appears. In this window, you can enable template support and, optionally, associate templates and data files with the policy. For instructions about associating templates with policies and how to use the features in this window, see Using Templates with Service Requests, page 9-24. When you have completed setting up templates and data files for the policy, click **Finish** in the Template Association window to close it and return to the Policy Editor window.

Note An additional window appears the policy workflow before the Template Association window. This window allows you to create user-defined attributes within the policy (and service requests based on the policy). For background information on how to use the additional information feature, see Appendix H, "Adding Additional Information to Services." If you are not using this feature, click **Next** to proceed to the Template Association window, or else click **Finish** to save the policy.

Step 28 Click **Finish**.

Note The VC ID is mapped from the VPN ID. By default, Prime Fulfillment will "auto pick" this value. However, you can set this manually, if desired. This is done by editing the associated VPN configuration. The Edit VPN window has an **Enable VPLS** check box. When you check this box, you can manually enter a VPN ID in a field provided. For more information on creating and modifying VPNs, see Setting Up Logical Inventory, page 2-53.

Managing a VPLS Service Request

This section contains the basic steps to provision a VPLS service. It contains the following subsections:

- Modifying the VPLS Service Request, page 3-163
- Using the Bridge Domain ID Attribute, page 3-165
- Saving the EVC Service Request, page 3-54

Introducing VPLS Service Requests

A VPLS service request consists of one or more attachment circuits, connecting various sites in a multipoint topology. When you create a service request, you enter several parameters, including the specific interfaces on the CE and PE routers and UNI parameters.

You can also associate Prime Fulfillment templates and data files with a service request. See Chapter 9, "Managing Templates and Data Files" for more about using templates and data files in service requests.

It is also possible to create user-defined attributes within a policy (and service requests based on the policy). For background information on how to use the additional information feature, see Appendix H, "Adding Additional Information to Services."

To create a service request, a service policy must already be defined, as described in Creating a VPLS Policy, page 3-130. Based on the predefined VPLS policy, an operator creates a VPLS service request, with or without modifications to the VPLS policy, and deploys the service. The service request must be the same service type (ERMS/EVP-LAN or EMS/EP-LAN) as the policy selected. Service creation and deployment are normally performed by regular network technicians for daily operation of network provisioning.

The following steps are involved in creating a service request for Layer 2 connectivity between customer sites:

- Choose a VPLS policy.
- Choose a VPN. For more information, see Defining VPNs, page 3-8.
- Add a link.
- Choose a CE or UNI interface.
- Choose a Named Physical Circuit (NPC) if more than one NPC exists from the CE or the UNI interface.
- Edit the link attributes.

For sample configlets for VPLS scenarios, see Sample Configlets, page 3-176.

Creating a VPLS Service Request

To create a VPLS service request, perform the following steps.

Step 1 Choose **Operate > Service Requests > Service Request Manager**.

The Service Requests Manager window appears.

Step 2 Click **Create**.

The Service Request Editor window appears.

Step 3 From the Policy drop-down list, choose a VPLS policy from the policies previously created (see Creating a VPLS Policy, page 3-130).

The L2VPN Service Request editor window appears.

The new service request inherits all the properties of that VPLS policy, such as all the editable and noneditable features and preset parameters.

Step 4 To continue creating a VPLS service request, go to one of the following sections:

- Creating an ERS (EVPL), ATM, or Frame Relay L2VPN Service Request with a CE, page 3-120.
- Creating an ERS (EVPL), ATM, or Frame Relay L2VPN Service Request without a CE, page 3-124.

Creating a VPLS Service Request with a CE

This section includes detailed steps for creating a VPLS service request with a CE present. In this example, the service request is for an VPLS policy over an MPLS core with an ERMS (EVP-LAN) service type and CE present.

Perform the following steps.

Step 1 Choose the appropriate VPLS policy.

The Edit VPLS Link window appears.

Step 2 Click **Select VPN** to choose a VPN for use with this CE.

The Select VPN window appears with the VPNs defined in the system. Only VPNs with the same service type (ERMS/EVP-LAN or EMS/EP-LAN) as the policy you chose appear.

Note The VC ID is mapped from the VPN ID. By default, Prime Fulfillment will "auto pick" this value. However, you can set this manually, if desired. This is done by editing the associated VPN configuration. The Edit VPN window has an **Enable VPLS** check box. When you check this check box, you can manually enter a VPN ID in a field provided. For more information on creating and modifying VPNs, see Setting Up Logical Inventory, page 2-53.

Step 3 Choose a **VPN Name** in the Select column.

Step 4 Click **Select**.

The Edit VPLS Link window appears with the VPN name displayed.

Step 5 Click **Add Link**.

The window updates, allowing you specify the CE endpoints.

Step 6 You can enter a description for the service request in the **Description** field.

The description will show up in this window and also in the Description column of the VPLS Service Requests window. The maximum length for this field is 256 characters.

Step 7 Click **Select CE** in the CE column.

The Select CPE Device window appears.

This window displays the list of currently defined CEs.

a. From the **Show CPEs with** drop-down list, you can display CEs by Customer Name, by Site, or by Device Name.

b. You can use the **Find** button to either search for a specific CE, or to refresh the display.

c. You can set the **Rows per page** to 5, 10, 20, 30, 40, or All.

Step 8 In the Select column, choose a CE for the VPLS link.

Step 9 Click **Select**.

The Edit VPLS Link window appears displaying the name of the selected CE in the CE column.

Step 10 Choose the CE interface from the drop-down list.

Note When you provision an ERMS (EVP-LAN) service (and when you choose a UNI for a particular device), Prime Fulfillment determines if there are other services using the same UNI. If so, a warning message is displayed. If you ignore the message and save the service request, all of the underlying service requests lying on the same UNI are synchronized with the modified shared attributes of the latest service request. In addition, the state of the existing service requests is changed to the Requested state.

Step 11 Click **Select one circuit** in the Circuit Selection column.

The Select NPC window appears. If only one NPC exists for the chosen CE and CE interface, that NPC is automatically populated in the Circuit Selection column and you need not choose it explicitly.

Step 12 Choose the name of the NPC from the Select column.

Step 13 Click **OK**.

Each time you choose a CE and its interface, the NPC that was precreated from this CE and interface is automatically displayed under **Circuit Selection**. This means that you do not have to further specify the PE to complete the link.

Step 14 If you want to review the details of this NPC, click **Circuit Details** in the Circuit Details column.

The NPC Details window appears and lists the circuit details for this NPC.

Step 15 The Circuit ID is created automatically, based on the VLAN data for the circuit.

Step 16 To edit values that were set by the VPLS policy, that is, the values that were marked “editable” during the VPLS policy creation, click the **Edit** link in the Link Attributes column for a link.

The Edit VPLS window appears.

Note For more information on setting attributes in this window, see Modifying the EVC Service Request, page 3-53.

Note For information on the Bridge Domain ID attribute, which shows up in some VPLS service request scenarios, see Modifying the VPLS Service Request, page 3-163.

Step 17 Continue to specify additional CEs, as in previous steps, if desired.

Step 18 Click **OK**.

Step 19 Click **Save**.

The service request is created and saved into Prime Fulfillment.

Creating a VPLS Service Request without a CE

This section includes detailed steps for creating a VPLS service request without a CE present. In this example, the service request is for an VPLS policy over an MPLS core with an EMS (EP-LAN) service type and no CE present.

Perform the following steps.

Step 1 Choose the appropriate VPLS policy.

The Edit VPLS Link window appears.

Step 2 Click **Select VPN** to choose a VPN for use with this PE.

The Select VPN window appears with the VPNs defined in the system. Only VPNs with the same service type (ERMS/EVP-LAN or EMS/EP-LAN) as the policy you chose appear.

Note The VC ID is mapped from the VPN ID. By default, Prime Fulfillment will "auto pick" this value. However, you can set this manually, if desired. This is done by editing the associated VPN configuration. The Edit VPN window has an **Enable VPLS** check box. When you check this check box, you can manually enter a VPN ID in a field provided. For more information on creating and modifying VPNs, see Setting Up Logical Inventory, page 2-53.

Step 3 Choose a **VPN Name** in the Select column.

Step 4 Click **Select**.

The Edit VPLS Link window appears with the VPN name displayed.

Step 5 Click **Add Link**.

The Edit VPLS Link window updates, allowing you specify the U-PE/PE-AGG/U-PE endpoints. You can add one or more links in the window.

Step 6 You can enter a description for the service request in the first **Description** field.

The description will show up in this window and also in the Description column of the VPLS Service Requests window. The maximum length for this field is 256 characters.

Step 7 Click **Select N-PE/PE-AGG/U-PE** in the N-PE/PE-AGG/U-PE column.

The Select PE Device window appears.

This window displays the list of currently defined PEs.

a. The **Show PEs with** drop-down list shows PEs by customer name, by site, or by device name.

b. The **Find** button allows a search for a specific PE or a refresh of the window.

c. The **Rows per page** drop-down list allows the page to be set to 5, 10, 20, 30, 40, or All.

Step 8 In the **Select** column, choose the PE device name for the VPLS link.

Step 9 Click **Select**.

The Edit VPLS Link window appears displaying the name of the selected N-PE/PE-AGG/U-PE in the N-PE/PE-AGG/U-PE column

Step 10 Choose the UNI interface from the drop-down list.

Note When you provision an ERMS (EVP-LAN) service (and when you choose a UNI for a particular device), Prime Fulfillment determines if there are other services using the same UNI. If so, a warning message is displayed. If you ignore the message and save the service request, all of the underlying service requests lying on the same UNI are synchronized with the modified shared attributes of the latest service request. In addition, the state of the existing service requests is changed to the Requested state.

Step 11 If the PE role type is U-PE, click **Select one circuit** in the Circuit Selection column.

The Select NPC window appears. If only one NPC exists for the chosen PE and PE interface, that NPC is automatically populated in the Circuit Selection column and you need not choose it explicitly.

Note If the PE role type is N-PE, the columns Circuit Selection and Circuit Details are disabled.

Step 12 Choose the name of the NPC from the **Select** column.

Step 13 Click **OK**.

Each time you choose a PE and its interface, the NPC that was precreated from this PE and interface is automatically displayed under **Circuit Selection**. This means that you do not have to further specify the PE to complete the link.

Step 14 If you want to review the details of this NPC, click **Circuit Details** in the Circuit Details column.

The NPC Details window appears and lists the circuit details for this NPC.

The Circuit ID is created automatically, based on the VLAN data for the circuit.

Step 15 To edit values that were set by the VPLS policy, that is, the values that were marked “editable” during the VPLS policy creation, click the **Edit** link in the Link Attributes column for a link.

Note For more information on setting attributes in this window, see Modifying the EVC Service Request, page 3-53.

Note For information on the Bridge Domain ID attribute, which shows up in some VPLS service request scenarios, see Modifying the VPLS Service Request, page 3-163.

Step 16 Continue to specify additional PEs, as in previous steps, if desired.

Step 17 Click **Save**.

The VPLS service request is created and saved into Prime Fulfillment.

Modifying the VPLS Service Request

You can modify a VPLS service request if you must change or modify the VPLS links. This is also where you can associate templates and data files to a link.

Perform the following steps.

Step 1 Choose **Operate > Service Requests > Service Request Manager**.

Step 2 Check a check box for a service request.

Step 3 Click **Edit**.

The Edit VPLS Link window appears.

Step 4 Specify items in the window as necessary for your configuration:

- Choose any of the blue highlighted values to edit the VPLS links.
- Click **Add Link** to add a VPLS link.
- Click **Delete Link** to delete a VPLS link.

Note If you are attempting to decommission a service request to which a template has been added, see Decommissioning Service Requests, page 8-12, for information on the proper way to do this.

- You can enter a description for the service request in the first **Description** field. The description will show up in this window and also in the Description column of the Service Requests window. The maximum length for this field is 256 characters.
- The Circuit ID is created automatically, based on the VLAN data for the circuit.

Step 5 To modify the link attributes, click **Edit** in the Link Attributes column as shown in the VPLS link editor.

The Edit VPLS window appears.

Step 6 Edit the link attributes as desired.

Note If you did not choose **VLANI D AutoPick** in the VPLS policy, you are prompted to provide the VLAN in a **Provider VLAN ID** field.

Note For information on the Bridge Domain ID attribute, which shows up in some VPLS service request scenarios, see Modifying the VPLS Service Request, page 3-163.

Step 7 To add a template and data file to a link, choose a Device Name, and click the **Add** link in the Templates column.

The Add/Remove Templates window appears.

Note To add a template to a link, you must have already created the template. For detailed steps to create templates, see Overview, page 9-1. For more information on how to use templates and data files in service requests, see Chapter 9, “Managing Templates and Data Files.”

Step 8 Click **Add**.

The Template Data File Chooser window appears.

Step 9 In the left pane, navigate to and select a template.

The associated data files are listed in rows in the main window.

Step 10 Check the data file that you want to add and click **Accept**.

The Add/Remove Templates window appears with the template displayed.

Step 11 Choose a Template name.

Step 12 Under Action, use the drop-down list and choose **APPEND** or **PREPEND**.

Append tells Prime Fulfillment to append the template generated CLI to the regular Prime Fulfillment (non-template) CLI. Prepend is the reverse and does not append the template to the Prime Fulfillment CLI.

Step 13 Choose Active to use this template for this service request.

If you do not choose Active, the template is not used.

Step 14 Click **OK**.

The Edit VPLS window appears with the template added.

Step 15 Click **OK**.

The Edit VPLS Link window appears.

Step 16 When you are finished editing the VPLS links, click **Save**.

Using the Bridge Domain ID Attribute

The Bridge Domain ID attribute appears in the Link Attributes window of some VPLS service request scenarios.

To use the Bridge Domain ID attribute, enter an ID number in the **Bridge Domain ID** text field to enable bridge domain functionality for the VPLS service request.

Acceptable values are 1 to 4294967295.

Usage notes:

- The Bridge Domain ID attribute is only available for the following service request scenarios:
  - Ethernet/ERMS (EVP-LAN) with a CE
  - Ethernet/ERMS (EVP-LAN) without a CE
  - Ethernet/EMS (EP-LAN) with a CE
  - Ethernet/EMS (EP-LAN) without a CE
- The Bridge Domain ID attribute is only supported for the Cisco GSR 12406 running IOS 12.0(32)SY6 and functioning in an N-PE role. This attribute will show up in a service request only for this platform; otherwise, the attribute will be filtered from the Link Attributes window of the service request.
- The following points apply to service requests based on this policy:
  - When an N-PE (GSR platform) is used as a UNI device, the standard UNI attributes are not displayed in the Link Attributes window of the service request workflow.
  - When a U-PE (non-GSR platform) is used as a UNI device, all standard UNI attributes are displayed in the Link Attributes window of the service request workflow.
  - For VPLS EMS services, a U-PE (non-GSR platform) should be used in the same circuit which is terminating on a GSR device (N-PE). In other words, an NPC circuit should be used to provision VPLS EMS on GSR devices.

Saving the VPLS Service Request

To save a VPLS service request, perform the following steps.

Step 1 When you are finished setting all the attributes for the attachment circuits, click **Save** to finish the VPLS service request creation.

If the VPLS service request is successfully created, you will see a list of service request s in the Service Request Manager window. The newly created VPLS service request is added with the state of REQUESTED.

Step 2 If, however, the VPLS service request creation failed for some reason (for example, a value chosen is out of bounds), you are warned with an error message.

In such a case, you should correct the error and save the service request again.

Deploying, Monitoring, and Auditing Service Requests

To apply L2VPN, VPLS, or EVC policies to network devices, you must deploy the service request. When you deploy a service request, Prime Fulfillment compares the device information in the Repository (the Prime Fulfillment database) with the current device configuration and generates a configlet. Additionally, you can perform various monitoring and auditing tasks on service requests. These common task that apply to all types of Prime Fulfillment service requests are covered in Chapter 8, "Managing Service Requests." See that chapter for more information on these tasks.

This section covers specific issues related to managing service request tasks for EVC, L2VPN and VPLS services.

Pre-Deployment Changes

You can change the Dynamic Component Properties Library (DCPL) parameter **actionTakenOnUNIVlanList** before you deploy an EVC, L2VPN, or VPLS service request. This will be necessary if the **trunk allowed vlan** list is not present on the User Network Interface (UNI).

To make this change, perform the following steps.

Step 1 Choose **Administration > Control Center > Hosts**.

Step 2 Choose the host that you want to change.

Step 3 Click **Config**.

The Host Configuration window appears.

Step 4 In the DCPL properties panel, choose **Provisioning > Service > shared > actionTakenOnUNIVlanList**.

The Attribute details appear.

Step 5 In the **New Value** drop-down list, choose one of the following:

- **prune** to have Prime Fulfillment create the minimum VLAN list. This is the default.
- **abort** to have Prime Fulfillment stop the L2VPN or VPLS service request provisioning with the error message: **trunk allowed vlan list is absent on ERS UNI**.
- **nochange** to have Prime Fulfillment allow all VLANs.

Step 6 Click **Set Property**.

Using Autodiscovery for L2 Services

All discovery steps are integrated in a discovery workflow, controlled from the Prime Fulfillment GUI. This is accessed in Prime Fulfillment through **Inventory > Physical Inventory > Discovery**. The following discovery features are supported:

- File-based device discovery is supported.
- Rules-based device role assignment is supported.
- Discovery progress messages and logs are viewable in the GUI to keep track of various discovery stages.
- Bulk creation of Provider, Customer, Site, and Region objects is available through an XML data file.

For detailed steps on using the autodiscovery feature in Prime Fulfillment, see Appendix G, "Inventory - Discovery."

Provisioning VPLS Autodiscovery on Devices using EVC Service Requests

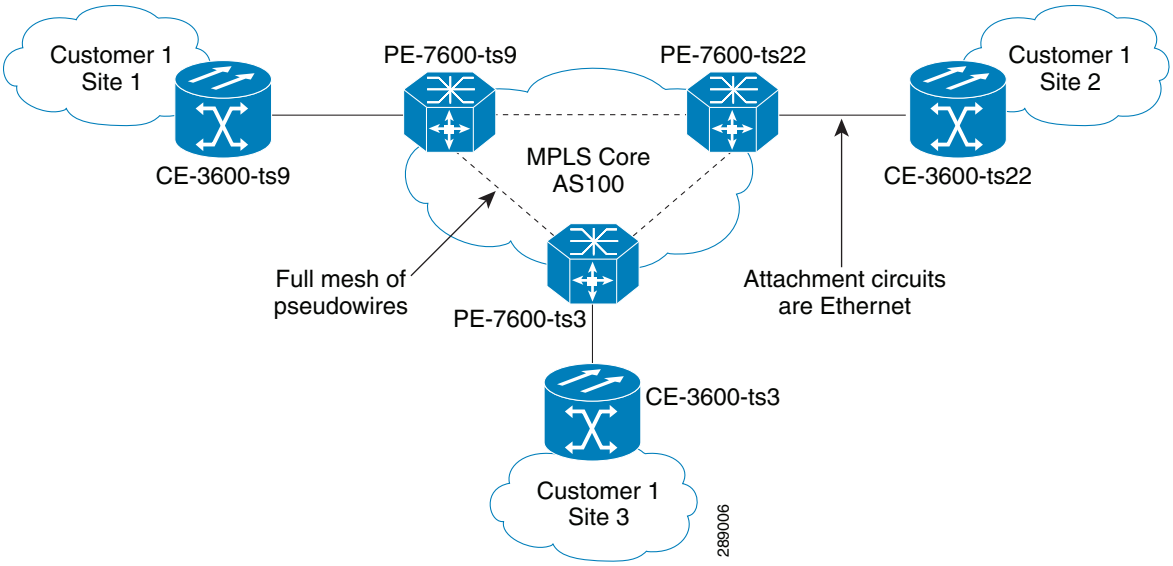
This section describes how enable the VPLS autodiscovery in Prime Fulfillment. It contains the following sections:

Overview

Earlier implementations of VPLS in IOS and IOS XR required manual configuration of each VPLS PE neighbor when devices were added or removed from the VPLS domain. VPLS auto discovery eliminates the need to manually configure the VPLS neighbors. It discovers PEs within the same VPLS domain and automatically detects when PEs are added or removed from the domain.

Figure 3-1 shows an example VPLS topology that will be referenced in this section. The three PE devices constitute the neighbors in the VPLS domain. As PEs are added or removed from the domain, VPLS autodiscovery keeps the PE configurations updated.

Figure 3-1 VPLS Autodiscovery Topology Example



To provision VPLS autodiscovery on PE devices in the VPLS domain, you must perform two basic tasks:

- You must preconfigure some configlets on the devices before they are provisioned by Prime Fulfillment. You must do this manually or through the use of templates. See Preconfiguring PE Devices to Support VPLS Autodiscovery, page 3-169.
- You must enable VPLS autodiscovery within the EVC service request(s) used to provision the PE(s) in the VPLS domain.

The rest of this section documents limitations and restrictions of VPLS autodiscovery, describes the steps you must perform in the workflow to enable it, and provides sample configlets generated on IOS and IOS XR devices.

Limitations and Restrictions for VPLS Autodiscovery

Keep in mind the following limitations and restrictions when using VPLS autodiscovery Prime Fulfillment.

- To use VPLS autodiscovery, all PE devices in the VPLS domain must be have VPLS autodiscovery enabled. Mixed topologies (that is, some PEs configured with VPLS autodiscovery enabled and some without) are not supported. The VPLS discovery mode should be enabled for all service requests under the same virtual forwarding interface (VFI).
- Some preconfiguration on the PEs in the VPLS domain is required. See Preconfiguring PE Devices to Support VPLS Autodiscovery, page 3-169.
- Split horizon should be enabled for when using VPLS autodiscovery.
- VPLS autodiscovery can only be configured in Prime Fulfillment using EVC Ethernet service requests for which the MPLS Core Connectivity Type is set as VPLS. The feature is not supported for other Prime Fulfillment service requests and/or connectivity types.
- The same discovery mechanism must be used to build a pseudowire between two PE peers. It is not valid for both auto discovered and manually configured pseudowires in the same VFI to go to the same peer PE. For example, it is not valid for PE1 to be manually configured for PE2 and PE2 be dynamically configured to discover PE1.

- Once the VPLS discovery mode is provisioned (as manual or autodiscovery) in the service required, it cannot be modified.
- VPLS autodiscovery is only supported for full-mesh topologies, not hub and spoke topologies like hierarchical VPLS (H-VPLS).
- VPLS autodiscovery is not supported with inter-autonomous system configurations.

Preconfiguring PE Devices to Support VPLS Autodiscovery

The following configlets must be preconfigured on IOS and IOS XR devices before provisioning VPLS autodiscovery on them. The configlets are required to setup MP-iBGP peering with other PEs and to enable VPLS L2VPN community information exchange with other PEs in the same VPLS domain.

```
!  Setup MP-iBGP peering with other PEs  !
router bgp 100
 no bgp default ipv4-unicast
 bgp log-neighbor-changes
 neighbor 193.193.20.3 remote-as 100
 neighbor 193.193.20.3 update-source Loopback0
 neighbor 193.193.20.5 remote-as 100
 neighbor 193.193.20.5 update-source Loopback0

 ! Enable VPLS l2vpn community info exchange with other PEs in the same VPLS domain  !
address-family l2vpn vpls
 neighbor 193.193.20.3 activate
 neighbor 193.193.20.3 send-community extended
 neighbor 193.193.20.5 activate
 neighbor 193.193.20.5 send-community extended
 exit-address-family
      !
```

Enabling VPLS Autodiscovery in the EVC Workflow

To enable VPLS discovery in the EVC Ethernet workflow, perform the following steps.

Step 1 In the EVC Ethernet policy or service request workflow, set the **MPLS Core Connectivity Type** to **VPLS**.

When the core connectivity is VPLS, the Discovery Mode attribute dynamically appears in the Service Request Details section of the EVC Service Request Editor window. This window describes the VPLS connectivity between the attachment circuits. VPLS connectivity allows the creation of a multipoint connection between two customer sites, using direct connect links or L2 access links.

Step 2 Choose the **Discovery Mode** type in the EVC Service Request Editor window.

The choices are:

- **Manual**— When the Manual option is selected, the **vfi** command will be configured as in legacy with the **manual** option. This is the same for both IOS and IOS XR devices. The signaling protocol implemented is LDP.
- **Auto Discovery**— When the Auto Discovery option is selected, the **vfi** command will be configured with the **autodiscovery** option, and the **neighbor** command is not required.

For examples of the resulting configlets generated by these choices, see Sample Configlets, page 3-170.

Step 3 Save the service request and deploy it on the device(s) in the VPLS domain.

Sample Configlets

This section provides sample configlets generated by Prime Fulfillment for both IOS and IOS XR devices for VPLS autodiscovery.

Sample Configlet for IOS Device

```
! Setup VPLS intstance,!
l2 vfi customer1 autodiscovery
 vpn id 100

! Set attachment circuit interface in VLAN mode  !
interface FastEthernet4/1
 description VPN for CE9-3640-ts22
 switchport
 switchport access vlan 100
 switchport mode access
 no cdp enable

! Bind VLAN100(AC) to the customer1 pseudowire !
interface Vlan100
 no ip address
 xconnect vfi customer1
```

Sample Configlet for IOS XR Device

```
    l2vpn
    bridge group abc
      bridge-domain east
       vfi vfiname
        vpn-id 678
        autodiscovery bgp
         rd auto
         route-target 456:567
```

Note For IOS XR devices, the Route Target value must be saved while creating the VPN.

Setting Up VLAN Translation for L2VPN ERS (EVPL) Services

This section provide supplemental information about how to set up VLAN translation for L2VPN ERS (EVPL) services. It contains the following subsections:

Note For helpful information to be aware of before you create policies and services using VLAN translation, review Platform-Specific Usage Notes, page 3-175.

VLAN Translation Overview

VLAN translation provides flexibility in managing VLANs and Metro Ethernet-related services. There are two types of VLAN translation—one is 1-to-1 translation (1:1), and the other one is 2-to-1 translation (2:1). This feature is available for L2VPN ERS (EVPL) (with and without a CE). The behavior of L2VPN ERS (EVPL) service remains the same, even though it is true that it is possible now for one Q-in-Q port to be shared by both EWS (EPL) and ERS (EVPL) service. VLAN translation is only for an Ethernet interface, not for other types of interfaces, such as ATM and Frame Relay.

With 1:1 VLAN translation, the VLAN of the incoming traffic (CE VLAN) is replaced by another VLAN (PE VLAN). It means the service provider is now able to handle the situation where incoming traffic from two different customers share the same CE VLAN. The SP can map these two CE VLANs to two different PE VLANs, and customer traffic will not be mixed.

With 2:1 VLAN translation, the double tagged (Q-in-Q) traffic at the U-PE UNI port can be mapped to different flows to achieve service multiplexing. The translation is based on the combination of the CE VLAN (inner tag) and the PE VLAN (outer tag). Without this translation, all the traffic from a Q-in-Q port can only go to one place because it is switched only by the outer tag.

Setting Up VLAN Translation

The following sections described how to create and manage policies and service requests to support VLAN translation:

Creating a Policy

VLAN translation is specified during policy creation for L2VPN for ERS (EVPL) (with and without a CE). The L2VPN (Point to Point) Editor window contains a new option called **VLAN Translation**.

There are three options for VLAN translation:

- **No**—This is the default choice. No VLAN translation is performed.

Note If you choose **No** and you do not want to deal with any behavior related to VLAN translation during service request creation, then uncheck the **Editable** check box. This is the recommendation when you choose no VLAN translation.

- **1:1**—1:1 VLAN translation. The VLAN of the incoming traffic (CE VLAN) is replaced by another VLAN (PE VLAN). The specification of the VLAN translation is done during the creation of the service request for the policy, as covered in Creating a Service Request, page 3-172.
- **2:1**—2:1 VLAN translation. The double tagged (Q-in-Q) traffic at the U-PE UNI port can be mapped to different flows to achieve service multiplexing. When you choose 2:1 VLAN translation, the L2VPN (Point to Point) Editor window dynamically changes to enable you to choose where the 2:1 VLAN translation takes place.

The choices for where 2:1 VLAN translation takes place are:

- **Auto** (This is the default choice.)
- **U-PE**
- **PE-AGG**
- **N-PE**

If you choose **Auto**, the 2:1 VLAN translation takes place at the device closest to the UNI port. The other choices come into play only when there is more than one place that 2:1 VLAN translation can be done. If there is only one place where the translation can be done, the choice is ignored.

The actual VLAN values are specified when you create a service request based on this policy. See Creating a Service Request, page 3-172.

Creating a Service Request

When you create a service request based on an L2VPN ERS (EVPL) policy, the VLAN options can be changed if they were set to be editable in the policy. You can overwrite the policy information for the VLAN translation type and the place where translation occurs. This flexibility allows the following provisioning:

- One AC can have 2:1 VLAN translation, while the other AC can have no VLAN translation or 1:1 VLAN translation.
- The VLAN translation for one AC can be on the UNI box, while the translation for the other AC can be on the PE-AGG.

Note Note these modifications can happen only when a new service request is created. They are not allowed during the modification of an existing service request.

The specification of the VLAN translation happens during the creation of the service request within the Link Attributes window. At that point, you can specify which VLAN is translated to which VLAN. The Link Attributes window is accessed after the UNI port is selected on the Attachment Tunnel Editor window. Because you can set the VLAN translation type after the UNI selection, the UNI port display list does not exclude any type for the UNI port. This is because:

- The UNI port list has to include the regular trunk port, in case you later (on the Link Attributes window) decide to perform no VLAN translation or 1:1 VLAN translation.
- The UNI port list has to include an EWS (EPL) (Q-in-Q) port, in case you decide to do 2:1 VLAN translation.

Even though you have all the ports to start with for VLAN translation, you must choose specific types of ports, based on the type of VLAN translation. More specifically:

- For no VLAN translation and 1:1 VLAN translation, you must choose an empty port or a trunk port as the UNI.
- For 2:1 VLAN translation, you must choose an empty port or a Q-in-Q port as the UNI port.

To help determine the proper port to use, you can click the **Details** button on the Attachment Tunnel Editor window to display the port type and associated service with that port.

The following sections show how the VLAN translation is defined on the Link Attribute window for the different types of VLAN translation.

No VLAN Translation

When you choose no VLAN translation, no additional information needs to be provided.

1:1 VLAN Translation

When you choose 1:1 VLAN translation, the window dynamically changes.

In the empty field, you must enter which CE VLAN is to be translated from. The VLAN number must be a number from 1 to 4096.

The PE VLAN that the CE VLAN is to be translated to can be "auto picked" or manually entered. Check the **VLAN ID AutoPick** check box above (on the Link Attributes window) to have PE VLAN automatically assigned.

If you uncheck the **VLAN ID AutoPick** check box, the window displays a Provider VLAN ID, where you can manually enter the PE VLAN.

Upon completion of the service request creation, Prime Fulfillment does an integrity check before saving the service request. For 1:1 VLAN translation, Prime Fulfillment rejects the service request if the CE VLAN has been used for another 1:1 VLAN translation on the same port.

2:1 VLAN Translation

When choosing 2:1 VLAN translation, the window dynamically changes.

Note If the UNI port has been provisioned with EWS (EPL) service, the outer VLAN value is grayed out.

In 2:1 VLAN translation, there are three VLANs involved:

- "A"—The CE VLAN to be translated from. You specify this in the "From CE VLAN field." For out-of-range translation, a value of "*" (asterisk character) should be provided
- "B"—The PE VLAN that is the outer VLAN of the Q-in-Q port. You specify this in the "Outer VLAN" field. You can choose this VLAN manually by entering a value, or you can choose the **AutoPick** check box to have one automatically assigned.
- "C"—The PE VLAN that the "A" and "B" VLANs are translated to. You specify this in the "VLAN and Other Information" section above (on the Link Attributes window).

You must specify VLAN "A" (the CE VLAN) and VLAN "C" (the PE VLAN translated to). For VLAN "B" (the Q-in-Q outer VLAN), what to specify depends on the UNI port type:

- If it is an empty port, you must specify VLAN "B."

- If it is an existing Q-in-Q port, then VLAN “B” has been defined, and it cannot be changed at this point.

Some additional comments on 2:1 VLAN translation:

- For 2:1 VLAN translation, if you build an ERS (EVPL) service on an empty port, then this UNI port will be provisioned as an ERS (EVPL) service. If you later add an EWS (EPL) service to the same port, the EWS (EPL) service will overwrite the previous ERS (EVPL) provisioning. The major difference between ERS (EVPL) and EWS (EPL) is the L2PT BPDU treatment. For ERS (EVPL), BPDU is blocked. For EWS (EPL), BPDU is tunneled.
- As an ERS (EVPL) service, the 2:1 VLAN translation can share the same port, just like a regular ERS (EVPL) port.
- An ERS (EVPL) 2:1 service can be added on top of an existing EWS (EPL) service.

Upon completion of the service request creation, Prime Fulfillment does an integrity check before saving the service request. For 2:1 VLAN translation, Prime Fulfillment rejects the service request if the CE VLAN and outer tag PE VLAN combination has been used for another 2:1 VLAN translation on the same port.

Modifying a Service Request

For both 1:1 and 2:1 VLAN translation, you can perform the following modifications on an existing service request:

- Change to a new CE VLAN to be translated from.
- All other normal changes for a service request are permitted.

However, the following modifications are not allowed:

- You cannot change the VLAN translation type for a given AC. For instance, you cannot change from 2:1 to 1:1 VLAN translation.
- You cannot change the place where 2:1 VLAN translation occurs.

Deleting a Service Request

During service request deletion, the following resources are released:

For 1:1 VLAN translation:

- The CE VLAN becomes available to be translated again.
- The PE VLAN is released.
- If the link being deleted is the last link on the UNI port, then this port is set to new.

For 2:1 VLAN translation:

- The CE VLAN becomes available to be translated again.
- The “translated to” PE VLAN is released.
- If the link being deleted is the last “CE-PE” pair on this UNI port, and there is no EWS (EPL) service on this port, then this port is set to new. In addition, the outer VLAN is released.

Platform-Specific Usage Notes

VLAN translation is available on 7600 and 3750 ME platforms. The 7600 and 3750 ME have different ways to support VLAN translation. Not only is the command syntax different, but so is the place where the VLAN translation is carried out. On the 7600, for 1:1 VLAN translation, the operation is done on the PFC card. For 2:1 VLAN translation, the operation is done on the uplink GE-WAN (OSM module). On the 3750 ME, however, both translations occur on the uplinks (ES ports).

VLAN Translation on the 3750

Be aware of the following points when performing VLAN translation on the 3750.

- The 3750 where VLAN translation occurs should be designated as a U-PE or PE-AGG role, not N-PE.
- VLAN translation on the up link (ES) port should be performed on the Gigabit 1/1/1 or Gigabit 1/1/2 port.
- If a 1:1 VLAN translation occurs on a ring that is made of 3750 PEs, all the 3750s use the ES port as uplink ports (the “east” and “west” ports) to connect other ring nodes.

VLAN Translation on the 7600

Be aware of the following points when performing VLAN translation on the 7600.

- 1:1 VLAN translation always occurs on the UNI port. However, not every Ethernet interface will support 1:1 VLAN translation. Such support is dependent on the line card.
- 2:1 VLAN translation always occurs on the GE-WAN port. The port must be an NNI uplink port.
- 2:1 VLAN translation only occurs on a 7600 that is a U-PE or a PE-AGG, not an N-PE. The reason is when the 2:1 VLAN translation is performed on the GE-WAN interface, this interface can no longer perform L3VPN and L2VPN service using the translated new VLAN. The L3/L2VPN service has to be provisioned on another (N-PE) box.

Failed Service Requests When Hardware Does Not Support VLAN Translation

For the 1:1 VLAN translation feature, a service request goes to the **Fail Deployed** state if the target hardware (line card) does not support the VLAN translation. The reason the service request goes to the **Fail Deployed** state instead of **Invalid** is that Prime Fulfillment does not know beforehand whether a particular line card will accept or reject the VLAN translation CLI commands. In this case, Prime Fulfillment attempts to push down the commands and the deployment fails. An **Invalid** status means Prime Fulfillment detects something wrong (in advance) and aborts the provisioning task. No CLI is pushed down in that case. This is a general behavior of Prime Fulfillment when a given hardware does not support a feature. In these cases, it is the user’s responsibility to select proper hardware to support the intended service.

Sample Configlets

This section provides sample configlets for L2VPN and Metro Ethernet service provisioning in Prime Fulfillment. It contains the following subsections:

Overview

The configlets provided in this section show the CLIs generated by Prime Fulfillment for particular services and features. Each configlet example provides the following information:

- Service
- Feature
- Devices configuration (network role, hardware platform, relationship of the devices and other relevant information)
- Sample configlets for each device in the configuration
- Comments

Note The configlets generated by Prime Fulfillment are only the delta between what needs to be provisioned and what currently exists on the device. This means that if a relevant CLI is already on the device, it does not show up in the associated configlet.

Note The CLIs shown in bold are the most relevant commands.

Note All examples in this section assume an MPLS core.

ERS (EVPL) (Point-to-Point)

Configuration

- Service: L2VPN/Metro Ethernet.
- Feature: ERS (EVPL) (point-to-point).
- Device configuration:
  - The N-PE is a Cisco 7600 with IOS 12.2(18)SXF, Sup720-3BXL.

    Interface(s): FA8/17.
  - The U-PE is a Cisco 3750ME with 12.2(25)EY1, no port security.

    Interface(s): FA1/0/4 – FA1/0/23.
  - L2VPN point-to-point.

Configlets

U-PE	N-PE
vlan 772 exit ! interface FastEthernet1/0/23 switchport trunk allowed vlan 500,772 ! interface FastEthernet1/0/4 no cdp enable no keepalive no ip address **switchport trunk allowed vlan 500,772** **spanning-tree bpdufilter enable** **mac access-group ISC-FastEthernet1/0/4 in** **!** **mac access-list extended** **ISC-FastEthernet1/0/4** **deny any host 0100.0ccc.cccc** **deny any host 0100.0ccc.cccd** **deny any host 0100.0ccd.cdd0** **deny any host 0180.c200.0000** **permit any any**	vlan 772 exit ! interface FastEthernet8/17 switchport trunk allowed vlan 1,451,653,659,766-768,772,878 ! **interface Vlan772** **no ip address** **description L2VPN ERS** **xconnect 99.99.8.99 89027 encapsulation** **mpls** **no shutdown**

Comments

- The N-PE is a 7600 with an OSM or SIP-600 module.
- The U-PE is a generic Metro Ethernet (ME) switch. Customer BPDUs are blocked by the PACL.

ERS (EVPL) (Point-to-Point, UNI Port Security)

Configuration

- Service: L2VPN/Metro Ethernet.
- Feature: ERS (EVPL) (point-to-point) with UNI port security.
- Device configuration:
  - The N-PE is a Cisco 7600 with IOS 12.2(18)SXF, OSM.

    Interface(s): FA2/18.
  - The U-PE is a Cisco 3550 with IOS 12.2(25)SEC2. Port security is enabled.

    Interface(s): FA3/31– FA3/23.
  - L2VPN point-to-point.

Configlets

U-PE

```
vlan 788
exit
!
interface FastEthernet3/23
no ip address
switchport trunk allowed vlan 783,787-788
!
interface FastEthernet3/31
no cdp enable
no keepalive
no ip address
switchport
switchport trunk encapsulation dot1q
switchport mode trunk
switchport trunk allowed vlan none
switchport trunk allowed vlan 788
switchport port-security
switchport nonegotiate
switchport port-security maximum 45
switchport port-security aging time 34
switchport port-security violation shutdown
switchport port-security mac-address
3456.3456.5678
spanning-tree bpdufilter enable
mac access-group ISC-FastEthernet3/31 in
!
mac access-list extended
ISC-FastEthernet3/31
deny any host 0100.0ccc.cccc
deny any host 0100.0ccc.cccd
deny any host 0100.0ccd.cdd0
deny any host 0180.c200.0000
deny any host 1234.3234.3432
permit any any
```

N-PE

```
vlan 788
exit
!
interface FastEthernet2/18
switchport trunk allowed vlan
350,351,430,630,777,780,783,785-788
!
interface Vlan788
no ip address
description L2VPN ERS with UNI port
security
xconnect 99.99.5.99 89028 encapsulation
mpls
no shutdown
```

Comments

- The N-PE is a 7600 with an OSM or SIP-600 module.
- The U-PE is a generic Metro Ethernet (ME) switch. The customer BPDUs are blocked by the PACL.
- Various UNI port security commands are provisioned.

- A user-defined PACL entry is added to the default PACL.

ERS (EVPL) (1:1 VLAN Translation)

Configuration

- Service: L2VPN/Metro Ethernet.
- Feature: ERS (EVPL) with 1:1 VLAN translation.
- Device configuration:
  - The N-PE is a Cisco 7600 with IOS 12.2(18)SXF, Sup720-3BXL

    Interface(s): FA8/34.
  - The U-PE is a Cisco 3750ME with IOS 12.2(25)EY1. VLAN translation on the NNI port (uplink).

    Interface(s): FA1/0/8 – GI1/1/1.
  - L2VPN point-to-point.

Configlets

U-PE

```
!
vlan 123
exit
!
interface FastEthernet1/0/8
no cdp enable
no keepalive
no ip address
switchport trunk allowed vlan 123
switchport nonegotiate
switchport port-security maximum 34
switchport port-security aging time 23
switchport port-security violation protect
switchport port-security
spanning-tree bpdufilter enable
mac access-group ISC-FastEthernet1/0/8 in
!
interface GigabitEthernet1/1/1
no ip address
switchport mode trunk
switchport trunk allowed vlan 1,123
switchport vlan mapping 123 778
```

N-PE

```
vlan 778
exit
!
interface FastEthernet8/34
switchport
switchport trunk encapsulation dot1q
switchport mode trunk
switchport trunk allowed vlan 1,778
!
interface Vlan778
no ip address
description L2VPN ERS 1 to 1 vlan
translation
xconnect 99.99.8.99 89032 encapsulation
mpls
no shutdown
```

Comments

- VLAN translation is only for L2VPN (point-to-point) ERS (EVPL).
- In this case, the 1:1 VLAN translation occurs on the U-PE, a 3750. It is provisioned on the NNI (uplink) port.
- The customer VLAN 123 is translated to the provider VLAN 778.

ERS (EVPL) (2:1 VLAN Translation)

Configuration

- Service: L2VPN/Metro Ethernet.
- Feature: ERS (EVPL) with VLAN 2:1 translation.Device configuration:
  - The N-PE is a Cisco 7600 with IOS 12.2(18)SXF, Sup720-3BXL

    Interface(s): FA8/34.
  - The U-PE is a Cisco 3750ME with IOS 12.2(25)EY1. VLAN translation on the NNI port (uplink).

    Interface(s): FA1/0/5 – GI1/1/1.
  - L2VPN point-to-point.

Configlets

U-PE	N-PE
vlan 567 exit ! interface FastEthernet1/0/5 no cdp enable no keepalive no ip address switchport **switchport access vlan 567** **switchport mode dot1q-tunnel** switchport trunk allowed vlan none switchport nonegotiate spanning-tree bpdufilter enable mac access-group ISC-FastEthernet1/0/5 in ! interface GigabitEthernet1/1/1 no ip address switchport trunk allowed vlan 1,123,567 **switchport vlan mapping dot1q-tunnel 567 234 779** ! mac access-list extended ISC-FastEthernet1/0/5 deny any host 0100.0ccc.cccc deny any host 0100.0ccc.cccd deny any host 0100.0ccd.cdd0 deny any host 0180.c200.0000 permit any any	vlan 779 exit ! interface FastEthernet8/34 switchport trunk allowed vlan 1,778-779 ! interface Vlan779 no ip address description L2VPN ERS 2 to 1 vlan translation xconnect 99.99.8.99 89033 encapsulation mpls no shutdown

Comments

- VLAN translation is only for L2VPN (point-to-point) ERS (EVPL).
- In this case, the 2:1 VLAN translation occurs on the U-PE, a 3750. It is provisioned on the NNI (uplink) port.
- The customer VLAN 123 and the provider VLAN 234 (as part of Q -in-Q) are translated to a new provider VLAN 779.

ERS (Pseudowire Class, E-Line, L2VPN Group Name, IOS XR Device)

Configuration

- Service: L2VPN/Metro Ethernet.
- Feature: ERS (EVPL).
- Device configuration:
  - The N-PE is a CRS-1 with IOS XR 3.6.1 or later.
  - UNI on N-PE.
  - UNI on U-PE.

Configlets

U-PE

```
!
vlan 700
exit
!
interface FastEthernet1/0/2
 switchport trunk encapsulation dot1q
 switchport trunk allowed vlan 700
 switchport mode trunk
 switchport nonegotiate
 no keepalive
 mac access-group ISC-FastEthernet1/0/2 in
 no cdp enable
 spanning-tree bpdufilter enable
!
!
interface GigabitEthernet1/0/1
 switchport trunk encapsulation dot1q
 switchport trunk allowed vlan 700
 switchport mode trunk
 keepalive 10
!
!
mac access-list extended
ISC-FastEthernet1/0/2
deny any host 0100.0ccc.cccc
deny any host 0100.0ccc.cccd
deny any host 0100.0ccd.cdd0
deny any host 0180.c200.0000
permit any any
!
```

N-PE

```
!
interface GigabitEthernet0/3/1/1.700
l2transport
 dot1q vlan 700
!
l2vpn
 pw-class PW_AD3-AD7_Customer1
  encapsulation mpls
   transport-mode vlan
   preferred-path interface tunnel-te 1370
fallback disable
  !
 !
 xconnect group L2VPN_Customer1-Gold_class
  p2p GoldPkg_AD3-AD7_Customer1
   interface GigabitEthernet0/3/1/1.700
   neighbor 192.169.105.30 pw-id 1000
    pw-class PW_AD3-AD7_Customer1
   !
  !
```

Comments

- The N-PE is a CRS-1 with IOS XR 3.7.
- The pseudowire class feature is configured with various associated attributes like encapsulation, transport mode, preferred-path, and fallback option.
- The disable fallback option is required for IOS XR 3.6.1 and optional for IOS XR 3.7 and later.
- The E-Line name (**p2p** command) and L2VPN Group Name (**xconnect group** command) is user configured.

ERS (EVPL) (NBI Enhancements for L2VPN, IOS Device)

Configuration

- Service: L2VPN/Metro Ethernet.
- Feature: ERS (EVPL).
- Device configuration:
  - The N-PE is a 12.2(18)SXF with IOS.
  - The U-PE is a 12.2(25)EY4with IOS.
  - UNI on N-PE.
  - UNI on U-PE.

Configlets

U-PE

```
!
vlan 3200
exit
!
interface FastEthernet1/0/2
no cdp enable
no ip address
duplex auto
switchport
switchport trunk encapsulation dot1q
switchport mode trunk
switchport trunk allowed vlan none
switchport trunk allowed vlan 3200
switchport nonegotiate
switchport port-security aging type
inactivity
switchport port-security maximum 100
switchport port-security aging time 1000
switchport port-security violation protect
switchport port-security
storm-control unicast level 1.0
storm-control broadcast level 50.0
storm-control multicast level 50.0
shutdown
keepalive
spanning-tree bpdufilter enable

!
interface GigabitEthernet1/0/1
no ip address
switchport
switchport trunk encapsulation dot1q
switchport mode trunk
switchport trunk allowed vlan 3200
!
```

N-PE

```
!
vlan 3300
exit
!
interface FastEthernet1/0/24
no cdp enable
no ip address
duplex auto
switchport
switchport trunk encapsulation dot1q
switchport mode trunk
switchport trunk allowed vlan none
switchport trunk allowed vlan 3300
switchport nonegotiate
switchport port-security aging type
inactivity
switchport port-security maximum 100
switchport port-security aging time 1000
switchport port-security violation protect
switchport port-security
storm-control unicast level 1.0
storm-control broadcast level 50.0
storm-control multicast level 50.0
shutdown
keepalive
spanning-tree bpdufilter enable

!
interface Vlan3300
no ip address
xconnect 192.169.105.40 7502 encapsulation
mpls
no shutdown
!
```

Comments

None.

ERS (EVPL) or EWS (EPL) (IOS XR Device)

Configuration

- Service: L2VPN/Metro Ethernet.
- Feature: ERS (EVPL) or EWS (EPL).
- Device configuration(s):
  - The N-PE is a CRS-1 with IOS XR 3.4.2.
  - UNI on N-PE. ERS (EVPL) only.
  - U-PE. EWS (EPL) or ERS (EVPL).

Configlets

N-PE

```
<?xml version="1.0" encoding="UTF-8"?>
<Request MajorVersion="1" MinorVersion="0">
  <Set>
    <Configuration Source="CurrentConfig">
      <InterfaceConfigurationTable>
        <InterfaceConfiguration>
          <Naming>
            <Name>GigabitEthernet0/0/0/1.302</Name>
            <Active>act</Active>
          </Naming>
          <InterfaceModeNonPhysical>L2Transport</InterfaceModeNonPhysical>
        </InterfaceConfiguration>
      </InterfaceConfigurationTable>
      <L2VPN>
        <Enabled>true</Enabled>
        <XConnectGroupTable>
          <XConnectGroup>
            <Naming>
              <Name>VPNSC</Name>
            </Naming>
            <Enabled>true</Enabled>
            <P2PXConnectTable>
              <P2PXConnect>
                <Naming>
                  <Name>GigabitEthernet0_0_0_1.302</Name>
                </Naming>
                <Enabled>true</Enabled>
                <AttachmentCircuitTable>
                  <AttachmentCircuit>
                    <Naming>
                      <Name>GigabitEthernet0/0/0/1.302</Name>
                    </Naming>
                    <Enabled>true</Enabled>
                  </AttachmentCircuit>
                </AttachmentCircuitTable>
                <PseudoWireTable>
                  <PseudoWire>
                    <Naming>
                      <Neighbor>
                        <IPV4Address>10.11.13.15</IPV4Address>
                      </Neighbor>
                      <PseudowireID>1005</PseudowireID>
                    </Naming>
                    <PseudoWireParameters/>
                  </PseudoWire>
                </PseudoWireTable>
              </P2PXConnect>
            </P2PXConnectTable>
          </XConnectGroup>
        </XConnectGroupTable>
      </L2VPN>
    </Configuration>
  </Set>
  <Commit/>
</Request>
```

Comments

- In IOS XR, device configuration is specified in XML format.

- With respect to the XML schemas, different versions of IOS XR generate different XML configlets. However the configurations will be almost identical, except for changes in the XML schema.
- There are different cases to consider. For example, when a service request is decommissioned or modified, the XML configuration will slightly differ.

ERS (EVPL) and EWS (EPL) (Local Connect on E-Line)

Configuration

- Service: L2VPN/Metro Ethernet.
- Feature: ERS (EVPL) and EWS (EPL).
- Device configuration:
  - The N-PE is a CRS-1 with IOS XR 3.6 or later.
  - The U-PE is a 12.2(18)SXF with IOS.

Configlets

U-PE	N-PE
	interface GigabitEthernet0/0/0/2.559 dot1q vlan 559 l2transport ! interface GigabitEthernet0/0/0/4.559 dot1q vlan 559 l2transport ! l2vpn xconnect group ISC p2p **cl-test-l2-crs1-1--0--559** interface GigabitEthernet0/0/0/2.559 interface GigabitEthernet0/0/0/4.559 ! ! !

Comments

- The default E-Line name has changed for local connect configlets.
- The format of the default E-line name is:
  device_name_with_underscores--VCID--VLANID

ERS (EVPL), EWS (EPL), ATM, or Frame Relay (Additional Template Variables for L2VPN, IOS and IOS XR Device)

Configuration

- Service: L2VPN/Metro Ethernet.
- Feature: ERS (EVPL), EWS (EPL), ATM and Frame Relay.
- Device configuration:
  - The N-PE is a 12.2(18)SXF with IOS for ERS (EVPL), EWS (EPL), Frame Relay service.
  - The N-PE is a CRS-1 with IOS XR 3.6 or later for ERS (EVPL), EWS (EPL) service; and IOS XR 3.7 or later for ATM service (ATM port mode).
  - The U-PE is a 12.2(25)EY4 with IOS for ERS (EVPL) or EWS (EPL) service.

Configlets

U-PE	N-PE

(None)	Template Content:
	`interface Loopback0`
	`description`
	`LocalLoopbackAddress=$L2VPNLocalLoopback`
	`LocalHostName=$L2VPNLocalHostName`
	`RemoteLoopbackAddress=$L2VPNRemoteLoopback`
	`RemoteHostName=$L2VPNRemoteHostName`
	
	Configlets:
	`interface Loopback0`
	`description LocalLoopbackAddress=`
	`192.169.105.40`
	`LocalHostName=cl-test-l2-7600-2`
	`RemoteLoopbackAddress=192.169.105.80`
	`RemoteHostName= cl-test-l2-7600-4`

Comments

- These four variables are supported only on the N-PE.
- The values will be empty for all other device roles (U-PE, PE-AGG, and CE).

EWS (EPL) (Point-to-Point)

Configuration

- Service: L2VPN/Metro Ethernet.
- Feature: EWS (EPL) (point-to-point).
- Device configuration:
  - The N-PE is a Cisco 7600 with IOS 12.2(18)SXF, Sup720-3BXL.

    Interface(s): FA8/17.
  - The U-PE is a Cisco 3750ME with IOS 12.2(25)EY1. No port security, no tunneling.

    Interface(s): FA1/0/20 – FA1/0/23.
  - L2VPN point-to-point.
  - Q-in-Q UNI.

Configlets

U-PE

```
system mtu 1522
!
vlan 774
exit
!
interface FastEthernet1/0/20
no cdp enable
no keepalive
switchport
switchport access vlan 774
switchport mode dot1q-tunnel
switchport nonegotiate
spanning-tree portfast
spanning-tree bpdufilter enable
!
interface FastEthernet1/0/23
no ip address
switchport trunk allowed vlan 774,787-788
```

N-PE

```
vlan 774
exit
!
interface FastEthernet8/17
switchport trunk allowed vlan
1,451,653,659,766-768,772,773-774,878
!
interface Vlan774
no ip address
description L2VPN EWS
xconnect 99.99.8.99 89029 encapsulation
mpls
no shutdown
```

Comments

- The N-PE is a 7600 with a OSM or SIP-600 module. Provisioning is the same as the ERS (EVPL) example.
- The U-PE is a generic Metro Ethernet (ME) switch.
- No PACL provisioned by default. BPDU can be tunneled if desired.
- The system MTU needs to set to 1522 to handle the extra 4 bytes of Q-in-Q frames.

EWS (EPL) (Point-to-Point, UNI Port Security, BPDU Tunneling)

Configuration

- Service: L2VPN/Metro Ethernet.
- Feature: EWS (EPL) (point-to-point) with Port security, BPDU tunneling.
- Device configuration:
  - The N-PE is a Cisco 7600 with IOS 12.2(18)SXF, Sup720-3BXL.
  - The U-PE is a Cisco 3750ME with IOS 12.2(25)EY1. No port security, with tunneling.
  - L2VPN point-to-point.
  - Q-in-Q UNI.

Configlets

U-PE	N-PE
system mtu 1522 ! vlan 775 exit ! system mtu 1522 ! vlan 775 exit ! interface FastEthernet1/0/19 no cdp enable no keepalive switchport switchport access vlan 775 switchport mode dot1q-tunnel switchport nonegotiate **switchport port-security maximum 34** **switchport port-security aging time 32** **switchport port-security violation shutdown** **switchport port-security** **l2protocol-tunnel cdp** **l2protocol-tunnel stp** **l2protocol-tunnel vtp** **l2protocol-tunnel shutdown-threshold cdp 88** **l2protocol-tunnel shutdown-threshold stp 99** **l2protocol-tunnel shutdown-threshold vtp 56** **l2protocol-tunnel drop-threshold cdp 56** **l2protocol-tunnel drop-threshold stp 64** **l2protocol-tunnel drop-threshold vtp 34** **storm-control unicast level 34.0** **storm-control broadcast level 23.0** **storm-control multicast level 12.0** spanning-tree portfast spanning-tree bpdufilter enable mac access-group ISC-FastEthernet1/0/19 in interface FastEthernet1/0/23 no ip address switchport trunk allowed vlan 774-775,787-788 ! mac access-list extended ISC-FastEthernet1/0/19 no permit any any **deny any host 3456.3456.1234** permit any any	vlan 775 exit ! interface FastEthernet8/17 switchport trunk allowed vlan 1,451,653,659,766-768,772,773-775,878 ! interface Vlan775 no ip address description L2VPN EWS xconnect 99.99.8.99 89029 encapsulation mpls no shutdown

Comments

- The N-PE is a 7600 with an OSM or SIP-600 module. Provisioning is the same as the ERS (EVPL) example.
- The U-PE is a generic Metro Ethernet (ME) switch.
- PACL with one user-defined entry.
- BPDUs (CDP, STP and VTP) are tunneled through the MPLS core.
- Storm control is enabled for unicast, multicast, and broadcast.

EWS (EPL) (Hybrid)

Configuration

- Service: L2VPN/Metro Ethernet.
- Feature: EWS (EPL) hybrid. One side is EWS (EPL) UNI; the other side is ERS (EVPL) NNI.
- Device configuration:
  - The N-PE is a Cisco 7600 with 12.2(18)SXF, Sup720-3BXL.

    Interface(s): FA8/17.
  - The U-PE is a Cisco 3750ME with 12.2(25)EY1. No port security, with tunneling.

    Interface(s): FA1/0/20 – FA1/0/23.
  - L2VPN point-to-point.
  - Q-in-Q UNI.

Note The first configlet example is the EWS (EPL) side (UNI). The second configlet is the ERS (EVPL) side (NNI).

Configlets (EWS)

U-PE	N-PE
system mtu 1522 ! vlan 775 exit ! system mtu 1522 ! vlan 775 exit ! interface FastEthernet1/0/19 no cdp enable no keepalive switchport switchport access vlan 775 switchport mode dot1q-tunnel switchport nonegotiate **switchport port-security maximum 34** **switchport port-security aging time 32** **switchport port-security violation shutdown** **switchport port-security** **l2protocol-tunnel cdp** **l2protocol-tunnel stp** **l2protocol-tunnel vtp** **l2protocol-tunnel shutdown-threshold cdp 88** **l2protocol-tunnel shutdown-threshold stp 99** **l2protocol-tunnel shutdown-threshold vtp 56** **l2protocol-tunnel drop-threshold cdp 56** **l2protocol-tunnel drop-threshold stp 64** **l2protocol-tunnel drop-threshold vtp 34** **storm-control unicast level 34.0** **storm-control broadcast level 23.0** **storm-control multicast level 12.0** spanning-tree portfast spanning-tree bpdufilter enable mac access-group ISC-FastEthernet1/0/19 in interface FastEthernet1/0/23 no ip address switchport trunk allowed vlan 774-775,787-788 ! mac access-list extended ISC-FastEthernet1/0/19 no permit any any **deny any host 3456.3456.1234** permit any any	vlan 775 exit ! interface FastEthernet8/17 switchport trunk allowed vlan 1,451,653,659,766-768,772,773-775,878 ! interface Vlan775 no ip address description L2VPN EWS xconnect 99.99.8.99 89029 encapsulation mpls no shutdown

Comments

- This is the EWS (EPL) side (UNI).
- N-PE is 7600 with an OSM or a SIP-600 module. Provisioning is the same as the ERS (EVPL).
- The U-PE is a generic Metro Ethernet (ME) switch.
- PACL with one user-defined entry.
- BPDUs (cdp, stp and vtp) are tunneled through the MPLS core.
- Storm control is enabled for unicast, multicast, and broadcast.

Configlets (ERS)

U-PE	N-PE
system mtu 1522 vlan 775 exit interface FastEthernet1/17 switchport trunk allowed vlan 1,451,653,659,766-768,772,773-775,878 interface FastEthernet1/10 switchport trunk allowed vlan 1,451,653,659,766-768,772,773-775,878	vlan 775 exit ! interface FastEthernet8/17 switchport trunk allowed vlan 1,451,653,659,766-768,772,773-775,878 ! interface Vlan775 no ip address description L2VPN EWS xconnect 99.99.8.99 89029 encapsulation mpls no shutdown

Comments

- This is the ERS (EVPL) side (NNI).
- The N-PE is a 7600 with an OSM or a SIP-600 module. Provisioning is the same as the ERS (EVPL).
- The U-PE is really a PE-AGG. It connects to the wholesale customer as an NNI. Both ports are regular NNI ports.

EWS (EPL) (Pseudowire Class, E-Line, L2VPN Group Name, IOS XR Device)

Configuration

- Service: L2VPN/Metro Ethernet.
- Feature: EWS (EPL).
- Device configuration:
  - The N-PE is a CRS-1 with IOS XR 3.6.1 or later.
  - UNI on U-PE.

Configlets

U-PE

```
!
system mtu 1522
!
vlan 700
exit
!
interface FastEthernet1/0/2
switchport
switchport access vlan 700
switchport mode dot1q-tunnel
switchport nonegotiate
no keepalive
no cdp enable
spanning-tree portfast
spanning-tree bpdufilter enable
!
interface GigabitEthernet1/0/1
no ip address
switchport
switchport trunk encapsulation dot1q
switchport trunk allowed vlan 700
switchport mode trunk
!
```

N-PE

```
!
interface GigabitEthernet0/3/1/1.700
l2transport
 dot1q vlan 700
!
!
l2vpn
 pw-class PW_AD7-AD3_Cutsomer2
  encapsulation mpls
   transport-mode ethernet
   preferred-path interface tunnel-te 2730
  !
 !
 xconnect group ISC
  p2p cl-test-l2-12404-2--1000
   interface GigabitEthernet0/3/1/1.700
   neighbor 192.169.105.30 pw-id 1000
    pw-class PW_AD7-AD3_Cutsomer2
    !
```

Comments

- The N-PE is a CRS-1 router with IOS XR 3.7.
- The pseudowire class feature is configured with various associated attributes like encapsulation, transport mode, preferred-path, and fallback option
- The disable fallback option is required for IOS XR 3.6.1 and optional for IOS XR 3.7 and later.
- The E-Line name (**p2p** command) and L2VPN Group Name (**xconnect group** command) is an Prime Fulfillment-generated default value, if user input is not provided.

EWS (EPL) (NBI Enhancements for L2VPN, IOS Device)

Configuration

- Service: L2VPN/Metro Ethernet.
- Feature: EWS (EPL).
- Device configuration:
  - The N-PE is a 12.2(18)SXF with IOS.
  - The U-PE is a 12.2(25)EY4with IOS.
  - UNI on N-PE.
  - UNI on U-PE.

Configlets

U-PE

```
!
vlan 3201
exit
!
interface FastEthernet1/0/2
no cdp enable
no ip address
duplex auto
switchport
switchport access vlan 3201
switchport mode dot1q-tunnel
switchport nonegotiate
switchport port-security aging type
inactivity
switchport port-security maximum 100
switchport port-security aging time 1000
switchport port-security violation protect
switchport port-security
storm-control unicast level 1.0
storm-control broadcast level 50.0
storm-control multicast level 50.0
shutdown
keepalive
spanning-tree bpdufilter enable

!
interface GigabitEthernet1/0/1
no ip address
switchport
switchport trunk encapsulation dot1q
switchport mode trunk
switchport trunk allowed vlan 3201
!
```

N-PE

```
!
vlan 3301
exit
!
interface FastEthernet1/0/24
no cdp enable
no ip address
duplex auto
switchport
switchport access vlan 3301
switchport mode dot1q-tunnel
switchport nonegotiate
switchport port-security aging type
inactivity
switchport port-security maximum 100
switchport port-security aging time 1000
switchport port-security violation protect
switchport port-security
storm-control unicast level 1.0
storm-control broadcast level 50.0
storm-control multicast level 50.0
shutdown
keepalive
spanning-tree bpdufilter enable

!
interface Vlan3301
no ip address
xconnect 192.169.105.40 7502 encapsulation
mpls
no shutdown
!
```

Comments

None.

ATM over MPLS (VC Mode)

Configuration

- Service: L2VPN.
- Feature: ATM over MPLS (ATMoMPLS, a type of AToM) in VC mode.
- Device configuration:
  - The N-PE is a Cisco 7200 with IOS 12.0(28)S.
  - No CE.
  - No U-PE.
  - L2VPN point-to-point (ATMoMPLS).
  - C7200 (ATM2/0).

Configlets

U-PE	N-PE

(None)	`interface ATM2/0.34234 point-to-point`
	`pvc 213/423 l2transport`
	`encapsulation aal5`
	`xconnect 99.99.4.99 89025 encapsulation mpls`

Comments

- The N-PE is any MPLS-enabled router.
- L2VPN provisioning is on the ATM VC connection.

ATM over MPLS (VP Mode)

Configuration

- Service: L2VPN.
- Feature: ATM over MPLS (ATMoMPLS, a type of AToM) in VP mode.
- Device configuration:
  - The N-PE is a Cisco 7200 with IOS 12.0(28)S.

    Interface(s): ATM2/0.
  - No CE.
  - No U-PE.
  - L2VPN point-to-point (ATMoMPLS).

Configlets

U-PE	N-PE

(None)	`pseudowire-class ISC-pw-tunnel-123`
	`encapsulation mpls`
	`preferred-path interface tunnel123`
	`disable-fallback`
	`!`
	`interface ATM2/0`
	`atm pvp 131 l2transport`
	`xconnect 99.99.4.99 89024 pw-class`
	`ISC-pw-tunnel-123`

Comments

- The N-PE is any MPLS-enabled router.
- L2VPN provisioning is on the ATM VP connection.
- The L2VPN pseudowire is mapped to a TE tunnel.

ATM (Port Mode, Pseudowire Class, E-Line, L2VPN Group Name, IOS XR Device)

Configuration

- Service: L2VPN/Metro Ethernet.
- Feature: ATM.
- Device configuration:
  - The N-PE is a CRS-1 with IOS XR 3.7 or later for ATM service (port mode only).
  - UNI on N-PE.

Configlets

U-PE	N-PE
(None)	interface ATM0/1/0/0 description UNIDesc_AC1 l2transport ! ! l2vpn pw-class PWClass-1 encapsulation mpls preferred-path interface tunnel-te 500 fallback disable ! ! xconnect group ISC p2p ELine_AC1 interface ATM0/1/0/0 neighbor 192.169.105.70 pw-id 100 pw-class PWClass-1 !

Comments

- The N-PE is a CRS-1 router.
- The pseudowire class feature is optional and not configured.
- The E-Line name (**p2p** command) and L2VPN Group Name (**xconnect group** command) are user configured.
- Only PORT mode is supported in IOS XR.
- This PORT mode will not generate any specific command, such as **pvp** or **pvc**, on IOS XR devices.
- The ATM interface is included under **xconnect**.

Frame Relay over MPLS

Configuration

- Service: L2VPN.
- Feature: Frame Relay over MPLS (FRoMPLS, a type of AToM).
- Device configuration:
  - The N-PE is a Cisco 7200 with IOS 12.0(28)S.

    Interface(s): ATM2/0.
  - No CE.
  - No U-PE.
  - L2VPN point-to-point (ATMoMPLS).

Configlets

U-PE	N-PE

(None)	`interface Serial1/1`
	`exit`
	`!`
	`connect C1_89001 Serial1/1 135 l2transport`
	`xconnect 99.99.4.99 89001 encapsulation mpls`

Comments

- The N-PE is any MPLS-enabled router.
- L2VPN provisioning is on the serial port for the Frame Relay connection.

Frame Relay (DLCI Mode)

Configuration

- Service: L2VPN over a L2TPv3 core.
- Feature: FR in DLCI mode.
- Device configuration:
  - The N-PE is a Cisco 7200 with IOS 12.0(28)S.

    Interface(s): ATM2/0.
  - No CE.
  - No U-PE.
  - L2VPN point-to-point (ATMoMPLS).

Configlets

U-PE	N-PE

(None)	`pseudowire-class ISC-pw-dynamic-default`
	`encapsulation l2tpv3`
	`ip local interface Loopback10`
	`ip dfbit set`
	`!`
	`interface Serial3/2`
	`encapsulation frame-relay`
	`exit`
	`!`
	`connect ISC_1054 Serial3/2 86 l2transport`
	`xconnect 10.9.1.1 1054 encapsulation l2tpv3`
	`pw-class ISC-pw-dynamic-default`

Comments

- The N-PE is any L2TPv3 enabled router.
- L2VPN provisioning is on the serial port for the Frame Relay connection.

VPLS (Multipoint, ERMS/EVP-LAN)

Configuration

- Service: L2VPN/Metro Ethernet.
- Feature: VPLS (multipoint) ERMS (EVP-LAN).
- Device configuration:
  - The N-PE is a Cisco 7600 with IOS 12.2(18)SXF, Sup720-3BX.L

    Interface(s): FA2/18.
  - The U-PE is a Cisco 3750ME with IOS 12.2(25)EY1. No port security, no tunneling.

    Interface(s): FA1/0/21 – FA1/0/23.
  - VPLS Multipoint VPN with VLAN 767.

Configlets

U-PE

```
vlan 767
exit
!
interface FastEthernet1/0/21
no cdp enable
no keepalive
no ip address
switchport
switchport trunk encapsulation dot1q
switchport mode trunk
switchport trunk allowed vlan none
switchport trunk allowed vlan 767
switchport nonegotiate
spanning-tree bpdufilter enable
mac access-group ISC-FastEthernet1/0/21 in
!
interface FastEthernet1/0/23
no ip address

mac access-list extended
ISC-FastEthernet1/0/21
deny any host 0100.0ccc.cccc
deny any host 0100.0ccc.cccd
deny any host 0100.0ccd.cdd0
deny any host 0180.c200.0000
permit any any
```

N-PE

```
l2 vfi vpls_ers_1-0 manual
vpn id 89017
neighbor 99.99.10.9 encapsulation mpls
neighbor 99.99.5.99 encapsulation mpls
!
vlan 767
exit
!
interface FastEthernet2/18
switchport trunk allowed vlan
350,351,430,630,767,780,783,785-791
!
interface Vlan767
no ip address
description VPLS ERS
xconnect vfi vpls_ers_1-0
no shutdown
```

Comments

- The N-PE is a 7600 with OSM or SIP-600 module.
- The VFI contains all the N-PEs (neighbors) that this N-PE talks to.
- The U-PE is a generic Metro Ethernet (ME) switch. The customer BPDUs are blocked by the PACL. The VPLS ERMS (EVP-LAN) UNI is the same as the L2VPN (point-to-point) ERS (EVPL) UNI.
- The SVI (interface 767) refers to the global VFI, which contains multiple peering N-PEs.

VPLS (Multipoint, EMS/EP-LAN), BPDU Tunneling)

Configuration

- Service: L2VPN/Metro Ethernet.
- Feature: VPLS (multipoint) EMS (EP-LAN) with BPDU tunneling.
- Device configuration:
  - The N-PE is a Cisco 7600 with IOS 12.2(18)SXF, Sup720-3BXL.

    Interface(s): FA2/18.
  - The U-PE is a Cisco 3750ME with IOS 12.2(25)EY1. No port security, no tunneling.

    Interface(s): FA1/0/12 – FA1/0/23.
  - VPLS Multipoint VPN, with VLAN 767.
  - Q-in-Q UNI.

Configlets

U-PE	N-PE
system mtu 1522 ! errdisable recovery interval 33 ! vlan 776 exit ! interface FastEthernet1/0/12 no cdp enable no keepalive switchport **switchport access vlan 776** **switchport mode dot1q-tunnel** **switchport nonegotiate** **l2protocol-tunnel cdp** **l2protocol-tunnel stp** **l2protocol-tunnel vtp** **l2protocol-tunnel shutdown-threshold cdp 88** **l2protocol-tunnel shutdown-threshold stp 64** **l2protocol-tunnel shutdown-threshold vtp 77** **l2protocol-tunnel drop-threshold cdp 34** **l2protocol-tunnel drop-threshold stp 23** **l2protocol-tunnel drop-threshold vtp 45** no shutdown spanning-tree portfast spanning-tree bpdufilter enable	**l2 vfi vpls_ews-89019 manual** **vpn id 89019** **neighbor 99.99.8.99 encapsulation mpls** **!** **vlan 776** **exit** **!** **interface FastEthernet8/17** **switchport trunk allowed vlan** **1,451,653,659,766-768,772-776,878** **!** **interface Vlan776** **no ip address** **description VPLS EWS** **xconnect vfi vpls_ews-89019** **no shutdown**

Comments

- The N-PE is a 7600 with an OSM or SIP-600 module.
- The VFI contains all the N-PEs (neighbors) that this N-PE talks to.
- The VPLS EMS (EP-LAN) UNI is the same as L2VPN (point-to-point) EWS (EPL) UNI.
- The SVI is the same as VPLS ERS (EVP-LAN) SVI.

EVC (Pseudowire Core Connectivity, UNI Port Security)

Configuration

- Service: EVC/Metro Ethernet.
- Feature: EVC with pseudowire core connectivity, with UNI port security.
- Device configuration:
  - The N-PE is a Cisco 7600 with IOS 12.2(33)SRB3.

    Interface(s): GI2/0/0.
  - The U-PE is a Cisco 3750ME with IOS 12.2(25)EY2. Port security is enabled.

    Interface(s): FA1/14– FA3/23.

Configlets

U-PE	N-PE
vlan 788 exit ! interface FastEthernet3/23 no ip address switchport trunk allowed vlan 783,787-788 ! interface FastEthernet1/14 no cdp enable no keepalive no ip address switchport **switchport trunk encapsulation dot1q** **switchport mode trunk** switchport trunk allowed vlan none **switchport trunk allowed vlan 788** **switchport port-security** switchport nonegotiate switchport port-security maximum 45 switchport port-security aging time 34 switchport port-security violation shutdown switchport port-security mac-address 3456.3456.5678 spanning-tree bpdufilter enable mac access-group ISC-FastEthernet3/23 in ! **mac access-list extended** **ISC-FastEthernet3/31** **deny any host 0100.0ccc.cccc** **deny any host 0100.0ccc.cccd** **deny any host 0100.0ccd.cdd0** **deny any host 0180.c200.0000** **deny any host 1234.3234.3432** **permit any any**	**interface GigabitEtherne4/0/1** **no shut** **service instance 10 ethernet** **encapsulation dot1q 500** **rewrite ingress tag push dot1q 555** **symmetric** **xconnect 192.169.105.20 505 encapsulation** **mpls**

Comments

- UNI on U-PE.
- Single match tag is performed.
- The rewrite operation **push** pushes the outer VLAN tag of 555.

EVC (Pseudowire Core Connectivity, UNI, without Port Security, with Bridge Domain)

Configuration

- Service: EVC/Metro Ethernet.
- Feature: EVC with pseudowire core connectivity, with UNI, without port security, and with bridge domain.
- Device configuration:
  - The N-PE is a Cisco 7600 with IOS 12.2(33)SRB3.

    Interface(s): GI2/0/0.
  - The U-PE is a Cisco 3750ME with IOS 12.2(25)EY2. Port security is enabled.

    Interface(s): FA1/14– FA3/23.

Configlets

U-PE	N-PE
vlan 772 exit ! interface FastEthernet3/23 **switchport trunk allowed vlan 500,772** ! interface FastEthernet1/14 no cdp enable no keepalive no ip address switchport trunk allowed vlan 500,772 spanning-tree bpdufilter enable mac access-group ISC-FastEthernet3/23 in ! **mac access-list extended** **ISC-FastEthernet1/14** **deny any host 0100.0ccc.cccc** **deny any host 0100.0ccc.cccd** **deny any host 0100.0ccd.cdd0** **deny any host 0180.c200.0000** **permit any any**	**vlan 100** **interface GigabitEtherne2/0/0** **no shut** **service instance 10 ethernet** **encapsulation dot1q 500** **rewrite ingress tag push dot1q 23** **second-dot1q 41 symmetric** **bridge-domain 100 split-horizon** **Interface Vlan100** **no shut** **xconnect 192.169.105.20 101 encapsulation** **mpls**

Comments

- UNI on U-PE.
- Single match tag is performed.
- The rewrite operation **push** pushes two tags.

EVC (Pseudowire Core Connectivity, UNI, and Pseudowire Tunneling)

Configuration

- Service: EVC/Metro Ethernet.
- Feature: EVC with pseudowire core connectivity, with UNI, with pseudowire tunneling.
- Device configuration:
  - The N-PE is a Cisco 7600 with IOS 12.2(33) SRB3.

    Interface(s): GI4/0/0 <–> GI2/0/0.

Configlets

U-PE	N-PE

(None)	`pseudowire-class ISC-pw-tunnel-2147`
	`encapsulation mpls`
	`preferred-path interface Tunnel2147`
	`disable-fallback`
	
	`interface GigabitEtherne4/0/0`
	`service instance 1 ethernet`
	`encapsulation dot1q 11 second-dot1q 41`
	`rewrite ingress tag pop 2 symmetric`
	`xconnect pw-class ISC-pw-tunnel-2147`

Comments

- UNI on N-PE (the CE is directly connected).
- Match of both tags is performed.
- The rewrite operation pops both the inner and outer VLAN tags.

EVC (VPLS Core Connectivity, UNI Port Security)

Configuration

- Service: EVC/Metro Ethernet.
- Feature: EVC with VPLS core connectivity, with UNI port security.
- Device configuration:
  - The N-PE is a Cisco 7600 with IOS 12.2(33) SRB3.

    Interface(s): GI4/0/1.
  - The U-PE is a Cisco 3750ME with IOS 12.2(25) EY2. Port security is enabled.

    Interface(s): FA1/14– FA3/23.

Configlets

U-PE

```
vlan 788
exit
!
interface FastEthernet3/23
no ip address
switchport trunk allowed vlan 783,787-788
!
interface FastEthernet1/14
no cdp enable
no keepalive
no ip address
switchport
switchport trunk encapsulation dot1q
switchport mode trunk
switchport trunk allowed vlan none
switchport trunk allowed vlan 788
switchport port-security
switchport nonegotiate
switchport port-security maximum 58
switchport port-security aging time 85
switchport port-security violation shutdown
switchport port-security mac-address
1252.1254.2544
spanning-tree bpdufilter enable
mac access-group ISC-FastEthernet3/23 in
!
mac access-list extended
ISC-FastEthernet3/31
deny any host 0100.0ccc.cccc
deny any host 0100.0ccc.cccd
deny any host 0100.0ccd.cdd0
deny any host 0180.c200.0000
deny any host 1234.3234.3432
permit any any
```

N-PE

```
l2 vfi attest-226 manual
vpn id 226
neighbor 192.169.105.20 encapsulation mpls

vlan 200
bridge-domain 200 split-horizon

interface GigabitEtherne4/0/1
no shut
service instance 10 ethernet
encapsulation dot1q 500
rewrite ingress tag translate 1-to-1 dot1q
222 symmetric

Interface vlan 200
xconnect vfi attest-226
```

Comments

- UNI on U-PE.
- The rewrite operation translates the incoming VLAN tag 500 to 222.

EVC (VPLS Core Connectivity, no UNI Port Security)

Configuration

- Service: EVC/Metro Ethernet.
- Feature: EVC with VPLS core connectivity, without UNI port security.
- Device configuration:
  - The N-PE is a Cisco 7600 with IOS 12.2(33) SRB3.

    Interface(s): GI4/0/1.
  - The U-PE is a Cisco 3750ME with IOS 12.2(25) EY2.

    Interface(s): FA1/14– FA3/23.

Configlets

U-PE

```
vlan 772
exit
!
interface FastEthernet3/23
switchport trunk allowed vlan 500,772
!
interface FastEthernet1/14
no cdp enable
no keepalive
no ip address
switchport trunk allowed vlan 500,772
spanning-tree bpdufilter enable
mac access-group ISC-FastEthernet3/23 in
!
mac access-list extended
ISC-FastEthernet1/14
deny any host 0100.0ccc.cccc
deny any host 0100.0ccc.cccd
deny any host 0100.0ccd.cdd0
deny any host 0180.c200.0000
permit any any
```

N-PE

```
l2 vfi attest1-458 manual
vpn id 452
neighbor 192.169.105.20 encapsulation mpls

vlan 200
bridge-domain 200 split-horizon

interface GigabitEtherne4/0/1
no shut
service instance 10 ethernet
encapsulation dot1q 500
rewrite ingress tag translate 1-to-2 dot1q
222 second-dot1q 41 symmetric

Interface vlan 200
xconnect vfi attest1-458
```

Comments

- UNI on U-PE.
- The rewrite operation translates the incoming VLAN tag 500 to two tags, 222 and 41.

EVC (Local Connect Core Connectivity, UNI Port Security)

Configuration

- Service: EVC/Metro Ethernet.
- Feature: EVC with local connect core connectivity, with UNI port security.
- Device configuration:
  - The N-PE is a Cisco 7600 with IOS 12.2(33) SRB3.

    Interface(s):GI2/0/0.
  - The U-PE is a Cisco 3750ME with IOS 12.2(25) EY2. Port security is enabled.

    Interface(s): FA1/14– FA3/23.

Configlets

U-PE

```
vlan 788
exit
!
interface FastEthernet3/23
no ip address
switchport trunk allowed vlan 783,787-788
!
interface FastEthernet1/14
no cdp enable
no keepalive
no ip address
switchport
switchport trunk encapsulation dot1q
switchport mode trunk
switchport trunk allowed vlan none
switchport trunk allowed vlan 788
switchport port-security
switchport nonegotiate
switchport port-security maximum 45
switchport port-security aging time 34
switchport port-security violation shutdown
switchport port-security mac-address
4111.4545.1211
spanning-tree bpdufilter enable
mac access-group ISC-FastEthernet3/23 in
!
mac access-list extended
ISC-FastEthernet3/31
deny any host 0100.0ccc.cccc
deny any host 0100.0ccc.cccd
deny any host 0100.0ccd.cdd0
deny any host 0180.c200.0000
deny any host 1234.3234.3432
permit any any
```

N-PE

```
Connect Customer_1 GigabitEthernet4/0/1 10
GigabitEthernet4/0/10 25

interface GigabitEtherne4/0/1
no shut
service instance 10 ethernet
encapsulation dot1q 500
rewrite ingress tag push dot1q 555
symmetric

interface GigabitEtherne4/0/10
no shut
service instance 25 ethernet
encapsulation dot1q 500 second-dot1q 501
rewrite ingress tag translate 2-to-1 dot1q
222 symmetric
```

Comments

- UNI on U-PE.
- Two tag matching operations are carried out.
- The rewrite operation translates two tags to a single tag.
- Two service instances are connected through the **connect** command.

EVC (Local Connect Core Connectivity, UNI, no Port Security, Bridge Domain)

Configuration

- EVC/Metro Ethernet.
- Feature: EVC with local connect core connectivity, with UNI, without port security, and with bridge domain.
- Device configuration:
  - The N-PE is a Cisco 7600 with IOS 12.2(33) SRB3.

    Interface(s):GI2/0/0.
  - The U-PE is a Cisco 3750ME with IOS 12.2(25) EY2.

    Interface(s):FA1/14– FA3/23.

Configlets

U-PE	N-PE
vlan 772 exit ! interface FastEthernet3/23 switchport trunk allowed vlan 500,772 ! interface FastEthernet1/14 no cdp enable no keepalive no ip address switchport trunk allowed vlan 500,772 spanning-tree bpdufilter enable mac access-group ISC-FastEthernet3/23 in ! **mac access-list extended** **ISC-FastEthernet1/14** **deny any host 0100.0ccc.cccc** **deny any host 0100.0ccc.cccd** **deny any host 0100.0ccd.cdd0** **deny any host 0180.c200.0000** permit any any	**interface GigabitEtherne2/0/0** **no shut** **service instance 10 ethernet** **encapsulation dot1q 500 second-dot1q 501** **rewrite ingress tag translate 2-to-2 dot1q 222 second-dot1q 41 symmetric** **bridge-domain 200 split-horizon** **interface GigabitEtherne2/0/10** **no shut** **service instance 15 ethernet** **encapsulation dot1q 24** **rewrite ingress tag pop 1 symmetric** **bridge-domain 200 split-horizon**

Comments

- UNI on U-PE.
- The rewrite operation maps/translates the incoming two tags into two different tags.
- The service instances here are connected through bridge domain.

EVC (Pseudowire Core Connectivity, Bridge Domain, Pseudowire on SVI)

Configuration

- EVC/Metro Ethernet.
- Feature: EVC with pseudowire core connectivity, with bridge domain, and with Pseudowire on SVI enabled on the N-PE.
- Device configuration:
  - The N-PE is a Cisco 7600 with IOS 12.2(33) SRB3.

    Interface(s): GigabitEthernet7/0/0.
  - The U-PE is a Cisco 3750ME with IOS 12.2(25) EY2.

    Interface(s): FastEthernet1/0/10.

Configlets

U-PE

```
vlan 452
exit
!
interface FastEthernet1/0/10
no ip address
switchport trunk allowed vlan add 452
!
interface FastEthernet1/0/13
no spanning-tree bpdufilter enable
switchport
no keepalive
no ip address
switchport
switchport trunk encapsulation dot1q
switchport mode trunk
switchport trunk allowed vlan 452
switchport nonegotiate
```

N-PE

```
vlan 3524
exit
!
ethernet evc Customer1_253
!
interface GigabitEthernet7/0/0
service instance 3 ethernet Customer1_253
encapsulation dot1q 452
rewrite ingress tag pop 1 symmetric
bridge-domain 3524 split-horizon
!
interface Vlan3524
no ip address
description BD=T,SVI=T,Flex
xconnect 22.22.22.22 52500 encapsulation
mpls
backup peer 22.22.22.22 52501
no shutdown
```

Comments

- None.

EVC (Pseudowire Core Connectivity, no Bridge Domain, no Pseudowire on SVI)

Configuration

- EVC/Metro Ethernet.
- Feature: EVC with pseudowire core connectivity, bridge domain disables, and with Pseudowire on SVI disabled on the N-PE.
- Device configuration:
  - The N-PE is a Cisco 7600 with IOS 12.2(33) SRB3.

    Interface(s): GigabitEthernet7/0/0.
  - The U-PE is a Cisco 3750ME with IOS 12.2(25) EY2.

    Interface(s): FastEthernet1/0/10.

Configlets

U-PE	N-PE
vlan 545 exit ! interface FastEthernet1/0/10 no ip address switchport trunk allowed vlan add 545 ! interface FastEthernet1/0/12 no spanning-tree bpdufilter enable switchport no keepalive no ip address switchport switchport trunk encapsulation dot1q switchport mode trunk switchport trunk allowed vlan 545 switchport nonegotiate mac access-group ISC-FastEthernet1/0/12 in	ethernet evc Customer1_248 ! interface GigabitEthernet7/0/0 **service instance 2 ethernet Customer1_248** encapsulation dot1q 545 rewrite ingress tag pop 1 symmetric **xconnect 22.22.22.22 52498 encapsulation mpls** **backup peer 22.22.22.22 52499**

Comments

- None.

EVC (AutoPick Service Instance Name)

Configuration

- EVC/Metro Ethernet.
- Feature: EVC with AutoPick Service Instance Name enabled and the Service Instance Name input field left blank.
- Device configuration:
  - The N-PE is a Cisco 7600 with IOS 12.2(33) SRB3.

    Interface(s): GigabitEthernet7/0/2.
  - The U-PE is a Cisco 3750ME with IOS 12.2(25) EY2.

    Interface(s): FastEthernet1/0/14.

Configlets

U-PE	N-PE
!	!
vlan 452	vlan 3524
exit	exit
!	!
interface FastEthernet1/0/10	ethernet evc C1_1
no ip address	!
switchport trunk allowed vlan add 452	interface GigabitEthernet7/0/0
!	service instance 3 ethernet C1_1
interface FastEthernet1/0/13	encapsulation dot1q 452
no spanning-tree bpdufilter enable	rewrite ingress tag pop 1 symmetric
switchport	bridge-domain 3524 split-horizon
no keepalive	!
no ip address	interface Vlan3524
switchport	no ip address
switchport trunk encapsulation dot1q	description BD=T,SVI=T,Flex
switchport mode trunk	xconnect 22.22.22.22 52500 encapsulation
switchport trunk allowed vlan 452	mpls
switchport nonegotiate	backup peer 22.22.22.22 52501
mac access-group ISC-FastEthernet1/0/13 in	no shutdown

Comments

- The transport type is pseudowire.
- The autopick Service Instance Name will take the value *CustomerName_JobID*.

EVC (No AutoPick Service Instance Name, No Service Instance Name)

Configuration

- EVC/Metro Ethernet.
- Feature: EVC with AutoPick Service Instance Name not enabled and the Service Instance Name input field left blank.
- Device configuration:
  - The N-PE is a Cisco 7600 with IOS 12.2(33) SRB3.

    Interface(s): GigabitEthernet7/0/2.
  - The U-PE is a Cisco 3750ME with IOS 12.2(25) EY2.

    Interface(s): FastEthernet1/0/14.

Configlets

U-PE

```
!
vlan 566
exit
!
interface FastEthernet1/0/14
no spanning-tree bpdufilter enable
switchport
no keepalive
no ip address
switchport trunk encapsulation dot1q
switchport mode trunk
switchport trunk allowed vlan 566
switchport nonegotiate
no shutdown
mac access-group ISC-FastEthernet1/0/14 in
!
interface FastEthernet1/0/18
no ip address
switchport trunk allowed vlan 566
!
mac access-list extended
ISC-FastEthernet1/0/14
deny any host 0100.0ccc.cccc
deny any host 0100.0ccc.cccd
deny any host 0100.0ccd.cdd0
deny any host 0180.c200.0000
permit any any
```

N-PE

```
!
interface GigabitEthernet7/0/2
service instance 43 ethernet
encapsulation dot1q 566
xconnect 1.1.1.1 453366 encapsulation mpls
```

Comments

- In this example, the user does not enable AutoPick Service Instance Name and also leaves the Service Instance Name input field blank.
- The global command **ethernet evc** is not generated, while the command **service instance 43 ethernet** is generated.
- There is no Service Instance Name available and the Service Instance ID is 43.

EVC (User-Provided Service Instance Name, Pseudowire Core Connectivity)

Configuration

- EVC/Metro Ethernet.
- Feature: EVC with pseudowire core connectivity and user-provided service instance name.
- Device configuration:
  - The N-PE is a Cisco 7600 with IOS 12.2(33) SRB3.

    Interface(s): GigabitEthernet7/0/0.
  - The U-PE is a Cisco 3750ME with IOS 12.2(25) EY2.

    Interface(s): FastEthernet1/0/10.

Configlets

U-PE

```
!
vlan 452
exit
!
interface FastEthernet1/0/10
no ip address
switchport trunk allowed vlan add 452
!
interface FastEthernet1/0/13
no spanning-tree bpdufilter enable
switchport
no keepalive
no ip address
switchport
switchport trunk encapsulation dot1q
switchport mode trunk
switchport trunk allowed vlan 452
switchport nonegotiate
mac access-group ISC-FastEthernet1/0/13 in
```

N-PE

```
!
vlan 3524
exit
!
ethernet evc ServiceInst
!
interface GigabitEthernet7/0/0
service instance 3 ethernet  ServiceInst
encapsulation dot1q 452
rewrite ingress tag pop 1 symmetric
bridge-domain 3524 split-horizon
!
interface Vlan3524
no ip address
description BD=T,SVI=T,Flex
xconnect 22.22.22.22 52500 encapsulation
mpls
backup peer 22.22.22.22 52501
no shutdown
```

Comments

- The transport type is PSEUDOWIRE.
- The user manually provided **ServiceInst** as the Service Instance Name. This is pushed onto the device, where the Service Instance ID is 3.

EVC (User-Provided Service Instance Name, Local Core Connectivity)

Configuration

- EVC/Metro Ethernet.
- Feature: EVC with local core connectivity and a user-provided service instance name.
- Device configuration:
  - The N-PE is a Cisco 7600 with IOS 12.2(33) SRB3.

    Interface(s): GigabitEthernet1/0/6, GigabitEthernet1/0/7.
  - The U-PE is a Cisco 3750ME with IOS 12.2(25) EY2.

    Interface(s): FastEthernet1/0/12, FastEthernet1/0/14.

Configlets

U-PE

```
vlan 45
exit
!
interface FastEthernet1/0/12
no ip address
switchport
switchport trunk encapsulation dot1q
switchport mode trunk
switchport trunk allowed vlan 45
!
interface FastEthernet1/0/14
no spanning-tree bpdufilter enable
switchport
no keepalive
no ip address
switchport trunk encapsulation dot1q
switchport mode trunk
switchport trunk allowed vlan 45
switchport nonegotiate
no shutdown
mac access-group ISC-FastEthernet1/0/14 in
!
mac access-list extended
ISC-FastEthernet1/0/14
deny any host 0100.0ccc.cccc
deny any host 0100.0ccc.cccd
deny any host 0100.0ccd.cdd0
deny any host 0180.c200.0000
permit any any
```

N-PE

```
ethernet evc  service_int
!
interface GigabitEthernet1/0/6
no shutdown
service instance 5 ethernet service_int
encapsulation dot1q 56
!
interface GigabitEthernet1/0/7
no shutdown
service instance 33 ethernet service_int
 encapsulation dot1q 45
!
connect Customer2_195 GigabitEthernet1/0/7
33 GigabitEthernet1/0/6 5
```

Comments

- The transport type is LOCAL.
- The user manually provided **service_int** as the Service Instance Name. This is pushed onto the device, where the Service Instance IDs are 5 and 33, respectively.

EVC (User-Provided Service Instance Name, VPLS Core Connectivity)

Configuration

- EVC/Metro Ethernet.
- Feature: EVC with VPLS core connectivity and user-provided service instance name.
- Device configuration:
  - The N-PE is a Cisco 7600 with IOS 12.2(33) SRB3.

    Interface(s): GigabitEthernet7/0/0.
  - The U-PE is a Cisco 3750ME with IOS 12.2(25) EY2.

    Interface(s): FastEthernet1/0/10.

Configlets

U-PE

```
!
vlan 452
exit
!
interface FastEthernet1/0/10
no ip address
switchport trunk allowed vlan add 452
!
interface FastEthernet1/0/13
no spanning-tree bpdufilter enable
switchport
no keepalive
no ip address
switchport
switchport trunk encapsulation dot1q
switchport mode trunk
switchport trunk allowed vlan 452
switchport nonegotiate
mac access-group ISC-FastEthernet1/0/13 in
```

N-PE

```
l2 vfi vpls-test manual
vpn id 300
neighbor 22.22.22.22 encapsulation mpls
!
vlan 500
!
ethernet evc ServiceInst
!
interface GigabitEtherne7/0/0
service instance 10 ethernet ServiceInst
encapsulation dot1q 400
rewrite ingress tag pop 1 symmetric
bridge-domain 500 split-horizon
!
interface vlan500
xconnect vfi vpls-test
```

Comments

- The transport type is VPLS.
- The user manually provided **ServiceInst** as the Service Instance Name. This is pushed onto the device, where the Service Instance ID is 10.

EVC (ATM-Ethernet Interworking, Pseudowire Core Connectivity, Point-to-Point Circuit)

Configuration

- EVC/ATM-Ethernet Interworking.
- Feature: EVC for ATM-Ethernet interworking with pseudowire core connectivity with an end-to-end circuit with multiple links. One link terminates on an ATM interface on N-PE 1, and the other link terminates on an Ethernet interface on N-PE 2.
- Device configuration:
  - N-PE 1 is a Cisco 7600 with IOS 12.2(33) SRB3.

    Interface(s): ATM1/0/0.370.
  - N-PE 2 is a Cisco 7600 with IOS 12.2(33) SRE.

    Interface(s): GigabitEthernet4/0/2.

Configlets

N-PE 1 (ATM)	N-PE 2 (Ethernet)
!	!
interface ATM1/0/0.370 point-to-point	ethernet evc 1-3_51
no atm enable-ilmi-trap	!
pvc 0/370 l2transport	interface GigabitEthernet4/0/2
encapsulation aal5snap	no ip address
xconnect 192.169.105.10 123 pw-class inter-ether	no mls qos trust
!	service instance 103 ethernet 1-3_51
	encapsulation dot1q 370
	rewrite ingress tag pop 1 symmetric
	xconnect 192.169.105.20 123 encapsulation mpls
	!

Comments

- None.

EVC (ATM-Ethernet Interworking, Pseudowire Core Connectivity, Multipoint Circuit)

Configuration

- EVC/ATM-Ethernet Interworking.
- Feature: EVC for ATM-Ethernet interworking with pseudowire core connectivity with a multipoint circuit. Link #1 terminates on an ATM interface on N-PE 1, link #2 terminates on an Ethernet interface on N-PE 1, and link #3 terminates on an Ethernet interface on N-PE 2.
- Device configuration:
  - The N-PE 1 is a Cisco 7600 with IOS 12.2(33) SRB3.

    Interface(s): GigabitEthernet7/0/4, ATM6/0/0.100.
  - The N-PE 2 is a Cisco 7600 with IOS 12.2(33) SRE.

    Interface(s): GigabitEthernet7/0/5.

Configlets

N-PE 1 (ATM + Ethernet)

```
!
vlan 500
exit
!
ethernet evc Customer1_166
!
interface GigabitEthernet7/0/4
no shutdown
service instance 1 ethernet Customer1_166
encapsulation dot1q 600
bridge-domain 500 split-horizon
!
interface ATM6/0/0.100 point-to-point
pvc 200/300
encapsulation aal5snap
bridge-domain 500 split-horizon
!
interface Vlan500
no ip address
description UT-9
xconnect 1.1.1.1 6 pw-class
ISC-pw-tunnel-400
no shutdown
```

N-PE 2 (Ethernet)

```
!
vlan 800
exit
!
ethernet evc Customer1_166
!
interface GigabitEthernet7/0/5
no shutdown
service instance 1 ethernet Customer1_166
encapsulation dot1q 623
bridge-domain 800 split-horizon
!
interface Vlan800
description UT-9
xconnect 192.169.105.20 6 pw-class
ISC-pw-tunnel-900
```

Comments

- None.

EVC (ATM-Ethernet Interworking, Local Core Connectivity, Point-to-Point Circuit)

Configuration

- EVC/ATM-Ethernet Interworking.
- Feature: EVC for ATM-Ethernet interworking with local core connectivity with a point-to-point circuit. The circuit terminates on different ATM interfaces on the same local N-PE.
- Device configuration:
  - The N-PE 1 is a Cisco 7600 with IOS 12.2(33) SRB3.

    Interface(s): ATM1/0/1, ATM4/1/0, ATM1/0/1.99, ATM4/1/0.98.

Configlets

N-PE 1 (ATM)	N/A
! interface ATM1/0/1 no shutdown ! interface ATM4/1/0 no shutdown ! interface ATM1/0/1.99 point-to-point pvc 99/99 l2transport encapsulation aal0 ! interface ATM4/1/0.98 point-to-point pvc 98/98 l2transport encapsulation aal0 ! connect ATM-to-ATM ATM1/0/1 99/99 ATM4/1/0 98/98 !	

Comments

- None.

EVC (ATM-Ethernet Interworking, Local Core Connectivity, Multipoint Circuit)

Configuration

- EVC/ATM-Ethernet Interworking.
- Feature: EVC for ATM-Ethernet interworking with local core connectivity for multiple links that terminate on the same local N-PE. Link #1 terminates on an ATM interface, and link #2 terminates on an Ethernet interface.
- Device configuration:
  - The N-PE 1 is a Cisco 7600 with IOS 12.2(33) SRB3.

    Interface(s): ATM1/0/0.99, TenGigabitEthernet6/0/0, TenGigabitEthernet6/0/1.

Configlets

N-PE 1 (ATM + Ethernet)	N/A

```
!
vlan 1001
exit
!
interface ATM1/0/0.99 point-to-point
 no atm enable-ilmi-trap
 pvc 99/99
  encapsulation aal5snap
 bridge-domain 1001
 !
!
interface TenGigabitEthernet6/0/0
 no ip address
 no mls qos trust
 service instance 104 ethernet 1-4_60
  encapsulation dot1q 11
  rewrite ingress tag pop 1 symmetric
  bridge-domain 1001
 !
!
interface TenGigabitEthernet6/0/1
 no ip address
 no mls qos trust
 service instance 105 ethernet 1-4_60
  encapsulation dot1q 12
  rewrite ingress tag pop 1 symmetric
  bridge-domain 1001
 !
```

Comments

- None.

EVC (ATM-Ethernet Interworking, Local Core Connectivity, Multipoint Circuit)

Configuration

- EVC/ATM-Ethernet Interworking.
- Feature: EVC for ATM-Ethernet interworking with local core connectivity. Multiple links terminate on the same local N-PE. Link #1 terminates on an ATM interface, link #2 terminates on an ATM interface, and link #3 terminates on an ATM interface.
- Device configuration:
  - The N-PE 1 is a Cisco 7600 with IOS 12.2(33) SRB3.

    Interface(s): ATM6/0/0.100, ATM6/0/1.101, ATM6/0/2.102.

Configlets

N-PE 1 (ATM)	N/A

```
!
vlan 500
exit
!
interface ATM6/0/0.100 point-to-point
pvc 200/300
encapsulation aal5snap
bridge-domain 500
!
interface ATM6/0/1.101 point-to-point
pvc 201/301
encapsulation aal5snap
bridge-domain 500
!
interface ATM6/0/2.102 point-to-point
pvc 202/302
encapsulation aal5snap
bridge-domain 500
!
```

Comments

- None.

EVC (ATM-Ethernet Interworking, Local Core Connectivity, Point-to-Point Circuit)

Configuration

- EVC/ATM-Ethernet Interworking.
- Feature: EVC for ATM-Ethernet interworking with local core connectivity. A point-to-point circuit terminates on different ATM interfaces on same local N-PE.
- Device configuration:
  - The N-PE 1 is a Cisco 7600 with IOS 12.2(33) SRB3.

    Interface(s): ATM1/0/0, ATM1/0/1.

Configlets

N-PE 1 (ATM)	N/A
! interface ATM1/0/0 atm pvp 33 l2transport ! interface ATM1/0/1 atm pvp 222 l2transport ! connect Customer1_208 ATM1/0/0 33 ATM1/0/1 222	

Comments

- None.

EVC (ATM-Ethernet Interworking, Pseudowire Core Connectivity, End-to-End Circuit)

Configuration

- EVC/ATM-Ethernet Interworking.
- Feature: EVC for ATM-Ethernet interworking with pseudowire core connectivity for end-to-end circuit with multiple links. One link terminates on ATM interface on N-PE 1, and other link terminates on an Ethernet interface on N-PE 2.
- Device configuration:
  - The N-PE 1 is a Cisco 7600 with IOS 12.2(33) SRB3.

    Interface(s): ATM1/0/0.370.
  - The N-PE 2 is a Cisco ASR 9000 with IOS XR 3.9.0.

    Interface(s): GigabitEthernet0/0/0/4.458.

Configlets

N-PE 1 (ATM)

```
!
interface ATM1/0/0.370 point-to-point
 no atm enable-ilmi-trap
 pvc 0/370 l2transport
  encapsulation aal5snap
  xconnect 192.169.105.10 123 pw-class
inter-ether
 !
```

N-PE 2 (Ethernet)

```
interface GigabitEthernet0/0/0/4.458
l2transport
  encapsulation dot1q 458
!
l2vpn
 xconnect group VPNSC
  p2p iscind-crs-1--48856
   interface GigabitEthernet0/0/0/4.458
   neighbor 192.168.118.167 pw-id 123
  !
 !
!
```

Comments

- None.

EVC (ATM-Ethernet Interworking, Pseudowire Core Connectivity, Multipoint Circuit)

Configuration

- EVC/ATM-Ethernet Interworking.
- Feature: EVC for ATM-Ethernet interworking with pseudowire core connectivity with an end-to-end circuit with multiple links. One link is terminating on an ATM interface on N-PE 1, and the other (non-flex) link terminates on an Ethernet interface on N-PE 2.
- Device configuration:
  - The N-PE 1 is a Cisco 7600 with IOS 12.2(33) SRB3.

    Interface(s): ATM4/1/0.8790.
  - The N-PE 2 is a Cisco 7600 with IOS 12.2(33) SRB3.

    Interface(s): GigabitEthernet4/0/17.600.

Configlets

N-PE 1 (ATM)	N-PE 2 (Ethernet)

`interface ATM4/1/0.8790 point-to-point`
`pvc 150/3454 l2transport`
`encapsulation aal5snap`
`xconnect 192.169.105.10 760 pw-class`
`ISC-pw-tunnel-1`
	`interface GigabitEthernet4/0/17.600`
	`encapsulation dot1Q 600`
	`xconnect 192.169.105.20 760 pw-class`
	`ISC-pw-tunnel-1`

Comments

- None.

EVC (ATM-Ethernet Interworking, Local Core Connectivity, Point-to-Point Circuit)

Configuration

- EVC/ATM-Ethernet Interworking.
- Feature: EVC for ATM-Ethernet interworking with local core connectivity for point-to-point circuit. The circuit terminates on the same, local N-PE 1. One link terminates on an ATM interface, and the other (non-flex) link terminates on an Ethernet interface.
- Device configuration:
  - The N-PE 1 is a Cisco 7600 with IOS 12.2(33) SRB3.

    Interface(s): ATM1/0/0.444.
  - The N-PE 1 is a Cisco 7600 with IOS 12.2(33) SRB3.

    Interface(s): FastEthernet3/39.674.

Configlets

N-PE 1 (ATM + Ethernet)	N/A
! interface FastEthernet3/39.674 encapsulation dot1Q 674 ! interface ATM1/0/0.444 point-to-point pvc 44/4444 l2transport encapsulation aal5snap ! connect Customer1_204 ATM1/0/0 44/4444 FastEthernet3/39.674 interworking ethernet	

Comments

- None.

EVC (ATM-Ethernet Interworking, Pseudowire Core Connectivity, End-to-End Circuit, with Bridge Domain)

Configuration

- EVC/ATM-Ethernet Interworking.
- Feature: EVC for ATM-Ethernet interworking with pseudowire core connectivity for end-to-end circuit with multiple links with bridge domain enabled. One link terminates on an ATM interface on N-PE 1, and the other link terminates on a flex Ethernet interface on N-PE 2.
- Device configuration:
  - The N-PE 1 is a Cisco 7600 with IOS 12.2(33) SRB3.

    Interface(s): ATM1/0/0.370.
  - The N-PE 2 is a Cisco ASR 9000 with IOS XR 3.9.0.

    Interface(s): GigabitEthernet0/0/0/25.341.

Configlets

N-PE 1 (ATM)	N-PE 2 (Ethernet)

!
 interface ATM1/0/0.370 point-to-point
  no atm enable-ilmi-trap
  pvc 0/370 l2transport
  encapsulation aal5snap
 xconnect 10.20.21.1 4531 pw-class
 ISC-pw-tunnel-1
	interface GigabitEthernet0/0/0/25.341
	 l2transport
	  encapsulation dot1q 341
	  rewrite ingress tag push dot1q 430
	 second-dot1q 349 symmetric
	 !
	 l2vpn
	  bridge group tml
	  bridge-domain CISCO
	  interface GigabitEthernet0/0/0/25.341
	  !
	  neighbor 192.169.105.20 pw-id 32190
	  !
	  !
	  !
	 !

Comments

- None.

EVC (ATM-Ethernet Interworking, Pseudowire Core Connectivity, End-to-End Circuit, with Bridge Domain)

Configuration

- EVC/ATM-Ethernet Interworking.
- Feature: EVC for ATM-Ethernet interworking with pseudowire core connectivity for end-to-end circuit with multiple links. Bridge domain is enabled. One link terminates on an ATM interface on N-PE 1, and the other (non-flex) link terminates on an Ethernet interface on N-PE 2.
- Device configuration:
  - The N-PE 1 is a Cisco 7600 with IOS 12.2(33) SRB3.

    Interface(s): ATM1/0/0.370.
  - The N-PE 2 is a Cisco ASR 9000 with IOS XR 3.9.0.

    Interface(s): GigabitEthernet0/0/0/20.712.

Configlets

N-PE 1 (ATM)

```
!
interface ATM1/0/0.370 point-to-point
 no atm enable-ilmi-trap
 pvc 0/370 l2transport
  encapsulation aal5snap
  xconnect 10.20.21.1 4531 pw-class
ISC-pw-tunnel-1
 !
```

N-PE 2 (Ethernet)

```
interface GigabitEthernet0/0/0/20.712
l2transport
 encapsulation dot1q 712
!
l2vpn
 bridge group tml
  bridge-domain CISCO
   interface GigabitEthernet0/0/0/20.712
   !
   neighbor 192.169.105.20 pw-id 1005
   !
  !
 !
!
```

Comments

- None.

EVC (ATM-Ethernet Interworking, Pseudowire Core Connectivity, End-to-End Circuit, no Bridge Domain)

Configuration

- EVC/ATM-Ethernet Interworking.
- Feature: EVC for ATM-Ethernet interworking with pseudowire core connectivity for end-to-end circuit with multiple links. Bridge domain is disabled. One link is terminates on an ATM interface on N-PE 1, and the other link terminates on an Ethernet interface on N-PE 2.
- Device configuration:
  - The N-PE 1 is a Cisco 7600 with IOS 12.2(33) SRB3.

    Interface(s): ATM1/0/0.370.
  - The N-PE 2 is a Cisco ASR 9000 with IOS XR 3.9.0.

    Interface(s): GigabitEthernet0/0/0/12.433.

Configlets

N-PE 1 (ATM)	N-PE 2 (Ethernet)

!
 interface ATM1/0/0.370 point-to-point
  no atm enable-ilmi-trap
  pvc 0/370 l2transport
  encapsulation aal5snap
 xconnect 10.20.21.1 4531 pw-class
 ISC-pw-tunnel-1 !
	interface GigabitEthernet0/0/0/12.433
	 l2transport
	  encapsulation dot1q 433
	  rewrite ingress tag push dot1q 43
	 second-dot1q 53 symmetric
	 !
	 l2vpn
	  xconnect group ISC
	  p2p CISCO
	  interface GigabitEthernet0/0/0/12.433
	  neighbor 192.169.105.20 pw-id 4531
	  !
	  !
	  !
	 !

Comments

- None.
-

CHAPTER 4

Managing RAN Backhaul Services

This chapter describes how to use Prime Fulfillment to manage radio access network (RAN) backhaul services in Prime Fulfillment. It contains the following sections:

Overview of RAN Backhaul Services

Radio access network (RAN) transport manages the backhaul traffic (both voice and data) from the cell site base transceiver stations (BTSs) to aggregation nodes and to base station controllers (BSCs), between BSCs, and between the BSC and an associated mobile switching center (MSC). Figure 4-1 shows an example RAN backhaul topology.

Figure 4-1 Example RAN Backhaul Topology

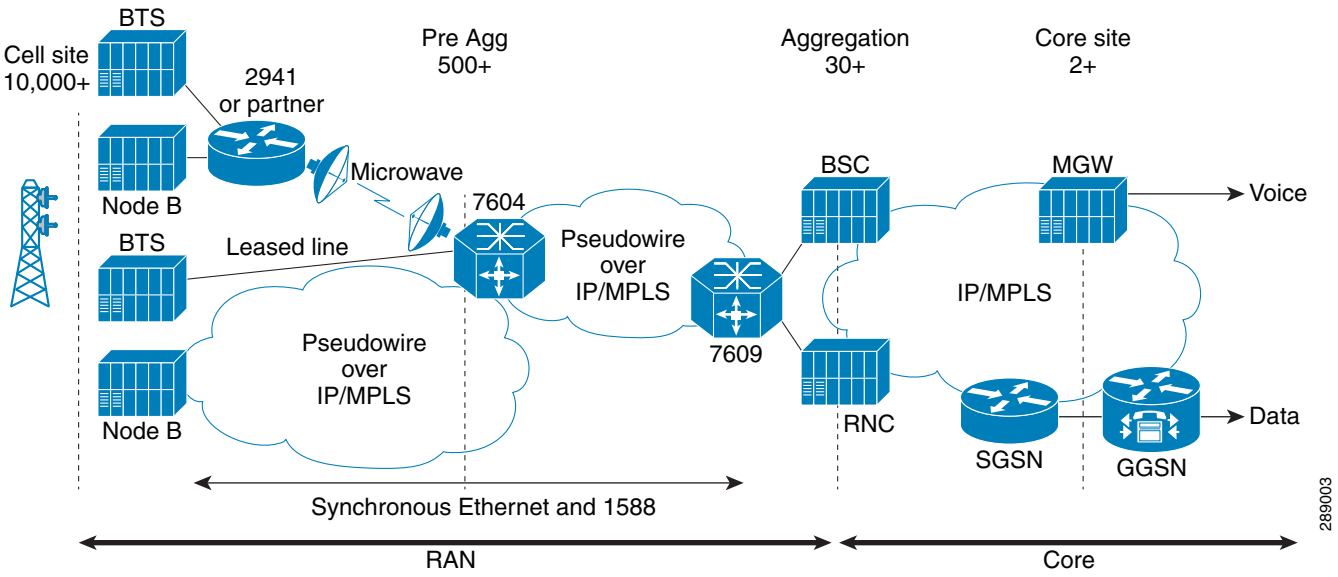
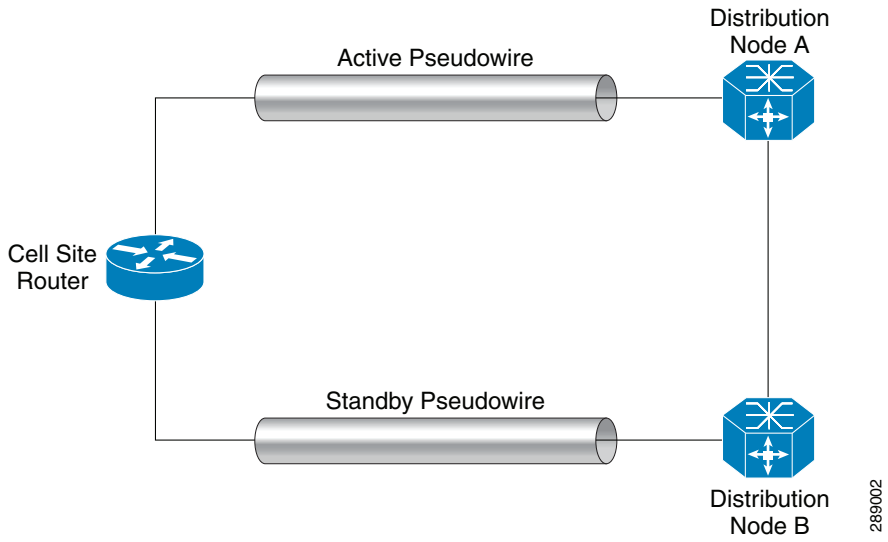


Figure 4-2 is an abstracted topology view that is used in this chapter when discussing how to configure RAN backhaul services in the Prime Fulfillment GUI.

Figure 4-2 Abstracted RAN Backhual Topology



Prime Fulfillment uses Internet Protocol (IP) to transport backhaul traffic in RANs. You use Ethernet Virtual Circuit (EVC) policies and service requests in Prime Fulfillment to provision the following services to support RAN backhaul traffic management:

- Circuit Emulation Time Delay Multiple Access (CEM TDM)
- Pseudowire provisioning of Asynchronous Transfer Mode (ATM)

In addition, the EVC service requests use CEM and pseudowire class objects to bundle common attributes for resuse on every node where the service is provisioned.

The basic workflow for configuring and managing RAN backhaul services in Prime Fulfillment, involves the following tasks:

1. Verify prerequisites and preform necessary setup tasks.
2. Create CEM and/or pseudowire classes to be used in RAN backhaul policies and service requests.
3. Create the CEM TDM or ATM policy.
4. Create template(s) for use in the CEM TDM or ATM service request.
5. Create the CEM TDM or ATM service request.
6. Deploy the service request to the device(s) on the network.

The chapter is organized into two sections, one for CEM TDM services and one for ATM services. The above workflow tasks are documented in each of these sections.

Prerequisites

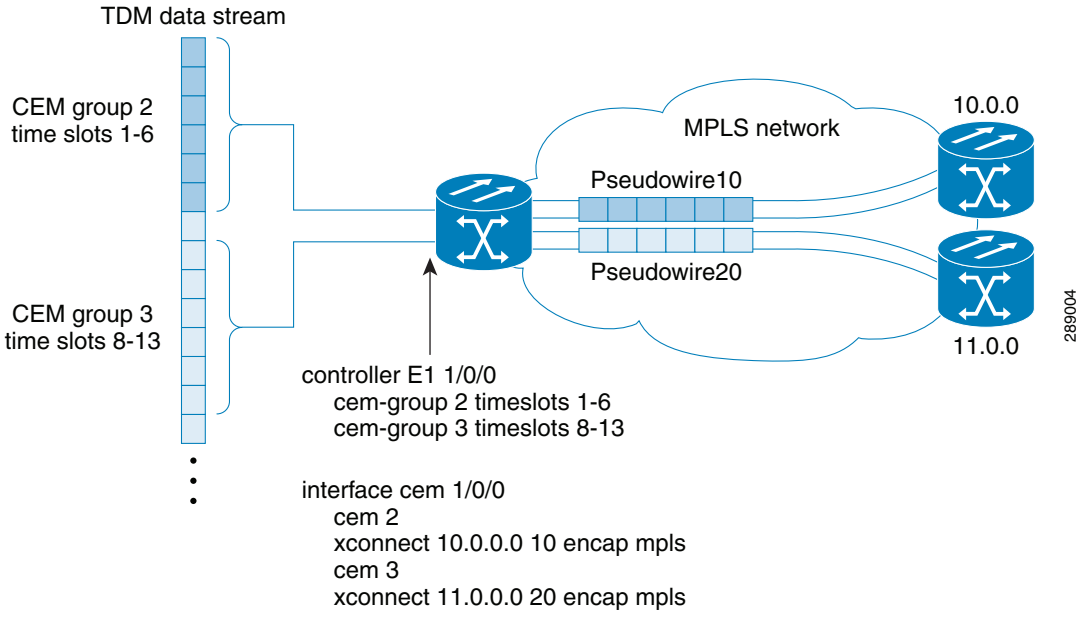
This section covers prerequisites and limitations you should be aware of before configuring RAN backhaul services in Prime Fulfillment.

To create CEM TDM policies and service requests, you must first define the service-related elements Prime Fulfillment, such as target devices and network links. Normally, you create these elements once. For some coverage of these tasks, see Setting Up the Prime Fulfillment Services, page 3-5. Also see other chapters of this guide for how to perform basic infrastructure set up and discovery tasks. The information in this chapter assumes you have already performed these preliminary tasks.

Working with CEM TDM Services

Circuit emulation is configured on a circuit emulation over packet (CEoP) shared port adaptor (SPA) to encapsulate time-division multiplexing (TDM) data in MPLS packets. It then sends the data over a CEM pseudowire to the remote provider edge (PE) router. An example topology is shown in Figure 4-3.

Figure 4-3 Example Circuit Emulation (CEM) Topology



Note the following points about this example:

- A TDM circuit is connected to port 0 on an SPA installed in slot 1, subslot 0 (E1 controller 1/0/0).
- Two pseudowires (PW10 and PW20) are configured to carry TDM data across the MPLS network.
- Two CEM groups (2 and 3) are configured for the data in the TDM time slots.
  - Time slots 1 through 6 are sent over pseudowire 10 to the remote PE router at 10.0.0.0.
  - Time slots 8 through 13 are sent to PE router 11.0.0.0 over pseudowire 20.

The following transport mechanisms are supported:

- SAToP PWE3—Structure Agnostic TDM over Packet / Pseudowire Edge-to-Edge
- CESoPSN PWE3—Circuit Emulation Service over Packet Switched Network / Pseudowire Edge-to-Edge

This rest of this section covers all of the tasks you need to perform to implement and manage CEM TDM services to support RAN backhaul in Prime Fulfillment. It contains the following subsections:

- Working with CEM Classes, page 4-4
- Creating a CEM TDM Policy, page 4-7
- Using Template Variables in CEM TDM Services, page 4-10
- Managing CEM TDM Service Requests, page 4-11

Working with CEM Classes

A CEM class object is used to configure CEM interface parameters so that they can be applied to a group of CEM interfaces. The CEM class can then be selected for use in a CEM TDM policy or service request. The CEM class object is used to configure the **cem class** command and its associated configuration settings on the devices configured by the service.

Note CEM TDM policies and service requests can also use pseudowire classes. Information about creating and managing pseudowire classes is covered in another section of this guide. Fore more information, see Creating and Modifying Pseudowire Classes, page 3-14.

This section covers the following topics:

Creating a CEM Class Object

Perform the following steps to create a CEM class.

Step 1 From the top-level menu in the Prime Fulfillment GUI, choose **Inventory > Logical Inventory > CEM class**.

The CEM Class window appears.

Step 2 Click **Create**.

The Create CEM Class window appears.

Step 3 Enter appropriates values into the fields of the window as follows:

- **Name**—Name for the CEM class object. This field is mandatory.
- **Description**—A description for the CEM class. This is optional.
- **Dejitter Buffer**—The size of the dejitter buffer used for network jitter in CEM configuration mode. The range is 1 to 500 milliseconds. This value is optional.
- **Payload Size**—The payload size used in CEM configuration mode. The range is 32 to 1312 bytes. This value is optional.
- **Idle Pattern**—The pattern of data used to replace the content of each lost CESoPSN data packet. The range is from 0x00 to 0xFF, in hexadecimal. The default pattern is 0xFF.

Step 4 Click **Save** to create the CEM class.

If the create operation is successful, a confirmation message appears, and the CEM Class window reappears showing the new CEM class in the Class Name column.

Editing a CEM Class Object

Perform the following steps to edit a CEM class.

Step 1 From the top-level menu in the Prime Fulfillment GUI, choose **Inventory > Logical Inventory > CEM class**.

The CEM Class window appears showing any CEM classes already created in Prime Fulfillment.

Step 2 Check the check box for the CEM class you would like to edit.

Step 3 Click the **Edit** button in the lower right of the window.

The Edit CEM Class window appears.

Step 4 Make changes to the attribute values as desired.

Step 5 Click the **Save** button to save the changes.

If the edit is successful, a confirmation message is given, and the CEM Class window reappears.

Usage notes for editing CEM class objects:

- The name of a CEM class cannot be changed after it has been created. Therefore, the Name field cannot be modified when editing a CEM class. All other fields are editable.
- When you edit a CEM class that is being used by a service request, that particular service request is subsumed. When multiple service requests use the edited CEM class, all of service requests are subsumed. “Subsumed” means that the service request goes to the Requested state and is ready for deployment.
- When any of the attributes are changed in a CEM class that is associated with one or more CEM TDM service requests, then all of the associated or affected service requests will be subsumed. A window appears in the GUI that shows the list of affected service requests. From the list of service requests, you can perform either of the following actions:
  - Click on the **Save** button to save the service request for a later deployment.
  - Click on the **Save and Deploy** button to save the service request. The service request goes to Requested state and is ready for deployment.

Deleting a CEM Class Object

Perform the following steps to delete a CEM class.

Step 1 From the top-level menu in the Prime Fulfillment GUI, choose **Inventory > Logical Inventory > CEM class**.

The CEM Class window appears showing any CEM classes already created in Prime Fulfillment.

Step 2 Check the check box for the CEM class you would like to delete.

Step 3 Click the **Delete** button in the lower right of the window.

A Confirm Delete window appears.

Step 4 Click the **Delete** button to confirm the deletion.

If the delete operation is successful, a confirmation message appears, and the CEM Class window reappears with the deleted CEM class removed from the Class Name column.

Usage notes for deleting CEM class objects:

- CEM classes in use with CEM TDM policies or service requests cannot be deleted.

Sample Configlets for CEM Classes

The following is a sample configlet generated to create a CEM class:

```
class cem ranCemClass
 payload-size 512
 dejitter-buffer 10
 idle-pattern 0x55
 !
```

The following is a sample configlet showing how a CEM class is included in a configuration:

```
interface cem 0/0
 no ip address
 cem 0
   cem class mycemclass
   xconnect 10.10.10.10 200 encapsulation mpls
   !
 !
```

Creating a CEM TDM Policy

This section describes how to create a CEM TDM policy.

You must define a CEM TDM policy before you can provision a service. A policy can be shared by one or more service requests that have similar service requirements. A policy is a template of most of the parameters needed to define a the service request. After you define the policy, it can be used by all the service requests that share a common set of characteristics. You create a new CEM TDM policy whenever you create a new type of service or a service with different parameters.

You can also associate Prime Fulfillment templates and data files with a policy. See Using Templates with Policies, page 9-21 for more about using templates and data files in policies.

It is also possible to create user-defined attributes within a policy (and service requests based on the policy). For background information on how to use the additional information feature, see Appendix H, "Adding Additional Information to Services."

To start defining a CEM TDM policy, perform the following steps.

Step 1 Choose **Service Design > Policies > Policy Manager**.

The Policy Manager window appears.

Step 2 Click **Create**.

The Policy Editor window appears.

Step 3 Choose **EVC** from the Policy Type drop-down list.

The Policy Editor window appears.

Step 4 Enter a **Policy Name** for the EVC policy.

Step 5 Choose the **Policy Owner** for the EVC policy.

There are three types of EVC policy ownership:

- Customer ownership
- Provider ownership
- Global ownership—Any service operator can make use of this policy.

This ownership has relevance when the Prime Fulfillment Role-Based Access Control (RBAC) comes into play. For example, an EVC policy that is customer-owned can only be seen by operators who are allowed to work on this customer-owned policy. Similarly, operators who are allowed to work on a provider's network can view, use, and deploy a particular provider-owned policy.

Step 6 Click **Select** to choose the owner of the EVC policy.

The policy owner was established when you created customers or providers during Prime Fulfillment setup. If the ownership is global, the Select function does not appear.

Step 7 Choose the **Circuit-Emulation-TDM** as the **Policy Type**.

Step 8 Click **Next**.

The Policy Editor window appears.

Step 9 Continue with the steps contained in the next section, Setting the Service Options, page 4-8.

Setting the Service Options

To set the service options for the CEM TDM policy, perform the following steps.

Note The MPLS Core Connectivity attributes set to PSEUDOWIRE by default and cannot be changed.

Step 1 Choose one of the **TDM CEM Service Options** from the drop-down list.

The choices are:

- **SAToP_UNFRAMED**—Structure-agnostic TDM over packet. This mode used to encapsulate T1 or E1 unstructured (unchannelized) services over packet-switched networks. In SAToP mode, bytes are sent out as they arrive on the TDM line. Bytes do not have to be aligned with any framing. In this mode, the interface is considered as a continuous framed bit stream. All signaling is carried transparently as a part of a bit stream.
- **CESoPN_TIMESLOT**—Circuit emulation services over packet-switched network. This mode is used to encapsulate T1or E1 structured (channelized) services over PSN. CESoPN identifies framing and sends only payload, which can be channelized T1s within DS3, and DS0s within T1. DS0s can be bundled into the same packet.

Step 2 Choose the **CEM Container Type** from the drop-down list.

The choices are:

- **T1**—T-1 digital circuit. Transmits voice/data over the PSTN network at 1.544 Mbps using the DS-1 (Digital Signalling level 1) signaling format.
- **E1**—E-1 digital circuit. Transmits 30 64Kbps digital channels (DS0) for voice or data calls, plus a 64Kbps channel for signaling, and a 64Kbps channel for framing and maintenance.

Step 3 Click **Next**.

The Policy Editor window appears.

Step 4 Continue with the steps contained in the next section, Setting the Service Attributes, page 4-8.

Setting the Service Attributes

To set the service attributes for the CEM TDM policy, perform the following steps.

Step 1 Check the **Enable PseudoWire Redundancy** check box to enable pseudowire redundancy (alternative termination device) under certain conditions.

See Appendix E, "Terminating an Access Ring on Two N-PEs" and, specifically, the section Using N-PE Redundancy in FlexUNI/EVC Service Requests, page E-3, for notes on how this option can be used.

Step 2 Check the **AutoPick VC ID** check box to have Prime Fulfillment autopick the VC ID during service request creation.

If this check box is unchecked, the operator will be prompted to specify a VC ID during service request creation.

Usage notes:

- When AutoPick VC ID is checked, Prime Fulfillment allocates a VC ID for pseudowires from the Prime Fulfillment-managed VC ID resource pool.

Step 3 Click **Next**.

The Policy Editor window appears.

Step 4 Continue with the steps contained in the next section, Using Pseudowire and CEM Classes, page 4-9.

Using Pseudowire and CEM Classes

To specify a pseudowire or CEM class to be used by the CEM TDM policy, perform the following steps.

Step 1 Check the **Use PseudoWireClass** check box to enable the selection of a pseudowire class.

This attribute is unchecked by default.

Usage notes:

- The pseudowire class name is used for provisioning **pw-class** commands on IOS XR devices. See Creating and Modifying Pseudowire Classes, page 3-14 for additional information on pseudowire class support.
- If **Use PseudoWireClass** is checked, an additional attribute, **PseudoWireClass**, appears in the GUI. Click the **Select** button to choose a pseudowire class previously created in Prime Fulfillment.
- Use PseudoWireClass is applicable only for IOS devices.

Step 2 Check the **Use CEM Class** check box to enable the selection of a CEM class.

This attribute is unchecked by default.

Usage notes:

- The CEM class is used for provisioning **cem class ranCemClass** commands on IOS devices. See Working with CEM Classes, page 4-4 for additional information on CEM class support.
- If **Use CEM Class** is checked, an additional attribute, **CEM Class**, appears in the GUI. Click the **Select** button to choose a CEM class that was previously created in Prime Fulfillment.
- Use CEM Class is only applicable for IOS devices.

Step 3 Click **Next**.

The Policy Editor window appears.

Step 4 Continue with the steps contained in the next section, Adding User-Defined Fields into the CEM TDM Policy Workflow, page 4-10.

Adding User-Defined Fields into the CEM TDM Policy Workflow

The Additional Information window allows you to create user-defined attributes within the policy (and service requests based on the policy). For information on how to use the additional information feature, see Appendix H, "Adding Additional Information to Services."

Continue with the steps contained in the next section, Enabling Template Association, page 4-10.

Enabling Template Association

The Prime Fulfillment template feature gives you a means to download free-format CLIs to a device. If you enable templates, you can create templates and data files to download commands that are not currently supported by Prime Fulfillment.

Note Template variable support is available for CEM TDM services. An example template and data file is available containing the CEM-related variables. See the next section Using Template Variables in CEM TDM Services, page 4-10, for how to access and use this template.

Step 1 To enable template association for the policy, click the **Next** button in the Interface Attribute window (before clicking **Finish**).

The Template Association window appears. In this window, you can enable template support and, optionally, associate templates and data files with the policy. For instructions about associating templates with policies and how to use the features in this window, see Using Templates with Policies, page 9-21.

Step 2 When you have completed setting up templates and data files for the policy, click **Finish** in the Template Association window to close it and return to the Policy Editor window.

Step 3 To save the CEM TDM policy, click **Finish**.

To create a service request based on an CDM TEM policy, see Managing CEM TDM Service Requests, page 4-11.

Using Template Variables in CEM TDM Services

This section describes how to access and use the example CEM template in Prime Fulfillment.

To create a data file for the example CEM template, perform the following steps.

Step 1 In the Prime Fulfillment GUI, choose **Service Design > Templates > Template Manager**.

The Template Manager window appears.

Step 2 In the **Templates** window, click on the root folder to expand it.

A list of subfolders appears, with the Examples folder on top.

Step 3 Click the Examples folder to expand it.

Several sample templates are visible, including the CEM template.

Step 4 Click on the CEM folder to choose it.

The CEM template shows in the Template window, along with a pre-loaded CEMProvisioning data file in the Data File Name column of the table.

Step 5 Either click the **Edit** button to edit the CEMProvisioning data file or else uncheck it and click **Create** Data File to create and new one.

In either case, the Data File Editor window appears. You can use this file to map the template variables required for provisioning CEM TDM services.

Step 6 When you have made the desired changes to the templates variables, click **Save** to save the changes.

Step 7 Click **Close** to close the Data File Editor window.

Managing CEM TDM Service Requests

This section describes the various tasks of the workflow for managing CEM TDM service requests. It contains the following sections:

Creating a CEM TDM Service Request

To begin creating the CEM TDM service request, perform the following steps.

Step 1 Choose **Operate > Service Requests > Service Request Manager**.

The Service Request Manager window appears.

Step 2 Click **Create**.

The Service Request Editor window appears.

Step 3 From the Policy drop-down list, choose an CEM TDM policy from the policies previously created (see Creating a CEM TDM Policy, page 4-7). This will be a policy of type EVC, as noted by (EVC) following the policy name.

The EVC Service Request editor window appears. This the first window of the workflow in which you can add and modify attributes for the service request. The new service request inherits all the properties of the chosen policy, such as all the editable and non-editable features and pre-set parameters.

Step 4 Continue with the steps contained in the next section, Setting the Service Request Details, page 4-12.

Setting the Service Request Details

To set the attributes in the Service Request Details section. perform the following steps.

Note The **Job ID** and **SR ID** fields are read-only. When the service request is being created for the first time, the fields display a value of NEW. When an existing service request is being modified, the values of the fields indicate the respective IDs that the Prime Fulfillment database holds within the editing flow of the service request.

Note The **Policy Name** field is read-only. It displays the name of the policy on which the service request is based. Clicking on the read-only policy name displays a list of all the attribute values set within the policy.

Step 1 Check the **AutoPick VC ID** check box if you want Prime Fulfillment to choose a VC ID.

If you do not check this check box, you will be prompted to provide the ID in the VC ID field, as covered in the next step.

When AutoPick VC ID is checked, Prime Fulfillment allocates a VC ID for pseudowires from the Prime Fulfillment-managed VC ID resource pool. In this case, the text field for the VC ID option is non-editable.

Step 2 If AutoPick VC ID was unchecked, enter a VC ID in the **VC ID** field.

Usage notes:

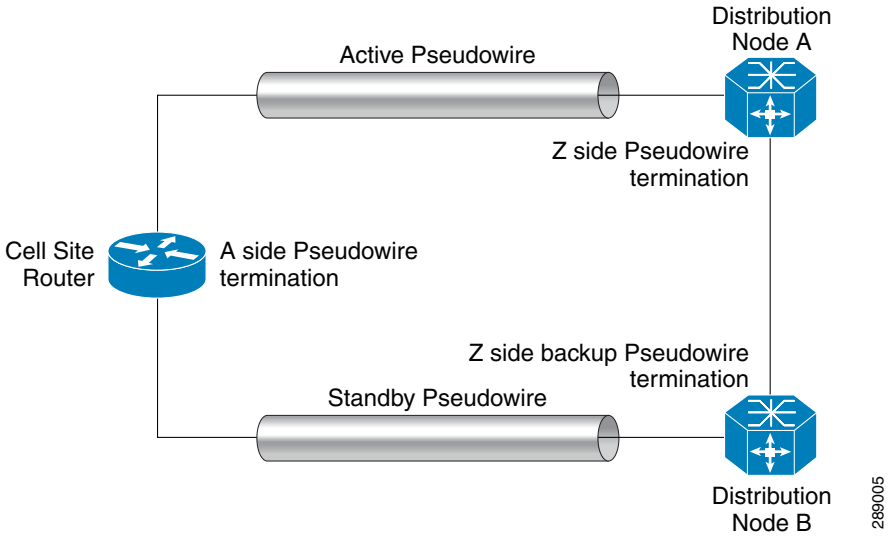
- The VC ID value must be an integer value corresponding to a VC ID.
- When a VC ID is manually allocated, Prime Fulfillment verifies the VC ID to see if it lies within Prime Fulfillment's VC ID pool. If the VC ID is in the pool but not allocated, the VC ID is allocated to the service request. If the VC ID is in the pool and is already in use, Prime Fulfillment prompts you to allocate a different VC ID. If the VC ID lies outside of the Prime Fulfillment VC ID pool, Prime Fulfillment does not perform any verification about whether or not the VC ID allocated. The operator must ensure the VC ID is available.
- The VC ID can be entered only while creating a service request. If you are editing the service request, the VC ID field is not editable.

Step 3 Check the **PseudoWire Redundancy** check box to enable pseudowire redundancy (alternative termination device) under certain conditions.

Usage notes:

- When PseudoWire Redundancy is unchecked, pseudowire redundancy is not provisioned in the service request. Therefore, there will be only two devices actively contributing to the service. See Figure 4-4 for an example configuration. One device is the "A" side of the pseudowire and one side is the "Z" side of the pseudowire. In this case, you would not be able to enter a Backup PW VC ID**.**

Figure 4-4 Pseudowire Termination Example



- When PseudoWire Redundancy check box is enabled there will be three devices actively contributing to the service. One device will be on the "A" side of pseudowire, and the other device will be on the "Z" side. In this case, you could configure the "Z" backup pseudowire using the Backup PW VC ID attribute.
- See Appendix E, "Terminating an Access Ring on Two N-PEs" and, specifically, the section Using N-PE Redundancy in FlexUNI/EVC Service Requests, page E-3, for notes on how this option can be used.

Step 4 If appropriate for the configuration, enter a VC ID for the backup pseudowire in the **Backup PW VC ID** field.

The backup VC ID behaves the same as the VC ID of the primary pseudowire.

Step 5 Choose the **CEM Container Type** from the drop-down list.

The choices are:

- **T1**—T-1 digital circuit. Transmits voice/data over the PSTN network at 1.544 Mbps using the DS-1 (Digital Signalling level 1) signaling format
- **E1**—E-1 digital circuit. Transmits 30 64Kbps digital channels (DS0) for voice or data calls, plus a 64Kbps channel for signaling, and a 64Kbps channel for framing and maintenance.

Usage notes:

- If the CEM Container Type is set to T1, the Framing Type attribute dynamically appears in the GUI, which can be set as covered in the next step.

Step 6 Choose the **Framing Type** from the drop-down list.

The choices are:

- **SDH**—Synchronous Digital Hierarchy.
- **SONET**—Synchronous Optical Networking.

These are related standards for synchronous data transmission over fiber optic networks. Details of these protocols are not covered in this user guide.

Step 7 Check the **Use CEM Class** check box to enable the selection of a CEM class object.

Usage notes:

- The CEM class is editable at the service request level. Therefore the CEM class can be modified from the one set in the policy for the service request. If the CEM class is not changed, the one specified in the policy will be retained for service provisioning.
- The CEM class is used for provisioning **cem class ranCemClass** commands on IOS devices. See Working with CEM Classes, page 4-4 for additional information on CEM class support.
- If **Use CEM Class** is checked, an additional attribute, **CEM Class**, appears in the GUI. Click the **Select** button to choose a CEM class previously created in Prime Fulfillment.
- Use CEM Class is only applicable for IOS devices.

Step 8 Continue with the steps contained in the next section, Selecting Devices, page 4-14.

Selecting Devices

The Select Devices section of the EVC Service Request Editor window allows you to set up links to the N-PE. In Prime Fulfillment, devices added for CEM TDM provisioning are considered as N-PE role-based devices. After the device is selected, you choose controllers and set other attributes for the devices.

The configuration example shown in Figure 4-4 is also used in this section.

Perform the following steps.

Step 1 Click the **Select Device** link to choose the "A" side pseudowire termination point.

The Select PE Device window appears.

Note The device types supported at the "A" node include MWR 2941-DC and 760X series devices having appropriate CEoP and SPA line cards.

Step 2 Choose the appropriate device and click **Save**.

Step 3 In the Controller column, choose the desired controllers from the drop-down list for the device.

Usage notes:

- The controllers that display in the drop-down list depend on the value of in the CEM Container Type attribute specified above.
- If the CEM Container Type is TI, only T1 controllers are populated in the list. If the container type is E1, only E1 controllers appear in the list.
- If there are no controllers for the given type on the selected device, the drop-down list will be empty.
- Also, if CEM Container Type is TI, the value of the addition Framing Type attribute changes the list of controllers. For example, if the Framing Type is SONET, then SONET controllers are displayed in the controller list. Then selecting a SONET controller from the list and clicking on Edit opens the SONET controller attributes window. If the Framing Type is SDH, then selecting a SONET controller from the list and clicking Edit opens the SDH controller attributes window.

Step 4 After selecting the controller for the "A" side termination device, click the **Edit** link in the Link Attributes column to set the controller attributes.

The EVC Service Request Editor - Standard UNI Details window appears. This displays a list of either T1/E1 controller attributes,

Step 5 Set T1/E1 controller attributes for the "A" side terminal device:

- **CEM Group ID**—The CEM Group ID under the controller creates a CEM interface that has the same slot/subslot/port information as the controller. The number it can take depends on the E1 or T1 line. A nuumber from 0 to 23.
- **Clock Source**—INTERNAL or LINE. The default is INTERNAL.
- **Time-Slot Range**—A value from 1 to 31 for T1 controllers, or from 1 to 24 for E1 controllers.

Note Note that the Time-Slot Range attribute only appears if the TDM CEM Service Options attribute in the policy was set to CESoPN_TIMESLOT. It does not appear if the attribute was set to SAToP_UNFRAMED.

- **Use PseudoWireClass**—Check the check box to associate an existing pseudowire class with the service request. A Select button appears in the GUI, which you can use to choose a pseudowire class. Uncheck the check box to dissociate the pseudowire class from the service request.
- **Use Backup PseudoWireClass**—(This attribute is only available when the Pseudowire Redundancy attribute is checked.) Check the check box to associate an existing pseudowire class as a backup pseudowire class with the service request. A Select button appears in the GUI, which you can use to choose a backup pseudowire class. Uncheck the check box to dissociate the pseudowire class from the service request. The functionality is similar to Pseudowire Class selection in the service request window. The Use Backup PseudowireClass attribute is only applicable for "A" terminals and not for "Z" and "Z - Backup" terminals.

Step 6 After setting the attributes for the T1/E1 controllers for the "A" terminal device, click **OK**.

The EVC Service Request Editor window reappears.

Step 7 Select the "Z" and, if applicable, the "Z - Backup" terminal devices and their controllers following the same steps you performed for the "A" terminal device.

SONET controllers are populated in the Controller drop-down list for "Z" and "Z - Backup" terminal devices.

Step 8 After selecting the controllers for these termination devices, click the **Edit** link in the Link Attributes column to set the controller attributes.

The Standard UNI Details window appears, displaying SONET controller attributes,

Step 9 Set the SONET controller attributes.

The SONET attributes that display in this window depend on the CEM Container Type, SONET controller framing type, administrative unit group (AUG) mapping, and channelization mode. This is summarized in Table 4-1.

Table 4-1 CEM Container Type and SONET Controller Attributes

CEM Container Type	SONET Controller Framing Sequence	AUG Mapping	Channelization Mode
E1	SDH	Au-4	C-12
T1	SDH	Au-3	C-11
T1	SONET	N/A	STS-1

The superset of possible attributes is provided below for reference. What actually appears in the GUI depends on the selections previously made in the GUI.

- **CEM Group ID**—The CEM Group ID under the controller creates a CEM interface that has the same slot/subslot/port information as the controller. The number it can take depends on the E1 or T1 line. A number from 0 to 23.
- **Clock Source**—INTERNAL or LINE. The default is INTERNAL.
- **AUG-Mapping**—Configures administrative unit group (AUG) mapping when SDH framing is selected. au-3 or au-4.
- **Channelization Mode**— Mode used to specify TDM Channelization. c-11, c-12, or sts-1.
- **au3 Number**—A number in the range from 1 to 3. This is used to configure a particular Administrative Unit type 3 (AU-3) of an E1 line that has been mapped to an AU-.3.
- **sts-1 Number**—A number user to identify a Synchronous Transport Signal. A number from 1 to 3.
- **sts-1 Mode**—Synchronous Transport Signal. It specifies VT-15 as the STS-1 mode of operation.
- **tug-2 Number**—Tributary Unit group type 2 (TUG-2). A number, or range of numbers, from 1 to 7. To specify a range of TUG-2 numbers use a dash between the values, for example 1-5. An individual TUG-2 can be specified using a comma between values, for example 2,4. The user must set the value in the text box. There is no default value.
- **tug-3 Number**—Tributary Unit group type 3 (TUG-3). A number, or range of numbers, from 1 to 7.
- **VTG Number**—Virtual tributary group carrying a T1. A number, or range of numbers, from 1 to 7.
- **T1 Line Number**—Specifies the T1 number for which service needs to be configured. A number from 1 to 4.
- **E1 Number**—Specifies the E1 number for which a service needs to be configured. A number from 1 to 3.
- **Time Slot**—A number from 1 to 24, or 1 to 31, depending on the container type (E1 or T1)
- **Time-Slot Range**—A number from 1 to 31 for T1 controllers, or from 1 to 24 for E1 controllers.

Note Note that the Time-Slot Range attribute only appears if the TDM CEM Service Options attribute in the policy was set to CESoPN_TIMESLOT. It does not appear if the attribute was set to SAToP_UNFRAMED.

- **Use PseudoWireClass**—Check the check box to associate an existing pseudowire class with the service request. A Select button appears in the GUI, which you can use to choose a pseudowire class. Uncheck the check box to dissociate the pseudowire class from the service request.

Step 10 After the SONET controller values are set, click **OK**.

The EVC Service Request Editor window appears.

Step 11 If desired, use the **Swap Terminals** drop-down list to reorder the devices in relation to the terminals.

The choices are based on the configuration:

- **Swap A - Z**
- **Swap A - Z** Backup
- **Swap Z- Z** Backup

Choose one of the options to perform the swap operation. The devices reorder in the Select Devices column based on the selection.

Usage notes:

- The Swap Terminals button only appears when you first create the service request. If you later edit the service request, the button does not appear and you cannot perform the swap operation at that time.
- The Swap A - Z Backup and Swap Z - Z Backup options are available only when the Pseudowire Redundancy attribute is checked.
- When devices and terminals are swapped, the controllers must be reset in the Controller column.

Step 12 When you have completed setting the attributes in the EVC Service Request Editor window, click the **Save** button at the bottom of the window to save the settings and create the service request.

If any attributes are missing or incorrectly set, Prime Fulfillment displays a warning. Make any corrections or updates needed (based on the information provided by Prime Fulfillment), and click the **Save** button.

For information on modifying an EVC service request see the section Modifying the CEM TDM Service Request, page 4-17. For additional information about saving an CEM TDM service request, see Saving the CEM TDM Service Request, page 4-18.

Modifying the CEM TDM Service Request

You can modify a CEM TDM service request if you must change or modify the links or other settings of the service request.

To modify a service request, perform the following steps.

Step 1 Choose **Operate > Service Requests > Service Request Manager**.

The Service Request Manager window appears, showing service requests available in Prime Fulfillment.

Step 2 Check a check box for a service request.

Step 3 Click **Edit**.

EVC Service Request Editor window appears.

Step 4 Modify any of the attributes, as desired.

Step 5 To add a template/data file to an attachment circuit, see the section Using Templates and Data Files with an CEM TDM Service Request, page 4-17.

Step 6 When you are finished editing the CEM TDM service request, click **Save**.

For additional information about saving an CEM TDM service request, see Saving the CEM TDM Service Request, page 4-18.

Using Templates and Data Files with an CEM TDM Service Request

Prime Fulfillment does not support configuration of all the available CLI commands on a device being managed by the application. In order to configure such commands on the devices, you can use Prime Fulfillment Template Manager functionality. Templates can be associated at the policy level on a per-device role basis. Templates can be overridden at service request level, if the policy-level setting permits the operator to do so.

To associate templates and data files in a service request select any link in the Service Request Editor window and click the **Template** button at the bottom of the window.

Note If the template feature has not been enabled in the associated policy then the Template button will not be available for selection.

The SR Template Association window appears. In this window, you can associate templates at a per-device level. The SR Template Association window lists the devices comprising the link, the device roles, and the template(s)/data file(s) associated with the devices. In this case, the template(s)/data file(s) have not yet been set up.

For further instructions on how to associate templates and data files with a service request, see Using Templates with Service Requests, page 9-24.

Saving the CEM TDM Service Request

To save an CEM TDM service request, perform the following steps.

Step 1 When you have finished setting the attributes for the service request, click **Save** to create the service request.

If the service request is successfully created, the Service Request Manager window appears. The newly created CEM TDM service request is added with the state of REQUESTED.

If, however, the service request creation fails for some reason (for example, a value chosen is out of bounds), you are warned with an error message. In such a case, you should correct the error and save the service request again.

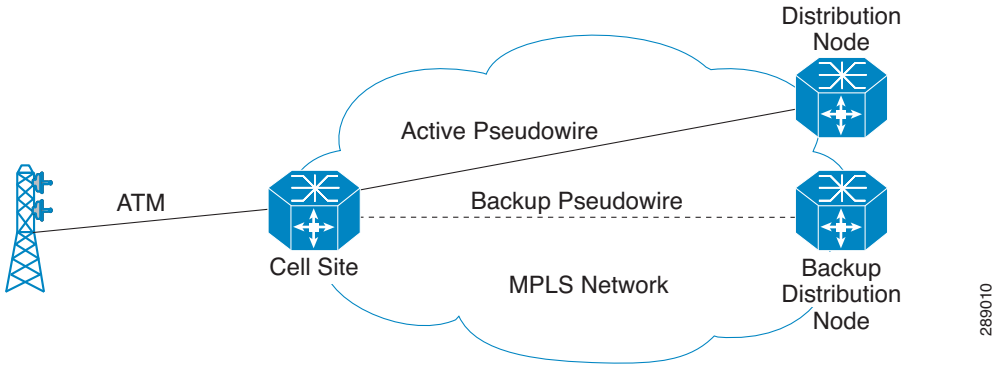
Step 2 If you are ready to deploy the CEM TDM service request, see Deploying Service Requests, page 8-10.

For sample configlets for CEM TDM services, see the section Sample Configlets for RAN Backhaul Services, page 4-33.

Working with ATM Services

RAN backhaul services can be configured on an inverse multiplexing for ATM (ATM/IMA) virtual channel connection (VCC) or permanent virtual path (PVP) circuit. Data is sent over an ATM pseudowire to the remote provider edge (PE) router. When creating pseudowire with an ATM endpoint, you can select IMA interfaces under which to create the permanent virtual circuit (PVC). Also, you can create a controller, which allows you to create the corresponding IMA interface. An example topology is shown in Figure 4-5.

Figure 4-5 Example ATM Topology



The following transport mechanisms are supported:

- ATM IMA VCC PWE3—ATM Inverse Multiplexing for ATM / Virtual Channel Connection / Pseudowire Edge-to-Edge.
- ATM IMA PVP PWE3—ATM Inverse Multiplexing for ATM / Permanent Virtual Path / Pseudowire Edge-to-Edge.

This section covers the various tasks in the workflow for managing ATM services to support RAN backhaul in Prime Fulfillment. It contains the following subsections:

Working with Pseudowire Classes

A pseudowire class is used to configure various attributes related in a class object. The pseudowire class supports configuration of the encapsulation, transport mode, fallback options, and selection of a traffic engineering tunnel down which the pseudowire can be directed. The pseudowire class is later used in ATM policies and/or service requests.

Note Information about creating and managing pseudowire classes is covered in another section of this guide. See Creating and Modifying Pseudowire Classes, page 3-14.

Creating an ATM Policy

This section describes how to create an ATM policy.

You must define an ATM policy before you can provision a service. A policy can be shared by one or more service requests that have similar service requirements. A policy is a template of most of the parameters needed to define a the service request. After you define the policy, it can be used by all the service requests that share a common set of characteristics. You create a new ATM policy whenever you create a new type of service or a service with different parameters.

You can also associate Prime Fulfillment templates and data files with a policy. See Using Templates with Policies, page 9-21. for more about using templates and data files in policies.

It is also possible to create user-defined attributes within a policy (and service requests based on the policy). For background information on how to use the additional information feature, see Appendix H, "Adding Additional Information to Services."

To start defining an ATM policy, perform the following steps.

Step 1 Choose **Service Design > Policies > Policy Manager**.

The Policy Manager window appears.

Step 2 Click **Create**.

The Policy Editor window appears.

Step 3 Choose **EVC** from the Policy Type drop-down list.

The Policy Editor window appears.

Step 4 Enter a **Policy Name** for the EVC policy.

Step 5 Choose the **Policy Owner** for the EVC policy.

There are three types of EVC policy ownership:

- Customer ownership
- Provider ownership
- Global ownership—Any service operator can make use of this policy.

This ownership has relevance when the Prime Fulfillment Role-Based Access Control (RBAC) comes into play. For example, an EVC policy that is customer-owned can only be seen by operators who are allowed to work on this customer-owned policy. Similarly, operators who are allowed to work on a provider's network can view, use, and deploy a particular provider-owned policy.

Step 6 Click **Select** to choose the owner of the EVC policy.

The policy owner was established when you created customers or providers during Prime Fulfillment setup. If the ownership is global, the Select function does not appear.

Step 7 Choose the **ATM** as the **Policy Type**.

Step 8 Click **Next**.

The Create New EVC Policy window appears.

Step 9 Continue with the steps contained in the next section, Setting the ATM Interface Attributes, page 4-20.

Setting the ATM Interface Attributes

This section describes how to set the ATM Interface attributes for the ATM policy.

To set the ATM interface attributes, perform the following steps.

Step 1 Choose the **Transport Mode** from the drop-down list.

The choices are:

- **VP**—Virtual path mode. This is the default.
- **VC**—Virtual circuit mode. If this option is chosen, an ATM Encapsulation attribute appears in the GUI with a default value of AAL0. The ATM Encapsulation cannot be changed.

Step 2 Click **Next**.

The Policy Editor window appears, displaying the Service Attributes section.

Step 3 Continue with the steps contained in the next section, Setting the Service Attributes, page 4-21.

Setting the Service Attributes

To set the service attributes, perform the following steps.

Step 1 Check the **Enable PseudoWire Redundancy** check box to enable pseudowire redundancy (alternative termination device) under certain conditions.

Step 2 Check the **AutoPick VC ID** check box to have Prime Fulfillment autopick the VC ID during service request creation.

If this check box is unchecked, the operator will be prompted to specify a VC ID during service request creation.

Usage notes:

- When AutoPick VC ID is checked, Prime Fulfillment allocates a VC ID for pseudowires from the Prime Fulfillment-managed VC ID resource pool.

Step 3 Click **Next**.

The Policy Editor window appears, displaying the Pseudowire section.

Step 4 Continue with the steps contained in the next section, Using Pseudowire Classes, page 4-21.

Using Pseudowire Classes

To specify a pseudowire class to be used by the ATM policy, perform the following steps.

Step 1 Check the **Use PseudoWireClass** check box to enable the selection of a pseudowire class.

This attribute is unchecked by default.

Usage notes:

- The pseudowire class name is used for provisioning **pw-class** commands on IOS devices. See Creating and Modifying Pseudowire Classes, page 3-14 for additional information on pseudowire class support for IOS XR devices.
- If **Use PseudoWireClass** is checked, an additional attribute, **PseudoWireClass**, appears in the GUI. Click the **Edit** button to choose a pseudowire class previously created in Prime Fulfillment.
- Use PseudoWireClass is only applicable for IOS devices.

Step 2 Click **Next**.

The Policy Editor window appears.

Step 3 Continue with the steps contained in the next section, Adding User-Defined Fields into the ATM Policy Workflow, page 4-22.

Adding User-Defined Fields into the ATM Policy Workflow

The Additional Information window allows you to create user-defined attributes within the policy (and service requests based on the policy). For information on how to use the additional information feature, see Appendix H, "Adding Additional Information to Services".

Continue with the steps contained in the next section, Enabling Template Association, page 4-22.

Enabling Template Association

The Prime Fulfillment template feature gives you a means to download free-format CLIs to a device. If you enable templates, you can create templates and data files to download commands that are not currently supported by Prime Fulfillment.

Note Template variable support is available for ATM policies and services. An example template and data file is available containing the ATM-related variables. See the section Using Template Variables in ATM Services, page 4-22, for how to access and use this template.

Step 1 To enable template association for the policy, click the **Next** button in the Policy Editor window (before clicking **Finish**).

The Policy Editor window appears, displaying the Template Information section. In this window, you can enable template support and, optionally, associate templates and data files with the policy. For instructions about associating templates with policies and how to use the features in this window, see Using Templates with Policies, page 9-21.

Step 2 When you have completed setting up templates and data files for the policy, click **Finish** in the Template Information window to close it and return to the Policy Editor window.

Step 3 To save the ATM policy, click **Finish**.

To create a service request based on an ATM policy, see Managing an ATM Service Request, page 4-26.

Using Template Variables in ATM Services

This section describes how to access and use the example ATM template in Prime Fulfillment.

To create a data file for the example ATM template, perform the following steps.

Step 1 In the Prime Fulfillment GUI, choose **Service Design > Templates > Template Manager**.

The Template Manager window appears.

Step 2 In the **Templates** window, click on the root folder to expand it.

A list of subfolders appears, with the Examples folder on top.

Step 3 Click the Examples folder to expand it.

Several sample templates are visible, including the ATM template.

Step 4 Click on the ATM folder to choose it.

The ATM template shows in the Template window, along with a pre-loaded ATMData data file in the Data File Name column of the table.

Step 5 Either click the **Edit** button to edit the ATMData data file or else uncheck it and click **Create** Data File to create and new one.

In either case, the Data File Editor window appears. You can use this file to map the template variables required for provisioning ATM services.

Step 6 When you have made the desired changes to the templates variables, click **Save** to save the changes.

Step 7 Click **Close** to close the Data File Editor window.

Creating an ATM/IMA Interface Using Templates

ATM/IMA interfaces are created in the device that needs to be provisioned. If they have not been previously created on the device manually, they can be created through the Device Console by using templates. Once the ATM/IMA interfaces are created in the device through Device Console, you must perform a Config collection task for the device. After the Config collection, the Prime Fulfillment inventory (repository) is populated with the newly created ATM/IMA interfaces. These interfaces can then be used for ATM service provisioning.

Creating Template and Data File and Downloading it to a Device.

Note The steps below are presented at a high-level and assume a basic working-knowledge of using templates and data files in Prime Fulfillment. If you require more detailed information on the steps necessary to create templates and data files, see Chapter 9, "Managing Templates and Data Files."

Perform the following steps.

Step 1 Choose **Service Design > Templates > Template Manager**.

Step 2 In the **Template Manager** tree, click on the Example folder to expand it.

Step 3 Create an IMA template as shown below and Save

Step 4 Click on the **Create Template** button to create the IMA template.

The Template Editor window appears.

Step 5 Enter the following:

- **Template Name** (required)—For example, "IMA MWR 2941," or whatever name you choose.
- **Description** (optional).
- **Body** (required)—Enter the configuration text, Velocity Template Language (VTL) directives, and variables that you want included. An example is:

```
controller $container $slot/$sub-slot
    clock source $option
```

```
ima-group $ima
```

Where:

- **container** is of string type with value either as **E1** or **T1**.
- **slot** and **sub-slot** refer to the respective slot and sub-slot.
- **option** value is of string type with value either **internal** or **line**.
- **ima** value is of interger type with minimum value of 0 and maximum value of 23.

Step 6 Click **Save** to save the template.

Step 7 Create an appropriate data file with values mentioned as defined in the previous step.

Now use the Device Console to select the device and specify the data file as follows.

Step 8 Choose **Inventory > Device Tools > Device Console**.

The Choose Operation window appears.

Step 9 Select **Download Template** and click **Next**.

The Download Template window appears.

Step 10 To add devices, click **Add**.

Step 11 From the resulting Device Selection window, check the check box(es) for each device you want to select.

Step 12 Click **Select**.

You return to the Download Template window with the added devices.

Step 13 Click **Next**.

The window refreshes, allowing you to add device groups.

Step 14 Click Next.

The window refreshes, allowing you to choose a template to download.

Step 15 Click the **Select** button.

The Add/Remove Template window appears.

Step 16 Click **Add** to add templates or **Remove** to remove templates.

When you click **Add** you get a Template Datafile Chooser window with the template choices in the tree. Navigate the folders and subfolders in the tree to find the ATM/IMA template you created previously.

Step 17 .When you have the template you want, click **OK**.

Step 18 Select the data file you created previously and click **Accept**.

You return to the Download Template window, which shows the updated information.

Step 19 Click **Next.**

The Template Summary section appears in the window.

Step 20 Check the check boxes for **Upload Config After Download** and **Retrieve device attributes**.

Checking these check boxes will perform a Collect Config on the device when the template download is submitted. This causes the device configuration with the template additions to be updated in the Prime Fulfillment inventory/repository.

Note You can also run the Collect as a separate task, as covered in the next section Adding ATM/IMA Interfaces to the Inventory, page 4-25.

Step 21 Click **Finish** to submit the download.

You receive a message showing the status.

Step 22 Click **Done**.

Adding ATM/IMA Interfaces to the Inventory

You can separately run a Config collection task for a device in order to populate the inventory with the ATM/IMA interfaces previously downloaded via template. This section describes how to connect to the physical device in the network, collect the device information, and populate the repository.

Perform the following steps.

Step 1 Choose **Operate > Tasks > Task Manager**.

The Choose Operation window appears.

Step 2 Click **Create**.

Step 3 Choose **Collect Config**.

The Create Task window appears.

Tip You might want to change the default **Name** and **Description** for this task, so you can more easily identify it in the task log.

Step 4 Click **Next**.

The Collect Config Task window appears.

Step 5 To choose devices associated to the task, in the Devices panel, click **Select**.

The Select Device window appears.

Step 6 Check to choose the desired device(s), then click **Select**.

The Collect Config Task window reappears.

Step 7 To choose device groups associated to the task, in the Groups panel, click **Select**.

A list of available device groups appears.

Step 8 Check to choose the desired device group(s), then click **Select**.

The Collect Config Task window reappears.

Step 9 Set schedule and task owner, if applicable.

Step 10 Click **Submit**.

The Tasks window appears.

Step 11 Choose your task in the Task Name column, then click **Details** to view more information.

The result of the Collect Config task is that the ATM/IMA interfaces created via the template previously download to the device will be updated in the device configuration in the Prime Fulfillment inventory/repository.

Managing an ATM Service Request

This section describes the various tasks of the workflow involved in managing ATM service requests to support RAN backhaul services. It contains the following sections:

Creating an ATM Service Request

To begin creating the ATM service request, perform the following steps.

Step 1 Choose **Operate > Service Requests > Service Request Manager**.

The Service Request Manager window appears.

Step 2 Click **Create**.

The Service Request Editor window appears.

Step 3 From the Policy drop-down list, choose an ATM policy from the policies previously created (see Creating an ATM Policy, page 4-19). This will be a policy of type EVC, as noted by (EVC) following the policy name.

The EVC Service Request Editor window appears. This is the first window of the workflow, in which you can add and modify attributes for the service request. The new service request inherits all the properties of the chosen policy, such as all the editable and non-editable features and pre-set parameters.

The attributes in this window describe the pseudowire connectivity between the attachment circuits. The pseudowire connectivity allows you to create a point-to-point connection between two customer sites using X-connect (that is, cross connection).

Step 4 Continue with the steps contained in the next section, Setting the Service Request Details, page 4-26.

Setting the Service Request Details

To set the attributes in the Service Request Details section, perform the following steps.

Note The **Job ID** and **SR ID** fields are read-only. When the service request is being created for the first time, the fields display a value of NEW. When an existing service request is being modified, the values of the fields indicate the respective IDs that the Prime Fulfillment database holds within the editing flow of the service request.

Note The **Policy Name** field is read-only. It displays the name of the policy on which the service request is based. Clicking on the read-only policy name displays a list of all the attribute values set within the policy.

Step 1 Check the **AutoPick VC ID** check box if you want Prime Fulfillment to choose a VC ID.

If you do not check this check box, you will be prompted to provide the ID in the VC ID field, as covered in the next step.

When AutoPick VC ID is checked, Prime Fulfillment allocates a VC ID for pseudowires from the Prime Fulfillment-managed VC ID resource pool. In this case, the text field for the VC ID option is non-editable.

Step 2 If AutoPick VC ID was unchecked, enter a VC ID in the **VC ID** field.

Usage notes:

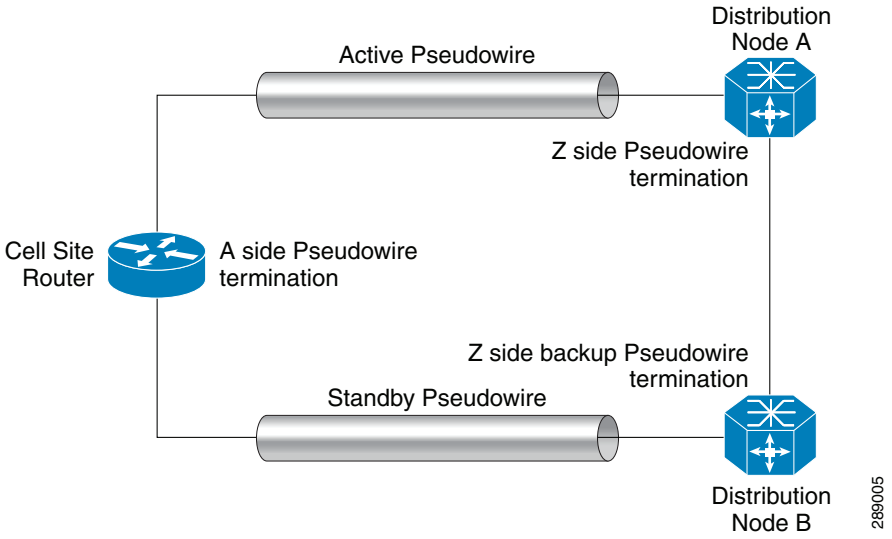
- The VC ID value must be an integer value corresponding to a VC ID.
- When a VC ID is manually allocated, Prime Fulfillment verifies the VC ID to see if it lies within Prime Fulfillment's VC ID pool. If the VC ID is in the pool but not allocated, the VC ID is allocated to the service request. If the VC ID is in the pool and is already in use, Prime Fulfillment prompts you to allocate a different VC ID. If the VC ID lies outside of the Prime Fulfillment VC ID pool, Prime Fulfillment does not perform any verification about whether or not the VC ID allocated. The operator must ensure the VC ID is available.
- The VC ID can be entered only while creating a service request. If you are editing the service request, the VC ID field is not editable.

Step 3 Check the **PseudoWire Redundancy** check box to enable pseudowire redundancy (alternative termination device) under certain conditions.

Usage notes:

- When PseudoWire Redundancy is unchecked, pseudowire redundancy is not provisioned in the service request. Therefore, there will be only two devices actively contributing to the service. See Figure 4-6 for an example configuration. One device is the "A" side of the pseudowire and one side is the "Z" side of the pseudowire. In this case, you would not be able to enter a Backup PW VC ID**.**

Figure 4-6 Pseudowire Termination Example



- When PseudoWire Redundancy check box is enabled there will be three devices actively contributing to the service. One device will be on the "A" side of pseudowire, and the other device will be on the "Z" side. In this case, you could configure the "Z" backup pseudowire using the Backup PW VC ID attribute.
- See Appendix E, "Terminating an Access Ring on Two N-PEs" and, specifically, the section Using N-PE Redundancy in FlexUNI/EVC Service Requests, page E-3, for notes on how this option can be used.

Step 4 If appropriate for the configuration, enter a VC ID for the backup pseudowire in the **Backup PW VC ID** field.

The backup VC ID behaves the same as the VC ID of the primary pseudowire.

Step 5 Continue with the steps contained in the next section, Setting the MCPT Timer Values, page 4-28.

Setting the MCPT Timer Values

The Setting MCPT Timer Values section of the EVC Service Request Editor window allows you to set the values for Martini cell-packing timers. The MCPT timers can be attached to a permanent virtual circuit (PVC) or permanent virtual path (PVP). If the associated MCPT timer expires before the maximum number of cells that can be packed is reached, then the packet is transmitted with the number of cells that have been packed thus far. The MCPT timers are specified when configuring the link attributes for devices used in the ATM service.

Perform the following steps.

Step 1 Enter appropriate values for the MCPT timers:

- **MCPT Timer 1**—Set a value between 500 and 10000 microseconds.
- **MCPT Timer 2**—Set a value between 1000 and 10000 microseconds.
- **MCPT Timer 3**—Set a value between 1500 and 10000 microseconds.

Step 2 Continue with the steps contained in the next section, Selecting Devices, page 4-29.

Selecting Devices

The Select Devices section of the EVC Service Request Editor window allows you to set up links to the N-PE. In Prime Fulfillment, devices added for circuit emulation provisioning are considered as N-PE role-based devices. After the device is selected, the respective ATM or ATM/IMA interfaces are populated in the Interface drop-down list(s).

The configuration example shown in Figure 4-6 is also used in this section.

Perform the following steps.

Step 1 Click the **Select Device** link to choose the "A" side pseudowire termination point.

The Select PE Device window appears.

Step 2 Choose the appropriate device and click **Save**.

Step 3 In the Interfaces column, choose the desired interface from the drop-down list for the device.

Usage notes:

- The interfaces that display in the drop-down list for the "A" side termination point will be ATM or ATM/IMA interfaces.

Step 4 After selecting the interface for the "A" side termination device, click the **Edit** link in the Link Attributes column to set the interface attributes.

The ATM UNI Details window appears. This displays a list of interface attributes,

Step 5 Set interface attributes for the "A" side terminal device.

Note The attributes in the window dynamically change depending whether the value of the Transport Mode attribute is set as VP (PVP service) or VC (PVC service). Refer to the appropriate substep below, depending on your configuration.

a. If the "A" side termination is a PVP service, set the following attributes displayed in the window:

- **Transport Mode**—The PVP transport type. In this case, VP appears in the drop-down list.
- **ATM VPI**—Virtual path identifier. Anumber between 0 and 255.
- **Maximum no. of cells to be packed**—The maximum number of cells to be packed into a packet (cell-packing). A number from 2 to 28.
- **Use MCPT Timer**—The number of MCPT timers to use. 1, 2, or 3.
- **Use PseudoWireClass**—Check the check box to associate an existing pseudowire class with the service request. A Select button appears in the GUI, which you can use to choose a pseudowire class. Uncheck the check box to dissociate the pseudowire class from the service request.
- **Use Backup PseudoWireClass**—(This attribute is only available when the Pseudowire Redundancy attribute is checked.) Check the check box to associate an existing pseudowire class as a backup pseudowire class with the service request. A Select button appears in the GUI, which you can use to choose a backup pseudowire class. Uncheck the check box to dissociate

the pseudowire class from the service request. The functionality is similar to Pseudowire Class selection in the service request window. The Use Backup PseudowireClass attribute is only applicable for "A" terminals and not for "Z" and "Z - Backup" terminals.

b. If the "A" side termination is a PVC service, set the following attributes displayed in the window:

- **Transport Mode**—The PVC transport type. In this case, VC appears in the drop-down list.
- **Sub-Interface #**—Creates the specified point-to-point sub-interface on the given port on the specified ATM SPA. Range for sub-interface is between 1 and 2147483647.
- **ATM VPI**—Virtual path identifier. A number between 0 and 255.
- **ATM VCI**—Virtual circuit identifier. A number between 1 and 65535.
- **Maximum no. of cells to be packed**—The maximum number of cells to be packed into a packet (cell-packing). A number from 2 to 28.
- **Use MCPT Timer**—The number of MCPT timers to use. 1, 2, or 3.
- **Use PseudoWireClass**—Check the check box to associate an existing pseudowire class with the service request. A Select button appears in the GUI, which you can use to choose a pseudowire class. Uncheck the check box to dissociate the pseudowire class from the service request.

Step 6 After setting the attributes for the interfaces for the "A" terminal device, click **OK**.

The EVC Service Request Editor window reappears.

Step 7 Select the "Z" and, if applicable, the "Z - Backup" terminal devices and configure their interfaces following the same steps you performed for the "A" terminal device.

Usage notes:

- ATM interfaces are populated in the Interface drop-down list for "Z" and "Z - Backup" terminal devices.
- When the Pseudowire Redundancy check box is checked in the EVC Service Request Editor window (previously in the workflow), you can select and configure a "Z - Backup" node once the link attributes have been set on for the "A" and "Z" terminal devices.
- As in case of the "A" terminal device, the interface attributes for the "Z" and "Z - Backup" terminal devices will depend on the type of ATM service (PVP or PVC).

Step 8 After selecting the interface for these termination devices, click the **Edit** link in the Link Attributes column to set the interface attributes.

Step 9 If desired, use the **Swap Terminals** drop-down list to reorder the devices in relation to the terminals.

The choices are based on the configuration:

- **Swap A - Z**
- **Swap A - Z** Backup
- **Swap Z- Z** Backup

Choose one of the options to perform the swap operation. The devices reorder in the Select Devices column based on the selection.

Usage notes:

- The Swap Terminals button only appears when you first create the service request. If you later edit the service request, the button does not appear and you cannot perform the swap operation at that time.
- The Swap A - Z Backup and Swap Z - Z Backup options are available only when the Pseudowire Redundancy attribute is checked.

- When devices and terminals are swapped, the interfaces must be reset in the Interfaces column.

Step 10 After the interface attributes are set, click **OK**.

The EVC Service Request Editor window appears.

Step 11 When you have completed setting the attributes in the EVC Service Request Editor window, click the **Save** button at the bottom of the window to save the settings and create the ATM service request.

If any attributes are missing or incorrectly set, Prime Fulfillment displays a warning. Make any corrections or updates needed (based on the information provided by Prime Fulfillment), and click the **Save** button.

For information on modifying an EVC service request see the section Modifying the ATM Service Request, page 4-31. For additional information about saving an ATM service request, see Saving the ATM Service Request, page 4-32.

Modifying the ATM Service Request

You can modify an ATM service request if you must change or modify the links or other settings of the service request.

To modify a service request, perform the following steps.

Step 1 Choose **Operate > Service Requests > Service Request Manager**.

The Service Request Manager window appears, showing service requests available in Prime Fulfillment.

Step 2 Check a check box for a service request.

Step 3 Click **Edit**.

The EVC Service Request Editor window appears.

Step 4 Modify any of the attributes, as desired.

Step 5 To add a template/data file to an attachment circuit, see the section Using Templates and Data Files with an ATM Service Request, page 4-31.

Step 6 When you are finished editing the ATM service request, click **Save**.

For additional information about saving an ATM service request, see Saving the ATM Service Request, page 4-32.

Using Templates and Data Files with an ATM Service Request

Prime Fulfillment does not support configuration of all the available CLI commands on a device being managed by the application. In order to configure such commands on the devices, you can use Prime Fulfillment Template Manager functionality. Templates can be associated at the policy level on a per-device role basis. Templates can be overridden at service request level, if the policy-level setting permits the operator to do so.

To associate templates and data files in a service request select any link in the Service Request Editor window and click the **Template** button at the bottom of the window.

Note If the template feature has not been enabled in the associated policy then the Template button will not be available for selection.

The SR Template Association window appears. In this window, you can associate templates at a per-device level. The SR Template Association window lists the devices comprising the link, the device roles, and the template(s)/data file(s) associated with the devices. In this case, the template(s)/data file(s) have not yet been set up.

For further instructions on how to associate templates and data files with a service request, see Using Templates with Service Requests, page 9-24.

Saving the ATM Service Request

To save an ATM service request, perform the following steps.

Step 1 When you have finished setting the attributes for the service request, click **Save** to create the service request.

If the service request is successfully created, the Service Request Manager window appears. The newly created ATM service request is added with the state of Requested.

If, however, the service request creation fails for some reason (for example, a value chosen is out of bounds), you are warned with an error message. In such a case, you should correct the error and save the service request again.

Step 2 If you are ready to deploy the ATM service request, see Deploying Service Requests, page 8-10.

For sample configlets for ATM services, see the section Sample Configlets for RAN Backhaul Services, page 4-33.

Sample Configlets for RAN Backhaul Services

This section provides sample configlets for RAN backhaul service provisioning in Prime Fulfillment. It contains the following subsections:

Overview

The configlets provided in this section show the CLIs generated by Prime Fulfillment for particular services and features. Each configlet example provides the following information:

- Service
- Feature
- Devices configuration (network role, hardware platform, relationship of the devices and other relevant information)
- Sample configlets for each device in the configuration
- Comments

Note The configlets generated by Prime Fulfillment are only the delta between what needs to be provisioned and what currently exists on the device. This means that if a relevant CLI is already on the device, it does not show up in the associated configlet.

Note The CLIs shown in bold are the most relevant commands.

CEM TDM using SAToP PW3

Configuration

- Service: RAN Backhaul.
- Feature: This sections contains sample configlets that would be generated for CEM TDM SAToP PW3 service on a cell site router and two distribution nodes (A and B).
- Device configuration:
  - The cell site router is an MWR 2941-DC router with an IOS image.

    Controller: E1 0/0

    Interface(s): CEM 0/0
  - Distribution node A is a 760X series device with IOS image.

    Contoller: SONET 3/0/0

    Interface(s): CEM 3/0/0
  - Distribution node B is a 760X series device with IOS image.

    Contoller: SONET 3/0/0

    Interface(s): CEM 3/0/0

Configlets

Cell Site Router

```
pseudowire-class c76a3-1
 encapsulation mpls
!
pseudowire-class c76a3-2
 encapsulation mpls
!
controller E1 0/0
 clock source internal
 cem-group 0 unframed
!
interface CEM0/0
 no ip address
 cem 0
  xconnect 10.0.0.1 2090102001 pw-class c76a3-1
   backup peer 10.0.0.4 2090403001 pw-class c76a3-2
```

Distribution Node A	Distribution Node B

pseudowire-class c76a3-1
 encapsulation mpls
 preferred-path interface Tunnel211
!
controller SONET 3/0/0
 ais-shut
 framing sdh
 clock source line
 aug mapping au-4
 !
 au-4 1 tug-3 2
  mode c-12
 tug-2 1 e1 1 description m29a2-3(CEM0/0)
  tug-2 1 e1 1 cem-group 100 unframed
!
interface CEM3/0/0
 no ip address
 cem 100
  xconnect 10.0.0.1 2090102001 pw-class
c76a3-1 sequencing both
	pseudowire-class c76a3-2
	 encapsulation mpls
	 preferred-path interface Tunnel340
	!
	controller SONET 3/0/0
	 ais-shut
	 framing sdh
	 clock source line
	 aug mapping au-4
	 !
	 au-4 1 tug-3 2
	  mode c-12
	 tug-2 1 e1 1 description m29a2-3(CEM0/0)
	  tug-2 1 e1 1 cem-group 100 unframed
	  tug-2 1 e1 1 framing unframed
	!
	interface CEM3/0/0
	 cem 100
	  xconnect 10.0.0.4 2090403001 pw-class
	c76a3-2 sequencing both

Comments

- None.

CEM TDM using CESoPSN

Configuration

- Service: RAN Backhaul.
- Feature: This sections contains sample configlets that would be generated for CEM TDM CESoPSN service on a cell site router and two distribution nodes (A and B).
- Device configuration:
  - The cell site router is an MWR 2941-DC router with an IOS image.

    Contoller: E1 0/4

    Interface(s): CEM 0/4
  - Distribution node A is a 760X series device with IOS image.

    Contoller: SONET 3/0/0

    Interface(s): CEM 3/0/0
  - Distribution node B is a 760X series device with IOS image.

    Contoller: SONET 3/0/0

    Interface(s): CEM 3/0/0

Configlets

Cell Site Router

```
pseudowire-class c76a3-1
 encapsulation mpls
!
pseudowire-class c76a3-2
 encapsulation mpls
!
controller E1 0/4
 clock source internal
 cem-group 0 timeslots 1-7
!
interface CEM0/4
 cem 0
  xconnect 10.0.0.1 3090102001 pw-class c76a3-1
   backup peer 10.0.0.4 3090403001 pw-class c76a3-2
```

Distribution Node A	Distribution Node B

pseudowire-class c76a3-1
 encapsulation mpls
 preferred-path interface Tunnel211
!
controller SONET 3/0/0
 ais-shut
 framing sdh
 clock source line
 aug mapping au-4
 !
 au-4 1 tug-3 2
  mode c-12
 tug-2 2 e1 2 description m29a2-3(CEM0/4 cem 0)
  tug-2 2 e1 2 cem-group 104 timeslots 1-7
 !
 interface CEM3/0/0
  cem 104
  xconnect 10.0.0.1 3090102001 pw-class c76a3-1 sequencing both
	pseudowire-class c76a3-2
	 encapsulation mpls
	 preferred-path interface Tunnel340
	!
	controller SONET 3/0/0
	 ais-shut
	 framing sdh
	 clock source line
	 aug mapping au-4
	 !
	 au-4 1 tug-3 2
	  mode c-12
	tug-2 2 e1 2 description m29a2-3(CEM0/4 cem 0)
	  tug-2 2 e1 2 cem-group 104 timeslots 1-7
	!
	interface CEM3/0/0
	 cem 104
	  xconnect 10.0.0.4 3090403001 pw-class c76a3-2 sequencing both

Comments

- None.

ATM/IMA PVP Service

Configuration

- Service: RAN Backhaul.
- Feature: This sections contains sample configlets that would be generated for ATM PVP service on a cell site router and two distribution nodes.
- Device configuration:
  - The cell site router is an MWR 2941-DC router with an IOS image.

    Contoller(s): E1 0/12, E1 0/13

    Interface(s): ATM0/IMA2
  - Distribution node A is a 760X series device with IOS image.

    Interface(s): ATM 3/1/1
  - Distribution node B is a 760X series device with IOS image.

    Interface(s): ATM 3/1/1

Configlets

Cell Site Router

```
pseudowire-class c76a3-1
 encapsulation mpls
!
pseudowire-class c76a3-2
 encapsulation mpls
!
controller E1 0/12
 framing NO-CRC4
 clock source internal
 ima-group 2 scrambling-payload
!
controller E1 0/13
 framing NO-CRC4
 clock source internal
 ima-group 2 scrambling-payload
!
interface ATM0/IMA2
 no ip address
 ima version 1.0
 ima group-id 2
 atm mcpt-timers 1000 5000 10000
 atm pvp 9 l2transport
  cell-packing 28 mcpt-timer 3
  xconnect 10.0.0.1 4090102003 pw-class c76a3-1
   backup peer 10.0.0.4 4090403003 pw-class c76a3-2
 no atm ilmi-keepalive
```

Distribution Node Z	Distribution Node Z Backup

pseudowire-class c76a3-1
 encapsulation mpls
 preferred-path interface Tunnel211
!
interface ATM3/1/1
 no ip address
 atm mcpt-timers 1000 5000 10000
 atm pvp 9 l2transport
  cell-packing 28 mcpt-timer 3
  xconnect 10.0.0.1 4090102003 pw-class c76a3-1
 no atm enable-ilmi-trap
	pseudowire-class c76a3-2
	 encapsulation mpls
	 preferred-path interface Tunnel340
	!
	interface ATM3/1/1
	 no ip address
	 atm mcpt-timers 1000 5000 10000
	 atm pvp 9 l2transport
	  cell-packing 28 mcpt-timer 3
	  xconnect 10.0.0.4 4090403003 pw-class c76a3-2
	 no atm enable-ilmi-trap

Comments

- None.

ATM/IMA VCC Service

Configuration

- Service: RAN Backhaul.
- Feature: This sections contains sample configlets that would be generated for ATM VCC service on a cell site router and two distribution nodes.
- Device configuration:
  - The cell site router is an MWR 2941-DC router with an IOS image.

    Contoller(s): E1 0/8, E1 0/9

    Interface(s): ATM0/IMA0, ATM0/ IMA0
  - Distribution node A is a 760X series device with IOS image.

    Interface(s): ATM 3/1/0
  - Distribution node B is a 760X series device with IOS image.

    Interface(s): ATM 3/1/0

Configlets

Cell Site Router

```
pseudowire-class c76a3-1
 encapsulation mpls
!
pseudowire-class c76a3-2
 encapsulation mpls
!
controller E1 0/8
 framing NO-CRC4
 clock source internal
 ima-group 0 scrambling-payload
!
controller E1 0/9
 framing NO-CRC4
 clock source internal
 ima-group 0 scrambling-payload
!
interface ATM0/IMA0
 ima version 1.0
 ima group-id 0
 atm mcpt-timers 1000 5000 10000
!
interface ATM0/IMA0.1 point-to-point
 snmp trap link-status
 pvc 9/34 l2transport
  cbr 255
  encapsulation aal0
  cell-packing 28 mcpt-timer 3
  xconnect 10.0.0.1 4090102001 pw-class c76a3-1
   backup peer 10.0.0.4 4090403001 pw-class c76a3-2
```

Distribution Node Z	Distribution Node Z Backup

pseudowire-class c76a3-1
 encapsulation mpls
 preferred-path interface Tunnel211
!
interface ATM3/1/0
 atm mcpt-timers 1000 5000 10000
!
interface ATM3/1/0.9001 point-to-point
 description m29a2-3 - ATM0/IMA0
 no atm enable-ilmi-trap
 pvc 9/34 l2transport
  cell-packing 28 mcpt-timer 3
  encapsulation aal0
  xconnect 10.0.0.1 4090102001 pw-class c76a3-1
	pseudowire-class c76a3-2
	 encapsulation mpls
	 preferred-path interface Tunnel340
	!
	interface ATM3/1/0
	 atm mcpt-timers 1000 5000 10000
	!
	interface ATM3/1/0.9001 point-to-point
	 description m29a2-3 - ATM0/IMA0
	 no atm enable-ilmi-trap
	 pvc 9/34 l2transport
	  cell-packing 28 mcpt-timer 3
	  encapsulation aal0
	  xconnect 10.0.0.4 4090403001 pw-class c76a3-2

Comments

- None.

CHAPTER 5

Managing MPLS VPN Services

This chapter describes the tasks required to get started using Cisco Prime Fulfillment 6.2, Multiprotocol Label Switching (MPLS) virtual private network (VPN).

Note The information in the section summarizes some of the key tasks required to get started using MPLS VPN. For additional information about setting up basic Prime Fulfillment services, see Setting Up the Prime Fulfillment Services, page 5-4.

This chapter covers the following topics:

Getting Started with MPLS VPN

This section covers the following topics:

Before You Begin

Before you can use MPLS VPN to provision, perform the following steps:

Step 1 Install Prime Fulfillment. See the *Cisco Prime Fulfillment Installation Guide 6.2*.

Step 2 Purchase the license.

Step 3 Assess your network.

For example, the network must meet certain criteria such as MPLS, MP-BGP enabled, PE routers in supported platforms, and so forth. Prime Fulfillment provisions only PE-CEs, not devices within a given network.

Step 4 Populate Prime Fulfillment.

Prime Fulfillment Service Activation

To activate MPLS services you must configure Prime Fulfillment so it "knows" about the preconfiguration information, such as devices, providers, customers, and so on, that Prime Fulfillment is going to manage and their roles. The major steps to achieve Prime Fulfillment service activation include setting up:

- Devices
- Provider information (providers, regions, and PEs)
- Customer information (customers, sites, and CPEs)
- Resource pools:
  - IP addresses
  - Route targets (RTs)
  - Route distinguishers (RDs)
  - Site of origin (SOO)
- Virtual Private Networks (VPNs)
- Customer edge (CE) routing communities (CERCs)
- Named Physical Circuits (NPCs)

Note These steps are covered in more detail in Setting Up the Prime Fulfillment Services, page 5-4

Working with MPLS Policies and Service Requests

After you have set up providers, customers, devices, and resources in Prime Fulfillment, you are ready to create MPLS policies, provision service requests, and deploy the services. After the service requests are deployed you can monitor, audit and run reports on them. All of these tasks are covered in this guide. To accomplish these tasks, perform the following steps:

Step 1 If necessary, review overview information about MPLS concepts.

Step 2 Set up an MPLS policy.

For basic information and key concepts, see MPLS VPN Service Policies, page 5-40 as well as subsequent chapters in this guide.

Step 3 Provision the MPLS service request.

See the appropriate section, depending on the type service request you want to provision:

- Provisioning Management VPN, page 5-120
- MPLS VPN Service Requests, page 5-78
- Provisioning Regular PE-CE Links, page 5-98
- Provisioning Multi-VRFCE PE-CE Links, page 5-109
- Provisioning Management VPN, page 5-120
- Provisioning Cable Services, page 5-129
- Provisioning Carrier Supporting Carrier, page 5-139
- Provisioning Multiple Devices, page 5-143
- Spanning Multiple Autonomous Systems, page 5-153

Step 4 Deploy the MPLS service request.

See MPLS VPN Service Requests, page 5-78

Step 5 Check the status of deployed services.

You can use one or more of the following methods:

- Monitor service requests. See the section Monitoring Service Requests, page 8-11.
- Audit service requests. See the section Deploying, Monitoring, and Auditing Service Requests, page 3-166.
- Run MPLS reports. See Generating MPLS Reports, page 10-40.

Step 6 Troubleshoot MPLS services.

See Troubleshooting MPLS VPNs, page 5-246

For additional information on specific topics, see the following sections of this guide:

- For information about IPv6 and 6VPE support, see IPv6 and 6VPE Support in MPLS VPN, page 5-30.
- For sample configlets generated by Prime Fulfillment for MPLS services, see Sample Configlets, page 5-165
- For information about using templates and data files in Prime Fulfillment policies and service requests, see Chapter 9, “Managing Templates and Data Files.”

Setting Up the Prime Fulfillment Services

This section contains the basic steps to set up the Prime Fulfillment services to support MPLS VPN service policies and service requests.

Note This section presents high-level information on Prime Fulfillment services that are relevant to MPLS VPN. For more detailed information on setting up these and other basic Prime Fulfillment services, see the Chapter 2, "Before Setting Up Prime Fulfillment"and Chapter 8, "Managing Service Requests".

This section covers the following topics:

Overview

To create an MPLS VPN service request, you must create the following infrastructure data:

- Devices

  A Device in Prime Fulfillment is a logical representation of a physical device in the network. You can import devices (configurations) into Prime Fulfillment by using Inventory Manager or the Prime Fulfillment GUI. You can also use the Auto Discovery feature of Inventory Manager to import devices into the Repository.

  To set device attributes, see Setting Up Devices and Device Groups of Chapter 2, "Before Setting Up Prime Fulfillment".

- Import or add raw devices

  Every network element that Prime Fulfillment manages must be defined as a device in the Prime Fulfillment repository. An element is any device from which Prime Fulfillment can collect information. In most cases, devices are Cisco IOS routers and switches. You can set up devices in Prime Fulfillment manually, through autodiscovery, or through importing device configuration files. For detailed steps for importing, adding, and collecting configurations for devices, see Appendix G, "Inventory - Discovery."

- Customers

  A customer is typically an enterprise or large corporation that receives network services from a service provider. A Customer is also a key logical component of Prime Fulfillment.

  - Sites

    A Site is a logical component of Prime Fulfillment that connects a Customer with a CE. It can also represent a physical customer site.

  - CPE/CE Devices

    A CPE is "customer premises equipment," typically a customer edge router (CE). It is also a logical component of Prime Fulfillment. You can create CPE in Prime Fulfillment by associating a device with a Customer Site.

For detailed steps to create customers and sites, see Setting Up Resources, page 2-40 of Chapter 2, "Before Setting Up Prime Fulfillment".

- Providers

  A provider is typically a "service provider" or large corporation that provides network services to a customer. A Provider is also a key logical component of Prime Fulfillment.

  - Regions

    A Region is a logical component of Prime Fulfillment that connects a Provider with a PE. It can also represent a physical provider region.

  - PE Devices

    A PE is a provider edge router or switch. It is also a logical component of Prime Fulfillment. You can create PE in Prime Fulfillment by associating a Device with a Provider Region. In Prime Fulfillment, a PE can be a "point of presence" router (POP) or a Layer 2 switch (CLE).

  To create a provider and a region, see Setting Up Resources, page 2-40 of Chapter 2, "Before Setting Up Prime Fulfillment".

- Access Domains (for Layer 2 Access)

  The Layer 2 Ethernet switching domain that connects a PE to a CE is called an Access Domain. All the switches attached to the PE-POP belong to this Access Domain. These switches belong to the Provider and are defined in Prime Fulfillment as PE-CLE.

  To create a provider and a region, see Setting Up Resources, page 2-40 of Chapter 2, "Before Setting Up Prime Fulfillment".

- Resource Pools
  - IP Addresses
  - Multicast
  - Route Distinguisher
  - Route Target
  - VLANs (for Layer 2 Access)

  To create a provider and a region, see Setting Up Resources, page 2-40 of Chapter 2, "Before Setting Up Prime Fulfillment".

- VPN

  Before creating a Service Policy, a VPN name must be defined within Prime Fulfillment.

- Route Target(s)

  To create a route target, see Setting Up Resources, page 2-40 of Chapter 2, "Before Setting Up Prime Fulfillment".

Setting Up Devices for IOS XR Support

Prime Fulfillment supports provisioning of basic MPLS VPNs on devices running Cisco's IOS XR software. IOS XR, a new member of the Cisco IOS family, is a unique self-healing and self-defending operating system designed for always-on operation while scaling system capacity up to 92Tbps.

Note For information about specific platforms and features supported for IOS XR devices for MPLS VPN, as well as IOS XR versions supported, see the *Release Notes for Cisco Prime Fulfillment 6.2.*

To enable IOS XR support in MPLS VPN, perform the following steps:

Step 1 Set the DCPL property **Provisioning/Service/mpls/platform/CISCO_ROUTER/IosXRConfigType** to XML.

Possible values are **CLI**, **CLI_XML**, and **XML** (the default).

Step 2 Set the DCPL property **DCS/getCommitCLIConfigAfterDownload** to true (the default).

This allows Prime Fulfillment to retrieve the committed CLI configuration after an XML configuration has been downloaded. See Viewing Configlets on IOS XR Devices, page 8-6 for more information.

Step 3 Create the device in Prime Fulfillment as an IOS XR device, as follows:

a. Create the Cisco device by choosing **Inventory > Physical Inventory > Devices > Create**.

The Create Cisco Device window appears.

b. Set the OS attribute, located under Device and Configuration Access Information, to IOS_XR.

Note For additional information on setting DCPL properties and creating Cisco devices, see Appendix B, "Property Settings".

Step 4 Create and deploy MPLS VPN service requests, following the procedures in this guide.

Sample configlets for IOS XR devices are provided in Sample Configlets, page 5-165.

Migrating PE Devices from IOS to IOS XR

For information on migrating PE devices from IOS to IOS XR, see Migrating PE Devices from IOS to IOS XR, page 5-98.

Defining VPNs

During service deployment, Prime Fulfillment generates the Cisco IOS commands to configure the logical VPN relationships. At the beginning of the provisioning process, before creating a Service Policy, a VPN can be defined within Prime Fulfillment.

Note It is also possible to specify VPN and VRF information in an independent VRF object, which is subsequently deployed to a PE device and then associated with an MPLS VPN link via an MPLS VPN service request. For details on using this feature, see Independent VRF Management, page 5-14

This section describes how to define MPLS VPNs and IP Multicast VPNs. It contains the following sections:

- Creating an MPLS VPN, page 5-7
- Creating an IP Multicast VPN, page 5-8
- Enabling a Unique Route Distinguisher for a VPN, page 5-11

Creating an MPLS VPN

At its simplest, a virtual private network (VPN) is a collection of sites that share the same routing table. A VPN is also a framework that provides private IP networking over a public infrastructure such as the Internet. In Prime Fulfillment, a VPN is a set of customer sites that are configured to communicate through a VPN service. A VPN is defined by a set of administrative policies.

A VPN is a network in which two sites can communicate over the provider's network in a private manner; that is, no site outside the VPN can intercept their packets or inject new packets. The provider network is configured such that only one VPN's packets can be transmitted through that VPN—that is, no data can come in or out of the VPN unless it is specifically configured to allow it. There is a physical connection from the provider edge network to the customer edge network, so authentication in the conventional sense is not required.

To create an MPLS VPN, perform the following steps:

Step 1 Choose **Inventory > Logical Inventory > VPNs**.

The VPNs window appears.

Step 2 From the VPNs window, click **Create**.

The Create New VPN window appears.

Step 3 Enter the name of the VPN in the Name field.

Step 4 Click **Select** and choose a customer associated with this VPN from the Customer filed.

Step 5 To create a default routing community, check the **Create Default Route Target(s)** check box and choose a provider.

Step 6 To enable the unique router distinguisher, check the check box.For coverage of this attribute see Enabling a Unique Route Distinguisher for a VPN, page 5-11

Step 7 Enter the OSPF domain IDvalue in decimal format. The Hex value field is a non-editable text field that displays the equivalent hex value. The hex value is what actually gets displayed on the device.

- You can modify the OSPF domain ID at any time. If you attempt to modify the OSPF domain ID for a VPN that is already deployed, all the service requests that use this VPN and have the attribute Use VRF/VPN Domain ID enabled are moved to the **Requested** state. Prime Fulfillment provides a list of the service requests that were moved to **Requested**, so that you can deploy them. This operation is similar to enable/disable multicast for a deployed VPN.
- OSPF domain ID is supported only on IOS XR devices. In the case of IOS devices, Prime Fulfillment ignores the this attribute if you select a VPN with an OSPF domain ID specified.
- For additional information, see the discussion of the OSPF Domain ID attribute in OSPF Protocol Chosen, page 5-59.

Step 8 To enable multicast for the VPN, you can check the **Enable IPv4 Multicast** or **Enable IPv6 Multicast** check boxes. See Creating an IP Multicast VPN, page 5-8.

Note These attributes are not supported for use with MVRFCE policies and service requests.

Note Enable IPv6 Multicast is not supported on IOS and IOS 6VPE devices.

Note Next set of attributes (up to **Route Target(s)**) only become active in the GUI if one of the enable multicast attributes is checked. See Creating an IP Multicast VPN, page 5-8, for coverage of these attributes.

Step 9 **Route Target(s):** If you do not choose to enable the default **Route Target(s)**, you can choose a customized **Route Target(s)** that you have already created in Prime Fulfillment.

Note You must specify a CERC if multicast is enabled.

a. From the CE Routing Communities pane, click **Select**.

   The Select CE Routing Communities dialog box appears.

b. Check the check box for the CERC you want used for this VPN, then click **Select**.

   You return to the Create VPN dialog box, where the new CERC selection appears, along with its hub route target (HRT) and spoke route target (SRT) values.

Step 10 Check the Enable VPLS check box to enable VPLS.

Step 11 Choose the VPLS service type from the Service Type drop-down menu: **ERS** (Ethernet Relay Service) or **EWS** (Ethernet Wire Service).

Step 12 Choose the VPLS topology from the drop-down menu: **Full Mesh** (each CE will have direct connections to every other CE) or **Hub and Spoke** (only the Hub CE has connection to each Spoke CE and the Spoke CEs do not have direct connection to each other).

Step 13 When satisfied with the settings for this VPN, click **Save**.

You have successfully created a VPN, as shown in the Status display in the lower left corner of the VPNs dialog box.

Creating an IP Multicast VPN

An IP address that starts with the binary prefix 1110 is identified as a *multicast group address*. There can be more than one sender and receiver at any time for a given multicast group address. The senders send their data by setting the group address as the destination IP address. It is the responsibility of the network to deliver this data to all the receivers in the network who are listening to that group address.

Note Before you can create a VPN with multicast enabled, you must define one or more multicast resource pools. See Creating a Multicast Pool, page 2-46, for further information.

If the multicast VPN is used in a service request on a device running IOS XR, not all of the multicast attributes in the Create VPN window are supported. This is because there is not a one-to-one mapping of IOS multicast commands to IOS XR commands. These exceptions are noted in the following steps: For a comparison of multicast routing commands in IOS and IOS XR, see Multicast Routing on IOS and IOS XR Devices, page 5-36.

Multicast VRF deployments are supported also. For more information about VRF object support in Prime Fulfillment, see Independent VRF Management, page 5-14

To create an IP Multicast VPN, follow the procedure described in Creating an MPLS VPN, page 5-7 to the place where you can enable multicast for the VPN, then perform the following steps:

Step 1 Check one or both of **Enable IPv4 Multicast** or **Enable IPv6 Multicast** check boxes to enable multicast for the VPN.

Note Enable IPv6 Multicast is not supported on IOS and IOS 6VPE devices.

The current window refreshes with additional fields becoming active.

Usage notes:

- For IOS XR PE devices running release 3.7.0 or later, Prime Fulfillment allows a multicast VPN to be deployed on an IPv6 PE-CE link and multicast to be enabled during the creation of the VRF object.
- When creating a VPN, you can enable multicast for IPv4, IPv6, or both. You can enter IPv6 addresses as static Rendezvous Point (RP) addresses if IPv6 multicast is enabled during the creation of a VPN or VRF object.
- You can also modify an existing VPN object to enable multicast for IPv4, IPv6, or both. When IPv4 multicast is enabled, all deployed service requests containing IPv4 links of the same VPN are moved into Requested state.
- In addition, you can specify within the MPLS service request whether you want to enable multicast for IPv4, IPv6, or both on a given MPLS link.
- When IPv6 multicast is enabled, all deployed service requests containing IPv6 links of the same VPN are moved into Requested state. If IPv4 is previously configured and only IPv6 multicast is enabled in a VPN, only the service requests with IPv6 links are moved into Requested state.
- You can modify an existing VPN object and add IPv6 static RP addresses when IPv6 multicast is enabled. Any service requests already in Deployed state are then moved to the Requested state.
- You can create a service policy or an MPLS VPN link in the service request with IPv6 Numbered or IPv4+IPv6 Numbered as the IP addressing scheme and a multicast VPN with multicast enabled.

Step 2 For MDT (Multicast Distribution Tree) addresses, either accept the default (check box already checked) to enable the auto pick function, or uncheck the auto pick check box, then enter values in the next two fields:

- **Default MDT Address**
- **Data MDT Subnet**

Step 3 From the **Data MDT Size** drop-down list, choose a value for Data MDT Size.

Step 4 In the **Data MDT Threshold** field, enter a valid value for Data MDT Threshold (1 - 4294967 kilobits/sec).

Step 5 For Default PIM (Protocol Independent Multicast) Mode, choose a mode from the **Default PIM Mode** drop-down list:

- SPARSE_MODE
- SPARSE_DENSE_MODE

Tip Multicast routing architecture allows the addition of IP multicast routing on existing IP networks. PIM is an independent unicast routing protocol. It can be operated in two modes: dense and sparse.

Note For IOS XR devices, when SPARSE_DENSE_MODE is chosen, no configlet will be generated. Sparse-dense mode is not supported by IOS XR, only sparse mode (default) and bidirectional mode. For IOS XR devices, sparse mode is running by default when multicast routing is enabled on an interface. Hence, no configlet will be generated for sparse mode either.

Step 6 In the **MDT MTU** field, enter a valid value for MDT MTU (Maximum Transmission Unit).

Note The ranges for IOS and IOS XR devices for this attribute are different. The range for IOS devices is from 576 to 18010, and for IOS XR devices it is from 1401 to 65535. Device type validations are done during service request creation when it is known what type of device the multicast VPN will be deployed on.

Step 7 To enable PIM SSM (Source Specific Multicast), check the associated check box.

When you check the check box:

a. The associated drop-down list goes active with the DEFAULT enumeration populated as the SSM default. This will create the following CLI: **ip pim vrf** *vrfName* **ssm default**.

Note For IOS XR devices, when DEFAULT is chosen, no configlet will be generated because this command is running by default on IOS XR devices, using the standard SSM range 232.0.0.0/8.

b. If you would like to associate an access-list number, or a named access-list, with SSM configuration, choose the RANGE enumeration from the SSM drop-down list instead of DEFAULT. This will create the following CLI: **ip pim vrf** *vrfName* **ssm range {ACL# | named-ACL-name}**.

Step 8 If you choose RANGE in the previous step, then the **SSM List Name** field goes active for you to enter Access-list number or Access-list name.

Step 9 In the **Multicast Route Limit** field, enter a valid value for the Multicast Route Limit (1–2147483647).

Usage notes:

- The command to set the route limit per VRF is supported for both IOS and IOS XR.
- The range listed in the GUI (1–2147483647) is for IOS. For IOS XR, the range is 1–200000. To display information on the range values in the GUI, click the tool tip icon for the attribute.
- Prime Fulfillment performs device-specific validations of the value when a service request is created using the VPN or VRF object using this attribute.
- The value of Multicast Route Limit is shared for both IPv4 and IPv6 address families.

Step 10 To enable the auto RP (Rendezvous Point) listener function, check the **Enable Auto RP Listener** check box.

Note For IOS XR devices, no configlet is generated for this attribute. By default, this feature is running on IOS XR devices.

Step 11 To configure Static RPs, check the **Configure Static-RP** check box.

When you check this, the Edit option for PIM Static RPs goes active.

Step 12 To edit or add PIM Static RPs, click **Edit** in the **PIM Static RPs** area.

The Edit PIM Static RPs window appears.

Step 13 Complete all applicable fields in the Edit PIM Static RP window, then click **OK**.

The data now appears in the main Create VPN window.

Step 14 To save your changes and add this Multicast VPN to your system, at the bottom of the window, click **Save**.

Enabling a Unique Route Distinguisher for a VPN

Note In ISC 6.0, enabling unique route distinguishers is supported for both IOS and IOS XR PE devices. It is also supported for IPv6 and dual-stacked services.

Support for multipath load sharing requires unique route distinguishers (RDs) for each PE router for a VPN (VRF). This is to prevent the same RDs from being allocated to different customers. This allows the use of the same RD for the same VRF. That is, all sites in the PE can have the same unique RD. The unique RD feature is optional. It is enabled at both a global VPN level or a service request level. To enable the unique RD per PE for a VPN, the Create VPN window contains the attribute **Enable Unique Route Distinguisher field**.

Each VPN deployed through Prime Fulfillment for which **Enable Unique Route Distinguisher** has been selected is marked as a multipath VPN. This ensures a unique RD allocation for each VRF on each PE. Enabling multipath for an already deployed VPN creates new VRFs on all the PEs of the VPN and assigns a unique RD. When **Enable Unique Route Distinguisher** is selected for the VPN, the **Allocate New Route Distinguisher** and **VRF and RD Overwrite** attributes will be disabled when setting up a policy or service request that uses this VPN.

To use the unique RD feature, perform the following steps:

Step 1 When creating a VPN, check the **Enable Unique Route Distinguisher** check box.

Step 2 When subsequently creating a service policy and/or service request, select the VPN in the VRF and VPN Membership window.

The Unique Route Distinguisher **field** appears.

Step 3 If the unique RD allocation functionality is required, check the **Unique Route Distinguisher** check box.

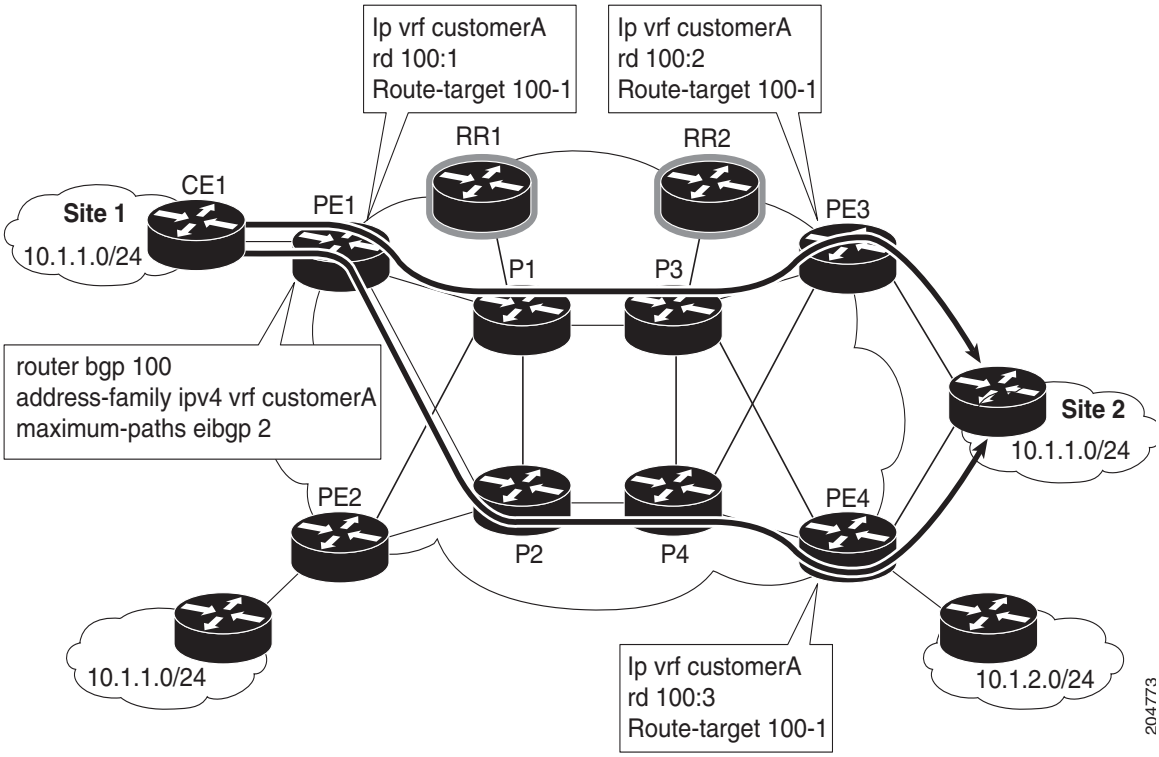
For additional information on how this feature is used with MPLS VPN policies and service requests, see Defining VRF and VPN Information, page 5-72.

Provisioning MPLS Service Requests Using Unique Route Distinguisher

The unique route distinguisher (RD) feature is used to implement multipath load balancing. Multihomed CEs often require load balancing across multiple available paths. In a full-mesh BGP environment, PEs receive all the available paths to a given prefix, and load balancing can easily be achieved. However, when route reflectors are present in the service provider core, PE routers receive only one route, even if multiple paths exist, and load balancing does not occur. To achieve load balancing, the service provider

needs to implement unique RD values for the customer VPN on each PE router. In addition, eiBGP configuration with the desired number of paths (across which load balancing is desired) needs to be enabled in the service provider environment. Figure 5-1 illustrates a load balancing example.

Figure 5-1 Load Balancing Using Different RDs



The support for multipath load sharing requires unique RDs for each PE router for a VPN (VRF). This is to prevent the same RDs from being allocated to different customers. This allows the use of the same RD for the same VRF. That is, all sites in the PE can have the same unique RD. The unique RD feature is optional. You can specify its use at both the policy or service request level.

It is enabled at both a global VPN level or a service request level.

Prime Fulfillment supports BGP multipath load sharing through fields and options in the Prime Fulfillment GUI. The following steps provide an overview of how to do this.

Step 1 When creating a VPN, check the **Enable Unique Route Distinguisher** check box in the Create VPN window.

For some additional coverage of this, see Enabling a Unique Route Distinguisher for a VPN, page 5-11.

Step 2 When setting the attributes in the policy (MPLS Policy Editor - VRF and VPN Membership window) or service request (MPLS Link Attribute Editor - VRF and VPN window), use the **BGP Multipath Load Sharing** check box to enable or disable BGP multipath load sharing.

Enabling BGP multipath load sharing by checking the check box causes additional attributes to appear in the GUI. For detailed coverage of these attributes and how to set them, see BGP Multipath Load Sharing and Maximum Path Configuration, page 5-75.

Step 3 When creating a service request based on this policy, check the **Unique Route Distinguisher** check box in the MPLS Link Attribute Editor - VRF and VPN window.

Note The Unique Route Distinguisher attribute is dynamic and only shows up in the GUI if a VPN with unique RD enabled is selected.

Step 4 Complete the service request creation, and save the service request.

Use Cases for Using Unique RD

The following use cases demonstrate the behavior of unique RD feature.

Use case details:

- The default values of the VPN/VRF are:

```
ip vrf V24:unique2
rd 1:33
route-target import 1:14
route-target import 1:15
route-target export 1:14
```

- Service requests are created using PEs and enabling or disabling the Unique RD attribute during service request creation, as shown in Table 5-1.

The outcomes for various cases are described in the Results column of the table.

Table 5-1 Unique RD Use Cases

SR #	PE	Unique RD	VRF:RD	Results
1	pe1	False	V24:33	Prime Fulfillment uses the default *vrfName:RD*, because this is the first time this PE has been configured with this *vrfName:RD* name.
2	pe2	False	V24:33	Prime Fulfillment uses the default *vrfName:RD*.
3	pe3	True	V25:34	Prime Fulfillment creates a new *vrfName:RD*, because Unique RD is true, and it is on a different PE. This PE (pe3) did not have this *vrfName:RD* configured.
4	pe3	True	V25:34	Prime Fulfillment uses the *vrfName:RD* from SR #3, because the new RD is already present on the PE router.
5	pe2	True	V26:35	Prime Fulfillment creates a new *vrfName:RD*, because this is the first time Unique RD is selected as true, even though a VRF of V24:33 was already configured in SR #2.
6	pe1	True	V27:36	Prime Fulfillment creates a new *vrfName:RD*, because this is the first time Unique RD is selected as true on this PE, even though a VRF of V24:33 was already configured in SR #1.

Table 5-1 Unique RD Use Cases (continued)

SR #	PE	Unique RD	VRF:RD	Results
7	pe1	False	V24:33	Prime Fulfillment uses the default *vrfName:RD*, as in SR #1.
8	pe3	False	V24:33	Prime Fulfillment uses the default *vrfName:RD*, as in SR #1.
9	pe3	True	V25:34	Prime Fulfillment uses the newly created *vrfName:RD* in SR #4, because it already created a new *vrfName:RD* for this PE.
10	pe2	True	V26:35	Prime Fulfillment uses the newly created *vrfName:RD* in SR #5, because it already create a new *vrfName:RD* for this PE.
11	pe1	True	V27:36	Prime Fulfillment uses the newly create *vrfName:RD* in SR #6, because it already create a new *vrfName:RD* for this PE.

Independent VRF Management

This section describes independent VRF management, which provides a means to create, deploy and manage VRF objects independent of MPLS VPN links and service requests. Deployed VRF objects can also be used with MPLS VPN links.

In the traditional VRF (VPN routing and forwarding) model available in previous releases of Prime Fulfillment, the operator first creates a VPN object and then associates it to an MPLS VPN link. The necessary VRF information is generated and deployed at the time the MPLS VPN link is provisioned. The VRF information is removed only when the last link associated with the VRF is decommissioned. However, in certain cases, it might be desirable to have the VRF information provisioned independent of the physical link. Prime Fulfillment now supports this scenario through the independent VRF management feature described in this section. This lets you create, modify, and delete VRF objects independently of MPLS VPN links. This provides several advantages:

- VRF information and templates can be directly deployed on a PE device without being associated with an interface.
- VRF information can exist without links pointing to it.
- A VRF object can be modified, even if it is associated with links.
- Route targets (RTs) can be added and removed without causing outages.

Managing VRFs independently of physical links involves the following tasks, which are covered in detail in the rest of this section:

- Creating, modifying, and deleting VRF objects.
- Creating, modifying, deploying, decommissioning, and deleting a new type of service request, called a VRF service request.
- Using deployed VRF objects with MPLS VPN links via service policies and service requests.
- Migrating traditional MPLS VPN service requests to the independent VRF model.

Note The traditional Prime Fulfillment VRF model is still supported for backward compatibility. The choice of which VRF model to use is available during MPLS VPN link creation. This is described in subsequent sections of this section.

Note Independent VRF association is not supported for MVRFCE-based policies and service requests.

This section covers the following topics:

Multicast Support for IPv6 on IOS XR Devices

For IOS XR PE devices running release 3.7.0 or later, Prime Fulfillment allows multicast to be enabled during the creation of the VRF object. When creating a VRF object, you can enable multicast for IPv4, IPv6, or both. You can enter IPv6 addresses as static Rendezvous Point (RP) addresses if IPv6 multicast is enabled during the creation of a VRF object.

You can also modify an existing VRF object to enable multicast for IPv4, IPv6, or both. When IPv4 multicast is enabled, all deployed service requests containing IPv4 links of the same VPN or VRF are moved into Requested state.

In addition, you can specify within the MPLS service request whether you want to enable multicast for IPv4, IPv6, or both on a given MPLS link.

When IPv6 multicast is enabled, all deployed service requests containing IPv6 links of the same VPN or VRF are moved into Requested state. If IPv4 is previously configured and only IPv6 multicast is enabled in a VPN, only the service requests with IPv6 links are moved into Requested state.

You can modify an existing VRF object and add IPv6 static RP addresses when IPv6 multicast is enabled. Any service requests already in Deployed state are then moved to the Requested state.

You can create a service policy or an MPLS VPN link in the service request with IPv6 Numbered or IPv4+IPv6 Numbered as the IP addressing scheme and a multicast VRF with multicast enabled.

Working with VRF Objects

This section describes how to create, modify, and delete VRF objects. Subsequent sections in this section cover how the VRF objects are used in service requests.

Creating a New VRF Object

Creating a VRF object is similar to creating a VPN. However, there are some extra attributes involved, such as Import RT List and Export RT List. After the VRF object is created, you will later provision it using a VRF service request, as covered in later sections of this section.

To create a VRF object, perform the following steps:

Step 1 Choose **Inventory > Logical Inventory > VRFs.**

Step 2 From the VRF Management window, click **Create**.

The Create New VRF window appears.

Step 3 **Name:** Enter the name of the VRF object.

This is a simple text field. Enter any name of your choice. It is recommended not to use special characters (' ` " < > ( ) [ ] { } / \ & ^ ! ? ~ * % = , . + |), as this may cause misconfiguration of the VRF name for certain devices.

This name will be directly deployed on the PE device. All the validations applicable for a VPN name while creating a VPN object in Prime Fulfillment are applicable for a VRF name. This attribute is required.

Step 4 **Provider:** To choose the provider associated with this VRF:

a. Click **Select**.

The Select Provider dialog box appears.

b. From the list of providers, choose the appropriate provider, then click **Select**.

Step 5 **Description:** Enter a description of the VRF, if desired.

No validation is done on the description entered.

Step 6 **Route Target(s):** To select a Route Target for this VRF:

a. Click **Select**.

The Select CE Routing Communities dialog box appears.

b. From the list, choose the appropriate Route Target, then click **Select**. Only one Route Target is allowed per VRF.

Step 7 **Import RT List:** Enter one or more Route Targets (RTs) to be imported in the VRF.

For multiple RTs, use a comma (,) separated list. An example RT list is 100:120,100:130,100:140.

Step 8 **Export RT List:** Enter one or more Route Targets (RTs) to be exported from the VRF.

For multiple RTs, use a comma (,) separated list.

Step 9 **Import Route Map:** Enter the name of a route map defined on the device.

Prime Fulfillment will validate this name while provisioning the VRF. If the route map is not defined, Prime Fulfillment will generate an error.

Step 10 **Export Route Map:** Enter the name of a route map defined on the device.

Prime Fulfillment will validate this name while provisioning the VRF. If the route map is not defined, Prime Fulfillment will generate an error.

Step 11 **Maximum Routes:** Specify the maximum number of routes that can be imported into the VRF.

This is an integer value from 1 to 4294967295 for IOS devices and from 32 to 2000000 for IOS XR devices.

Step 12 **Threshold:** Specify the threshold value, which defines a percentage, which, if exceeded, generates a warning message.

This is an integer value from 1 to 100. This attribute is mandatory for IOS devices and optional for IOS XR devices. Validations for specific device type will be done during service request creation.

Step 13 **RD Format:** To specify the format of the RD (route distinguisher) format, choose a format type from the drop-down list.

- RD_AS—Specify RD in AS (autonomous system) format. This is the default selection.
- RD_IPADDR—Specify RD in IP address format. This is supported for IOS and IOS XR PE devices.

The RD format chosen determines the how the RD should be set in the next step.

Step 14 **RD:** Specify a RD (route distinguisher) manually (according to the format chosen in the previous step), or check the **Autopick RD** check box to have Prime Fulfillment automatically choose an RD from the Route Distinguisher pool (if one has been set up).

Usage notes:

- This attribute is required.
- Checking the Autopick RD check box disables the RD text entry field.
- If the Autopick RD check box is checked in conjunction with the RD_IPADDR format, then the VPN ID for the RD will automatically selected from the RD pool of the respective provider and appended to the label *IP* to form the RD. Example: IP:1245. (This value appears when the VRF object is saved and then edited.) You choose the actual IP address when the service request is created, as the IP address (that is, the loopback IPv4 address) might differ for different PEs.
- If the Autopick RD check box is checked in conjunction with the RD_AS format, then Prime Fulfillment picks the value from the Route Distinguisher pool and assigns it to this particular VRF object.
- If Autopick RD is not checked, you must specify the RD manually in the provided text field using one of the following formats (as specified in the RD Format attribute):
  - The RD value for the RD_AS format must be *as_number:number*, where *as_number* is an AS number (2-byte value) and *number* is a 4-byte integer value. The AS number can be in the range 1 through 65,535. Example: 100:1254.
  - The RD value for RD_IPADDR must be *ip_address:number*, where *ip_address* is an IPv4 address and *number* is a 4-byte integer value. The number can be in the range 1 through 65,535 only. Example: 10.23.6.5:1245.
- If the RD value is entered manually in IP address format, the operator is responsible for the deployment of the VRF across different PEs.
- RD format validation is performed based on the RD format set in the RD Format attribute.
- No check is done to verify the association with the PE, other than validating the new RD format.
- Prime Fulfillment allows the modification of an existing VRF object with the new RD format only if the VRF object is not deployed.
- The following Prime Fulfillment template variables support RD Format:
  - RD_FORMAT
  - RD_IPADDRESS

Step 15 **OSPF Domain ID:** Enter an OSPF domain ID in decimal format.

Usage notes:

- Enter the value in decimal format. The Hex value: field is a non-editable text field that displays the equivalent hex value. The hex value is what actually gets displayed on the device.

- You can modify the OSPF domain ID at any time. If you attempt to modify the OSPF domain ID for a VRF that is associated with a deployed MPLS service request and has the Use VRF/VPN Domain ID attribute enabled, those service requests are moved to the **Requested** state. Prime Fulfillment provides a list of the service requests using this VRF object, so that you can deploy them.
- The OSPF Domain ID property has no effect on the VRF service request, and no configuration related to OSPF Domain ID gets deployed with VRF service request.
- OSPF domain ID is supported only on IOS XR devices. In the case of IOS devices, Prime Fulfillment ignores the this attribute if you use a VRF object with an OSPF domain ID specified.
- The OSPF domain ID attribute uniquely identifies the OSPF domain from which a route is redistributed. This domain ID should be unique per customer. For IOS devices, because IOS allows only one VRF per process, the default behavior is that the OSPF process ID is considered as the OSPF domain ID. IOS XR supports multiple VRFs per process. Therefore, for IOS XR devices, you need to explicitly configure a unique OSPF domain ID for each VRF. You can configure one VRF per OSPF process, but it is not a scalable solution.
- For additional information, see the discussion of the OSPF Domain ID attribute in OSPF Protocol Chosen, page 5-59.

Step 16 **Enable IPv4 Multicast** or **Enable IPv6 Multicast:** Check one or both of these check boxes to enable multicast VRF.

The multicast attributes below this check box are enabled for use. For details on how to set the multicast attributes, see Creating an IP Multicast VPN, page 5-8.

Note This attribute is not supported for use with MVRFCE policies and service requests.

Note Enable IPv6 Multicast is not supported on IOS and IOS 6VPE devices.

Note Route Target is mandatory if multicast is enabled.

Note For the MDT MTU attribute: The range for IOS devices is from 576 to 18010. The range for IOS XR devices is from 1401 to 65535. Validations for specific device type will be done during service request creation.

Step 17 When you are satisfied with the settings for this VRF object, click **Save**.

Prime Fulfillment creates a new VRF object based the attributes selected. The new VRF is listed in the VRF Name column of the window.

Copying a VRF Object

You can use an existing VRF object as the basis for a new one. You do this by copying a VRF object, renaming the copy, and (optionally) modifying its attributes.

To copy an existing VRF object, perform the following steps:

Step 1 Choose **Inventory > Logical Inventory > VRFs**.

The VRF Management window appears.

Note The example assumes that a VRF object has already been created. See Creating a New VRF Object, page 5-15 for information on how to create a VRF object.

Step 2 Select an existing VRF object (for example, VRF_1) by checking the check box for the VRF object.

When you select a VRF object, the Edit, Copy, and Delete buttons become active.

Step 3 To copy the VRF object, click the **Copy** button.

The attribute fields are populated with values from the VRF object being copied.

Step 4 Provide a name for the new VRF object by changing the name in the **Name** field.

Step 5 Edit other attributes in the Create VRF window as desired.

Note The copy VRF function copies all attributes of the parent except the route distinguisher (RD), Default MDT Address, and Data MDT Subnet.The RD is always set to auto pick (the Autopick RD check box is checked by default). If auto pick is set for the parent VRF, it will be carried to the VRF object created by the copy function.

Step 6 When you are finished with the edits, click the **Save** button.

The VRF Management window appears, with the new VRF object.

Step 7 The VRF object copy operation is complete.

Searching for VRF Objects in the Prime Fulfillment Repository

All VRF objects are stored in the Prime Fulfillment repository. You can display these by accessing the VRF Management window at **Inventory > Logical Inventory > VRF** in the Prime Fulfillment GUI. You can search for VRF objects using the **Show VRF with** drop-down list together with the **matching** field. The **Show VRF with** drop-down list enables you to display VRF objects by searching for these attributes:

- VRF Name
- Provider
- Route Distinguisher
- Route Target

Note The search is case-insensitive, and wildcard (*) searches are supported.

Modifying Non-Deployed VRF Objects

VRF objects can be modified individually (single VRF edit) or in batch mode (multi-VRF edit). This section covers the basic steps for modifying VRF objects which have not yet been deployed via a VRF service request or associated with MPLS VPN links. There are some special considerations when modifying VRFs which have been deployed, as described in Modifying Deployed VRF Objects, page 5-21.

Single-VRF Edit Mode

To edit one VRF object, perform the following steps:

Step 1 Choose **Inventory > Logical Inventory > VRF** to list the VRF objects in the Prime Fulfillment repository.

The VRFs window appears.

Step 2 Select the VRF you want to edit and click the **Edit** button.

Step 3 Update any attributes you want to edit.

Step 4 Click **Save** to save the edits.

Multi-VRF Edit Mode

The multi-VRF edit feature allows you to modify common attributes on more than one VRF. For example, multi-VRF edit is useful for adding and/or removing route targets on multiple VRFs.

To edit multiple VRF objects simultaneously, perform the following steps:

Step 1 Choose **Inventory > Logical Inventory > VRFs** to list the VRF objects in the Prime Fulfillment repository.

The VRFs window appears.

Step 2 Select the VRFs you want to edit and click the **Edit** button.

The Edit Multiple VRFs window appears.

The Edit VRFs window is similar to the Create VRF and Edit VRF windows. However, there is an additional field, **VRF Details**, and the format of the RT import/export fields are laid out differently. Also, some attributes are not available for editing in multi-VRF edit mode.

Step 3 To see details of the VRFs being edited, click the **Attributes** link in the VRF Details row.

The VRF Details window appears. This lists the VRFs being edited and displays the following attributes for each VRF:

- Name
- Provider
- Route Target
- Import Route Map
- Export Route Map
- Import Route Target
- Export Route Target
- MultiCast IPv4

- MultiCast IPv6

Step 4 To add or remove import or export route maps, enter the desired values in the provided fields.

You can enter more than one RT in each field. For multiple RTs, use a comma (,) separated list.

Step 5 Update the **Route Target(s)**, **Import Route Map**, **Export Route Map**, and **Multicast Attributes** settings as desired.

Note The **Provider** attribute cannot be edited in multi-VRF editing mode.

Step 6 To save the edits, click **Save**.

Modifying Deployed VRF Objects

After a VRF object is deployed on a PE device through a VRF service request (see Deploying VRF Service Requests, page 5-24), there are some special considerations to be aware of when modifying the VRF object.

- The VRF object might have been associated with multiple links and/or VRF service requests.
- Unlike traditional VPN objects, you can modify a VRF object even if it is referenced by multiple VRF service requests.
- The **VRF Name**, **Provider**, and **RD** attributes cannot be changed after the VRF object is deployed.

Note The **RD** attribute can be modified if the VRF service request is deployed on a PE device running IOS 12.0 (32) SY or greater.

To modify a deployed VRF object, perform the following steps:

Step 1 When you attempt to modify a deployed VRF object, the Affected Jobs window appears.

The window displays the affected VRF service requests associated with the VRF object being modified. The Job ID, SR ID, Link ID, VRF Name, and Description information for each VRF service request are listed.

Step 2 To display more details about a VRF service request, click the **Job ID** link.

The Service Request Details window appears.

Step 3 Verify the service request details, if desired.

Step 4 Perform one of the following actions:

a. Click **Save** to save the VRF object and move all of the affected VRF service requests to the **Requested** state.

b. Click **Save and Deploy** to save the VRF object, move all of the affected VRF service requests to the **Requested** state, and schedule an immediate deployment for all of the VRF service requests.

c. Click **Cancel** to cancel the operation and return to the Edit VRFs window.

Deleting VRF Objects

To delete VRF objects from the Prime Fulfillment repository, perform the following steps:

Note There are some prerequisite steps you must perform if the VRF object or objects are still in use by a VRF service request, as mentioned in the notes following the procedure.

Step 1 Choose **Inventory > Logical Inventory > VRF** to list the VRF objects in the Prime Fulfillment repository.

The VRFs window appears.

Step 2 Select the VRFs you want to delete and click the **Delete** button.

Step 3 Click **Delete** to confirm.

If the VRF objects are not in use, the selected VRF objects are deleted.

Deleting VRF Objects Associated with VRF Service Requests

A VRF object cannot be deleted if it is still associated with any VRF service request. If you attempt to do so, you receive a Delete VRF Failed message in the Status window. In this case you must first decommission, deploy, and purge all of the related VRF service requests before you can delete the VRFs object. Use the information provided in the error message to identify the VRF services requests and links related to the VRF object you are attempting to delete.

Working with VRF Service Requests

Saved VRF objects are deployed on a Provider Edge (PE) device through a special type of service request called a VRF service request.

Overview of VRF Service Requests

The VRF service request allows the VRF object to be configured on a router without having to select a physical interface. Each VRF service request consists of one or more links. Each link consists of the following elements:

- One VRF object
- One PE object
- One template (optional)

In addition, VRF service requests are associated to a customer.

Note An important difference between regular MPLS service requests and VRF service requests is that there is no service policy required for a VRF service request. As a result, the VRF service request is not associated with a service policy.

The VRF service request states follow the normal Prime Fulfillment service request state transitions, as described in the Service Enhancements, page 5-79.

Defining VRF Service Requests

To define a VRF service request, perform the following steps:

Step 1 Choose **Operate > Service Requests > VRF** to access the VRF Service Requests window.

The VRF Service Request Editor window appears.

Note If necessary, click the **Add Link** button to create a row for setting the link information.

This window allows you to define the VRF service request by setting up one or more links, each consisting of a VRF object, PE device, and an optional template. You also specify the address scheme for each link. You can also view or, in some cases, set the Route Distinguisher (RD) value. This depends on how the RD format and RD were specified when creating the VRF object. You can deploy any number of links with any combination of PE devices and VRF objects. An important point to note is that no physical interface on the router needs to be selected.

To set up a link, continue with the steps in the procedure, as follows:

Step 2 Set the customer for the VRF service request by clicking on the link beside the Customer attribute.

The Select Customer window appears. Choose the desired customer and click the **Select** button. This attribute is optional.

Step 3 Click the **Select VRF** link to choose a VRF object from the Prime Fulfillment repository.

The Select Independent VRF window appears.

Step 4 Choose a VRF object by clicking on a radio button and clicking the **Select** button.

If desired, you can limit the VRF objects displayed by searching by VRF Name, Provider, Route Distinguisher, or Route Target using the **Show VRFs with** and **matching** fields.

Note For steps on how to add VRF objects to the Prime Fulfillment repository, see Creating a New VRF Object, page 5-15.

Step 5 Click the **Select PE** link to choose a PE device for the link.

The Select PE Device window appears.

Step 6 Choose a PE by clicking on a radio button and clicking the **Select** button.

If desired, you can limit the PE devices displayed by using the **Show PEs with** and **matching** fields.

This step specifies the PE device on which to deploy the VRF object selected in Steps 4 and 5.

Note Because the VRF object and the PE device must belong to the same provider, Prime Fulfillment limits the list of PEs displayed to those with the same provider specified in the VRF object chosen for the link.

After the PE is selected, the RD IP Address Value column will display a message or, in some cases, a text field in which to enter an IP address. This is covered in subsequent steps below.

Step 7 Click the **Add Template** link to choose a template data file to be associated with the link.

The Add/Remove Templates window appears. This is a standard Prime Fulfillment window for selecting a data file and specifying operations such as append and prepend. For information on working with templates in Prime Fulfillment, see Chapter 9, "Managing Templates and Data Files." For specific information about using the Add/Remove Templates window, see Using Templates with Service Requests, page 9-24.

Step 8 Specify the address scheme by choosing the appropriate selection from the **Address Family** drop-down list for the link.

The choices are:

- IPv4
- IPv6
- IPv4 and IPv6

The IPv4 and IPv6 option causes the VRF object to be deployed with both IPv4 and IPv6 configurations.

Step 9 If appropriate for your configuration, enter an RD IP address in the text field of the RD IP Address Value column. Alternatively, you can click the **Select_Loopback** link to pick a loopback IP address of the PE device used in the service request.

Usage notes:

- The contents and behavior of the RD IP Address Value field depend on how the RD Format and RD attributes were specified for the VRF object that is being used in the service request, as follows:
  - If the VRF object has RD Format set as RD_IPADDR and Autopick is checked for the RD attribute, then the RD IP Address Value column provides a text field in which to manually enter the RD IP address value. Alternatively, you can pick a loopback IP address of the PE device used in the service request. The RD is formed by appending to this IP address the VPN ID picked from the RD pool of the respective provider. Prime Fulfillment validates the IP address entered. Basic IPv4 addresses are allowed. No network prefixes are permitted.
  - If the VRF object has RD Format set as RD_IPADDR and you manually entered an RD IP address for the RD attribute, then the RD IP Address Value column states "RD IP Address Manual". You do not enter an IP address in this case.
  - If the VRF object has RD Format set as RD_AS and Autopick was checked for the RD attribute, or a value was entered manually, then the RD IP Address Value column states "RD AS Format". You do not enter a value in either of these cases.
- After the VRF service request is deployed with the RD using an IP address you entered in the text field, the RD IP Address Value field is disabled and cannot be changed. If the RD IP Address Value needs to be modified, you must decommission, purge, and redeploy the VRF service request.

Step 10 If you want to set up additional links for the VRF service request, click the **Add Link** button and repeat Steps 4 through 9 for each link.

Step 11 When you have completed setting up the link(s) for the VRF service request, click **Save** to save the VRF service request.

The Service Requests window appears and you see the VRF service request displayed with Job ID, State, Type and other attributes. The VRF service request is initially in the Requested state.

Step 12 To deploy a VRF service request, see Deploying VRF Service Requests, page 5-24.

Deploying VRF Service Requests

To deploy a VRF service request, perform the following steps:

Step 1 In the Service Requests window, choose the VRF service request you want to deploy.

Step 2 Click the **Deploy** button and choose **Deploy** from the drop-down list.

The Deploy Service Request task window appears.

Step 3 Set the task parameters as desired and click the **Save** button.

To immediately start the deploy task, keep the defaults and click **Save**. The Service Request window reappears and the VRF service request moves to the Deployed state.

For steps on how to check the status of the deployed VRF service request, see the information in Migrating PE Devices from IOS to IOS XR, page 5-98 and Monitoring Service Requests, page 8-11.

Modifying VRF Service Requests

To add links or modify existing link attributes for a VRF service request, perform the following steps:

Step 1 Choose **Operate > Service Requests >Service Request Manager** to access the Service Requests window.

Step 2 Choose the VRF service request in the Service Requests window and click **Edit**.

The VRF Service Request Editor window appears.

Step 3 Modify the VRF service request attributes as desired.

Note You can only modify VRF service request links that are not associated with any MPLS VPN links. When you attempt to modify any VRF service request link that is associated with an MPLS VPN link, Prime Fulfillment generates an error while saving the VRF service request.

Step 4 Click **Save** to save your edits.

Decommissioning and Deleting VRF Service Requests

VRF service requests are decommissioned and deleted like other Prime Fulfillment service requests.

Note Decommissioning a VRF service request is not allowed if any of the links in the VRF service request with a VRF object referred in MPLS service request exists.

To decommission a VRF service request, perform the following steps:

Step 1 Choose **Operate > Service Requests >Service Request Manager** to access the Service Requests window.

Step 2 Choose the VRF service request in the Service Requests window and click the **Decommission** button.

The Confirm Request window appears.

Step 3 Click **OK** to confirm.

The Service Request window appears, showing the VRF service request with a DELETE operation type.

Step 4 Deploy the service request with the DELETE operation type, to ensure the successful decommission of the service request.

Searching for VRF Service Requests by VRF Object Name

To search for and display VRF service requests in the Prime Fulfillment repository by VRF object name, perform the following steps:

Step 1 Choose **Operate > Service Requests >Service Request Manager** to access the Service Requests window.

Step 2 Choose **VRF Object Name** in the **Show Services with** drop-down list.

Step 3 Set the **matching** and **of Type** fields as desired.

To search only VRF service requests, choose **VRF** in the **of Type** field.

Step 4 Click **Find** to search for service requests with the associated VRF object name you specified.

Viewing the Configlet Generated by a Deployed VRF Service Request

To view the configlet generated by a deployed VRF service request, perform the following steps:

Step 1 Choose **Operate > Service Requests >Service Request Manager** to view the available service requests.

Step 2 Check the appropriate check box to select the VRF service request for which you want to view the associated configlets.

Step 3 Click the **Details** button.

The Service Request Details window appears.

Step 4 Click the **Configlets** button.

The Service Request Configlets window appears. This window displays a list of devices for which configlets have been generated.

Step 5 To view configlets that were generated for a device, select a device and click the **View Configlet** button.

By default, the latest generated configlet is displayed.

Note If the configlet is deployed on an IOS XR device, you have the option of displaying the configlet in XML or CLI formats or both. For more details on this behavior, see Viewing Configlets on IOS XR Devices, page 8-6.

Step 6 If applicable, you can display configlets for a device based on the time of creation. Choose the desired time of creation in the Create Time list to display a specific configlet based on the time the configlet was generated for the service request.

Step 7 Click **OK** when you are finished viewing the VRF configlet data.

Using VRFs with MPLS VPN Service Requests and Policies

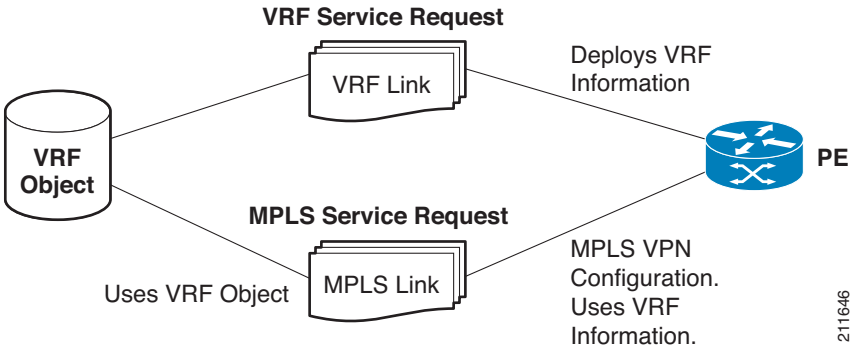
VRF objects which have been deployed can be used within MPLS VPN service requests and service policies.

Note Independent VRF association is not supported for MVRFCE-based policies and service requests.

Relationship of VRF Object and Service Requests and PE Device

Figure 5-2 shows the relationships between the VRF object, MPLS service request, VRF service request, and the PE device. See this figure to understand concepts discussed in the procedures that follow.

Figure 5-2 VRF Object, VRF Service Request, MPLS VPN Service Request, and PE



Specifying VRF Objects within MPLS VPN Service Requests

VRF objects can be selected during the creation of the MPLS VPN service request at the time when the VRF and VPN attributes are set. At that stage, you can either set the VPN attributes individually (as in previous releases of IP Solution Center) or else use an existing VRF object. In the latter case, the MPLS VPN link "inherits" the VPN and VRF data from the VRF object. The VRF object might be either undeployed or deployed. If the VRF object is not deployed, Prime Fulfillment will deploy it automatically. For additional information about the function of VRF objects with MPLS VPN service requests, see Notes On Using a VRF Object in an MPLS Service Request, page 5-29.

To create an MPLS VPN service request using a VRF object, perform the following steps:

Step 1 You must create or use an existing MPLS VPN service request and follow the workflow up to the point where you define the VRF and VPN attributes. This is done in the MPLS Link Editor – VRF and VPN window.

Note If necessary, see the relevant sections of this guide for how to arrive at this window in the MPLS VPN service request workflow.

Step 2 If you do not want to use a VRF object with this MPLS VPN link, leave **Use VRF Object** unchecked.

In this case, set the attributes for the VPN, as normally done with MPLS service requests. These steps are covered in other sections of this guide.

Step 3 To use a VRF object with the MPLS VPN link, check the **Use VRF Object** check box.

All of the standard VPN and VRF attributes, except BGP Multipath Load Sharing, are hidden, and the VRF Object attribute appears.

Step 4 To select a VRF object, click the **Select** button to the right of the VRF Object attribute.

The Select Independent VRF window appears.

This Select Independent VRF window lists all of the VRF objects deployed on the PE, along with their RD value, provider and CERC information.

Step 5 To enable the unique route distinguisher feature, check the **Unique RD** check box.

Note The Unique RD feature is restricted to one MPLS VPN link per MPLS service request. If you select the Unique RD option, it is advised that only one MPLS VPN link is present in that service request.

Be aware of the following use case scenarios when enabling the Unique RD feature:

- If the selected VRF is not deployed on any device, a VRF service request is created for the selected VRF and PE device.
- If the selected VRF is not deployed on the PE device but is deployed on a different PE device, a new VRF object is created (which is a copy of the selected VRF) and a VRF service request is created for the newly created VRF and the PE device.
- If the selected VRF is deployed only on the PE device, then nothing is done. In this case, uniqueness is automatic.
- If the selected VRF is deployed on the PE device and also on some other devices, then a new copy of the VRF object is created with an updated name and a VRF service request is created for the newly created VRF and the PE device.
- It is possible to have two VRFs with the same name but different RDs.

Step 6 Choose the desired VRF Object and click the **Select** button.

Note For information about how the selection of the VRF object is subsequently managed in Prime Fulfillment, see Notes On Using a VRF Object in an MPLS Service Request, page 5-29, following this procedure.

Step 7 Click the **Select** button to confirm the selection of the VRF object and return to the MPLS Link Editor – VRF and VPN window.

Step 8 To set up BGP multipath load sharing, check the **BGP Multipath Load Sharing** check box.

For information on setting the additional attributes, see BGP Multipath Load Sharing and Maximum Path Configuration, page 5-75.

Note Use the **Force Modify Shared Multipath Attributes** attribute to enable forced modification of the shared VRF attributes used by other links. This field is not persisted.

Step 9 Click the **Next** button, if you want to associate templates or data files to the service request.

The Template Association window appears. In this window, you can associate templates and data files with a device by clicking the **Add** button in Template/Data File column for the device. When you click the **Add** button, the Add/Remove Templates window appears. For instructions about associating templates with service requests and how to use the features in this window, see Chapter 9, "Managing Templates and Data Files." When you have completed setting up templates and data files for the service request, click **Finish** in the Template Association window to close it and return to the Service Request Editor window.

Step 10 If you did not add templates, click **Finish** in the MPLS Link Editor – VRF and VPN window.

The MPLS Service Request Editor window appears.

Step 11 Click the **Save** button to complete the creation of the MPLS VPN service request using the VRF object.

The Service Requests window appears showing that the service request is in the Requested state and ready to deploy.

Notes On Using a VRF Object in an MPLS Service Request

Be aware of the following considerations when using VRF objects with MPLS VPN service requests:

- If the selected VRF object is not deployed on the PE device, Prime Fulfillment creates a new VRF service request with the selected VRF object and PE device and deploys it as part of the current MPLS VPN service request deployment process.
- If the VRF object selected in the MPL VPN service request is not deployed on the PE device but a VRF service request exists in the Requested state or any failed states, Prime Fulfillment will attempt to deploy the VRF service request as part of the MPLS VPN service request.
- When decommissioning an MPLS VPN service request for which VRF service requests were created, Prime Fulfillment will not delete the VRF service requests automatically. The user must decommission and deploy such VRF service requests in order to delete the configuration from the device.
- When VRF configuration is selected, no VRF-related information will be provisioned on the device. The VRF name will be use in all the MPLS VPN configuration commands, such as ip vrf forwarding on interface, address family configuration in BGP, OSPF, EIGRP, and so on.

Searching for MPLS VPN Service Requests by VRF Object Name

To search for and display VRF service requests in the Prime Fulfillment repository by VRF object name, perform the following steps:

Step 1 Choose **Operate > Service Requests >Service Request Manager** to access the Service Requests window.

Step 2 Choose **VRF** in the **of Type** drop-down list.

Step 3 Set the **matching** and **of Type** fields as desired.

To search only MPLS VPN service requests, choose **MPLS VPN** in the **of Type** field.

Step 4 Click the **Find** button to search for MPLS VPN service requests with the associated VRF object name you specified.

Specifying VRF Objects within MPLS VPN Service Policies

VRF object selection is supported while defining MPLS VPN policies. This is done during the MPLS VPN policy workflow in the MPLS Policy Editor – VRF and VPN Membership window.

The procedure for using the VRF Object attribute is similar to what is covered in Specifying VRF Objects within MPLS VPN Service Requests, page 5-27. See that section for details on using these attributes.

If you select a VRF object for the MPLS policy, it will subsequently be used by MPLS VPN service requests that use that policy. As per standard Prime Fulfillment policy usage, you can check the **Editable** check box next to the VRF Object attribute to ensure that service requests based on the policy use the same VRF object specified in the policy.

Note If you are not using the independent VRF object feature for the policy, then you must set the VRF and VPN attributes available in the MPLS Policy Editor – VRF and VPN Membership window. See Defining VRF and VPN Information, page 5-72, for more information.

Migrating Existing MPLS VPN Service Requests to the VRF Object Model

Prime Fulfillment provides a migration script to migrate traditional MPLS VPN service requests to the independent VRF model. The script takes as input one or more MPLS VPN service request ID numbers and creates appropriate VRF objects and VRF service requests for each service request. The script is located in the $PRIMEF_HOME/bin directory. The script and its syntax is as follows:

runMplsSRMigration *srid1* [*srid2*] [*srid3*] ...

Where *srid1* is the first MPLS VPN service request ID, [*srid2*] is the second service request, and so on.

Prime Fulfillment performs the following tasks for each MPLS VPN service request passed to the script:

- Creates a VRF object based on the VPN and VRF attributes defined for the service request.
- Copies all the VPN properties to the VRF object.
- Creates a VRF service request, with the VRF object and PE selected in the MPLS VPN link.
- Modifies the MPLS VPN link to point to the VRF object.
- Runs a configuration audit on the VRF service request and the MPLS service request to ensure the correctness of the migration.

IPv6 and 6VPE Support in MPLS VPN

This section provides an overview of IPv6 and 6VPE support in MPLS VPN.

Note For information on how MPLS VPN features are implemented and supported in the Prime Fulfillment GUI, see the appropriate sections of this guide, as indicated by the references provided.

Overview of IPv6 and 6VPE

The Prime Fulfillment MPLS VPN management application supports the configuration and management of Cisco devices running IOS and IOS XR for provisioning of IPv6 VPNs and 6VPEs for Prime Fulfillment Layer 3 VPN services.

Note For the most current information about IOS and IOS XR versions and hardware platforms supporting IPv6, see *Release Notes for Cisco Prime Fulfillment 6.2.*

This section provides an overview of IPv6 and 6VPE technologies. For an overview of how Prime Fulfillment supports IPv6, see MPLS VPN Support for IPv6 and 6VPE, page 5-32.

Internet Protocol Version 6 (IPv6)

IPv6 is an IP protocol designed to replace IPv4, the Internet protocol that is predominantly deployed and extensively used throughout the world. IPv6 quadruples the number of network address bits from 32 bits (in IPv4) to 128 bits, or approximately 3.4×10^{38} addressable nodes. This provides more than enough globally unique IP addresses for every network device on the planet. Cisco Systems has added IPv6 to its Cisco IOS and IOS XR Software. This means that current Cisco Systems-based networks are IPv6-capable, enabling coexistence and parallel operation between IPv4 and IPv6, thereby allowing network managers to configure IPv6 when it is required. While many see IPv6 as a way to build a larger global Internet, it does not eliminate the need to create VPNs for Intranets and other similar applications.

A variety of deployment strategies are available for deploying IPv6 over MPLS backbones. Currently, service providers have two approaches to support IPv6 without making any changes to the current IPv4 MPLS backbones:

- **6PE.** Cisco IOS IPv6 Provider Edge Router (6PE) over MPLS. 6PE lets IPv6 domains communicate with each other over an IPv4 cloud without explicit tunnel setup, requiring only one IPv4 address per IPv6 domain. The 6PE technique allows service providers to provide global IPv6 reachability over IPv4 MPLS. It allows one shared routing table for all other devices.
- **6VPE.** Cisco IPv6 VPN Provider Edge Router (6VPE) over MPLS. This facilitates the RFC 2547bis-like VPN model for IPv6 networks. 6VPE is more like a regular IPv4 MPLS VPN provider edge, with the addition of IPv6 support within Virtual Routing and Forwarding (VRF). It provides logically separate routing table entries for VPN member devices.

MPLS VPN in Prime Fulfillment uses 6VPE to manage Layer 3 VPN services for deployment of IPv6 over a MPLS backbone.

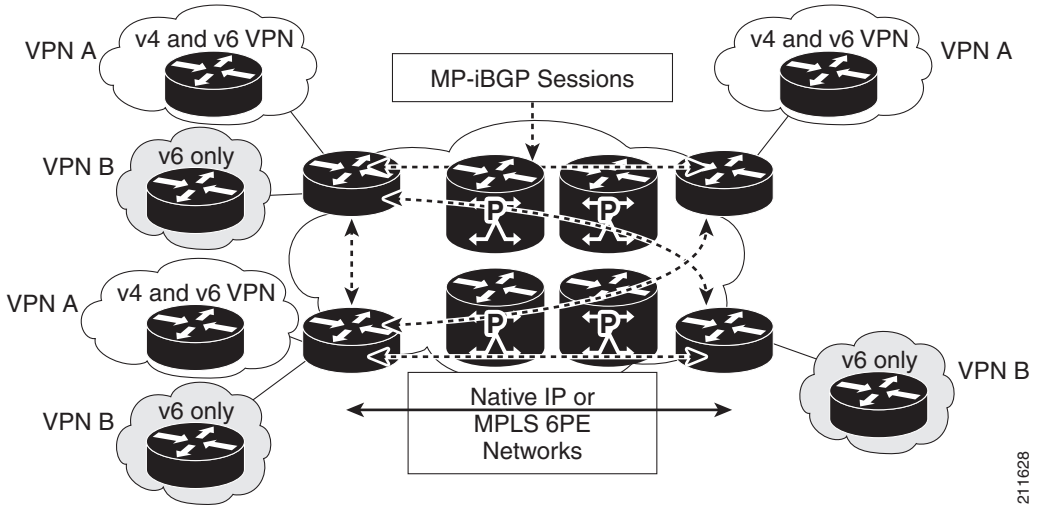
IPv6 VPN Provider Edge Router (6VPE)

Cisco Systems's 6VPE solution smoothly introduces IPv6 VPN service in a scalable way, without any IPv6 addressing restrictions. It does not jeopardize a well-controlled service provider IPv4 backbone or any customer networks. VPN service backbone stability is a key issue for those service providers who have recently stabilized their IPv4 infrastructure. For IPv4 VPN customers, IPv6 VPN service is exactly the same as MPLS VPN for IPv4.

The IPv6 MPLS VPN service model is similar to that of IPv4 MPLS VPNs. Service providers who have already deployed MPLS IPv4 VPN services over an IPv4 backbone can deploy IPv6 MPLS VPN services over the same IPv4 backbone by upgrading the PE router IOS version and dual-stack configuration,

without any change on the core routers. IPv4 services can be provided in parallel with IPv6 services. A PE-CE link can be an IPv4 link, an IPv6 link, or a combination of an IPv4 and IPv6 link, as shown in Figure 5-3.

Figure 5-3 6VPE Deployment



IPv6 VPN service is exactly the same as MPLS VPN for IPv4. 6VPE offers the same architectural features as MPLS VPN for IPv4. It offers IPv6 VPN and uses the same components, such as:

- Multiprotocol BGP (MP-BGP) VPN address family
- Route distinguishers
- VPN Routing and Forwarding (VRF) instances
- Site of Origin (SOO)
- Extended community
- MP-BGP

The 6VPE router exchanges either IPv4 or IPv6 routing information through any of the supported routing protocols, and switches IPv4 and IPv6 traffic using the respective fast switching CEF or distributed CEF path over the native IPv4 and IPv6 VRF interfaces. The 6VPE router exchanges reachability information with the other 6VPE routers in the MPLS domain using Multiprotocol BGP, and shares a common IPv4 routing protocol (such as OSPF or IS-IS) with the other P and PE devices in the domain. Separate routing tables are maintained for the IPv4 and IPv6 stacks. A hierarchy of MPLS labels is imposed on an incoming customer IPv6 packet at the edge LSR:

- Outer label (IGP Label) for iBGP next-hop, distributed by LDP.
- Inner label (VPN Label) for the IPv6 prefix, distributed by MP-BGP.

Incoming customer IPv6 packets at the 6VPE VRF interface are transparently forwarded inside the service provider's IPv4 core, based on MPLS labels. This eliminates the need to tunnel IPv6 packets. P routers inside the MPLS core are unaware that they are switching IPv6 labelled packets.

MPLS VPN Support for IPv6 and 6VPE

This section summarizes how the MPLS VPN management application supports IPv6 and 6VPE.

See Setting Up the Prime Fulfillment Services, page 5-4 for information setting up Prime Fulfillment services mentioned in this section.

IOS and IOS XR Support for IPv6

IPv6 services are available in Prime Fulfillment for supported versions of IOS and IOS XR and hardware platforms for both PE and CE roles.

Note For the most current information about IOS and IOS XR versions and hardware platforms supporting IPv6, see *Release Notes for Cisco Prime Fulfillment 6.2.*

The IPv6 features described in the following sections are supported for both IOS and IOS XR devices, unless otherwise noted.

Inventory and Device Management

To activate MPLS VPN services, you must configure Prime Fulfillment so it "knows" about the preconfiguration information, such as devices, providers, customers, and so on, that Prime Fulfillment is going to manage. Prime Fulfillment features that support inventory and device management for IPv6 and 6VPE include:

Discovery:

- Prime Fulfillment Inventory Manager supports bulk-import of 6VPE devices into the Prime Fulfillment repository.

Collect Config Task:

- The Collect Config task retrieves the OS type and the version information. If the device is a Cisco 12000 Series router, Cisco CRS-1 Carrier Routing System, or ASR 9000 Series router and is running IOS XR, the device will be marked as 6VPE supported. (By default, the "6VPE" check box in the Create PE Device window will be checked for XR devices). The "6VPE" check box in the Create PE Device window must be checked manually to designate an N-PE device as 6VPE for IOS devices.
- The Collect Config task for an IOS device with IPv6 services is the same as for IPv4 IOS devices.

Device Configuration:

- 6VPE devices with IPv6 addressing can be created and managed in the Prime Fulfillment GUI.
  - A "6VPE" check box in the Create PE Device window must be checked to designate an N-PE device as a 6VPE. IPv6 services for IOS and IOS XR devices are only available in MPLS and VRF service requests if this check box is checked.

    Note If the 6VPE check box is checked for a device in the Prime Fulfillment GUI and the device does not actually support IPv6 services, MPLS VPN service requests deployed on that device will result in a Failed Deploy state.

  - A column in the Interface Attributes window shows IPv6 addresses. It is not possible to bulk change the IPv6 addresses by selecting multiple interfaces. The IPv6 Address column is noneditable.
  - The Edit Device Interface window shows IPv6 addresses on interfaces. In case of dual-stack interfaces containing both IPv4 and IPv6 addresses, both addresses are displayed.

- Prime Fulfillment supports multiple IPv6 addresses on the PE interface for IOS XR PE and IOS 6VPE devices.
- The Create CPE Device window displays IPv6 addresses on interfaces. In case of dual-stack interfaces containing both IPv4 and IPv6 addresses, both addresses are displayed.
- You cannot create an IPv6 interface using the existing Create Interface feature. This screen currently lets you create interfaces in the repository only, with the device configuration remaining unchanged. This feature does not support IPv6 addresses. The IPv6 interface creation in the device is supported through the MPLS VPN service deployment.

VPN Creation and Configuration

There are no changes in the Prime Fulfillment VPN workflow for IPv6 and 6VPE.

Multicast VPN support for IPv6 is not available on IOS devices this release. Currently, it is only available for supported IOS XR devices. See the following sections for more information:

- Multicast Routing on IOS and IOS XR Devices, page 5-36
- Multicast Support for IPv6 (IOS XR Only), page 5-37

Independent VRF Object Support

Prime Fulfillment allows you to specify VPN and VRF information in an independent VRF object, which is subsequently deployed to a PE device and then associated with an MPLS VPN link via an MPLS VPN service request. Prime Fulfillment supports IPv4, IPv6, and dual-stack addressing in VRF objects.

For details on using creating and managing independent VRF objects, see Independent VRF Management, page 5-14

Resource Pools

Prime Fulfillment uses resource pools to automatically assign critical parameters like VLAN, VCID, and IP Addresses during the service provisioning. IPv6 address pools are not supported in this release.

MPLS VPN Service Provisioning

Prime Fulfillment MPLS VPN management application supports the provisioning of IPv6 Layer 3 VPNs on an IPv6 Provider Edge router (6VPE). Prime Fulfillment provides the ability to configure the following on the 6VPE:

- Use IPv6 addressing on 6VPE (optionally, IPv4, IPv6, or both IPv6+IPv4 addresses).
- Assign a static route to the 6VPE facing interface on a CE device.
- Enable MP-BGP peering with target 6VPE.
- Redistribute connected (if needed).

The following sections describe features of MPLS VPN policy definition, service request creation, and service request auditing to support IPv6 and 6VPE in Prime Fulfillment.

MPLS VPN Policies

Support for MPLS VPN policy definition for IPv6 and 6VPE includes:

- MPLS VPN service policy design supports the configuration of IPv6 on a 6VPE router for the following policy types:

  - Regular: PE-CE (with unmanaged CE)
  - Both Unmanaged CE and no-CE scenarios are supported for IPv6.
- Service policies support the following addressing schemes:
  - IPv4
  - IPv6
  - Dual-stacked (both IPv4 and IPv6)
- The IP Numbering Scheme field in the MPLS Policy Editor - IP Address Scheme window allows you to specify each of the supported address schemes.
- IPv4 routing and IPv6 routing are independent. The Prime Fulfillment GUI allows you to input the same or different routing protocols for IPv4 and IPv6.
- When setting up the policy, the following PE-CE routing protocols are supported for the IPv6 addressing scheme:
  - Static
  - BGP
  - EIGRP (only supported for IOS XR devices)
  - None
- IPv6 multicast VPNs are not supported for IOS 6VPE configurations. For information on support for multicast VPNs for IOS XR devices, see Multicast Routing on IOS and IOS XR Devices, page 5-36.
- IPv6 validity checks. The following checks will be performed on addresses entered in the IPv6 address fields:
  - The address can be specified eight consecutive blocks of 16-bit each separated by the ":" (colon) character. Each 16-bit block can be specified as 4-digit hexadecimal number. Example: 21DA:00D3:0000:2F3B:02AA:00FF:FE28:9C5A.
  - The leading zeros can be skipped in each hexadecimal block. Here is the modified valid address from the previous example: 21DA:D3:0:2F3B:2AA:FF:FE28:9C5A.
  - Where there are consecutive "0:" blocks, they can be replaced with "::". Example: 21DA:D3:0:0:0:FF:FE28:9C5A can be represented as 21DA:D3::FF:FE28:9C5A.
  - The string "::" cannot appear more than once in the address. Example: 21DA:0000:0000:2F3B:0000:0000:0000:9C5A can be represented as 21DA::2F3B:0000:0000:0000:9C5A or 21DA:0000:0000:2F3B::9C5A, but not as 21DA::2F3B::9C5A.

See MPLS VPN Service Policies, page 5-40 for information on defining MPLS VPN service policies.

MPLS VPN Service Requests

Attributes set during MPLS VPN policy creation to support IPv6 and 6VPE are carried over to the corresponding windows in the service request creation workflow. If the options were set as editable during policy creation, they can be modified when the service request is created.

- The IP Numbering Scheme field in the MPLS Link Attribute Editor - IP Address Scheme window allows you to specify each of the supported address schemes.
- The IPv4 and IPv6 Unnumbered schemes are not supported on IOS XR devices. When you select an IOS XR (or IOS 6VPE) device and go the to IP Addressing Scheme window, only the following options are displayed:

  - IPv4 Numbered
  - IPV6 Numbered
  - IPV4+IPV6 Numbered
- As part of the regular PE-CE MPLS service, the required VRF will be configured on the PE device. The CE-facing interface will be configured with the IPv6 address and the interface will be assigned to the VRF. The IPv6 address-family configuration in BGP along with the PE-CE routing information will be configured.
- If the PE Interface is dual-stacked (contains both IPv4 and IPv6 addresses), you can enter the routing information for both IPv4 and IPv6 independently. The GUI provides steps to enter the IPv6 routing information in addition to the existing IPv4 routing information.
- Prime Fulfillment supports the scenario of the CE device not present in the service request. This release also supports the Unmanaged CE devices being present in the service request. In the later case, the configlets for service provisioning will be generated but not rolled onto the CE device.
- It is possible to modify a 6VPE service request.
- If the PE device is an IOS XR device, all of the configuration operations will be performed using the IOS XR interface.
- For IOS XR 6VPE devices, all configlets generated are in XML format. Different versions of IOS XR will generate different XML configlets. However, the configurations will be almost identical, except for changes in the XML schema.
- For IOS 6VPE devices, all configurations are generated in CLI format.

See MPLS VPN Service Requests, page 5-78 and subsequent chapters in this guide for information on creating MPLS VPN service requests.

MPLS Service Request Audits

L3 VPN functional audit supports IPv6 VPNs (IPv6 addresses and 6VPE devices). This includes checking the routes to remote CEs in the VRF route tables on the PE devices. See Viewing Audit Reports Service Requests, page 8-4, for information on auditing service requests.

Multicast Routing on IOS and IOS XR Devices

Multicast VRF deployments for IOS XR devices are supported for IPv4, IPv6, IPv4+IPv6 services. Currently, multicast on IOS XR is supported only for specified versions of IOS XR versions. For a list of supported IOS XR versions in this release, see *Release Notes for Cisco Prime Fulfillment 6.2.*

This section describes how Prime Fulfillment supports multicast routing on IOS XR devices. There are no changes in the GUI (Create VPN window) to support this feature. The IOS XR XML does not support multicast routing command, so the corresponding IOS XR CLI is used to push the configuration to the device.

The following sections shows an example of the relevant IOS commands and the corresponding IOS XR commands to enable multicast routing.

IOS Commands

The following is a sample IOS configuration:

```
ip vrf V27:MulticastCERC3
rd 100:124
address-family ipv4
route-target import 100:406
route-target import 100:407
```

```
route-target export 100:406
mdt default 226.2.3.4
mdt data 226.5.6.7 0.0.0.15 2000
mdt mtu 2000
ip multicast-routing vrf V27:MulticastCERC3
ip pim vrf V28:VPN13 ssm default
ip pim vrf V27:MulticastCERC3 rp-address 10.20.1.1
ip pim vrf V27:MulticastCERC3 rp-address 10.20.3.1 test2
ip pim vrf V27:MulticastCERC3 rp-address 10.20.2.1 test1 override
```

IOS XR Commands

The following IOS commands are not supported on the IOS XR devices, because the corresponding commands do not exist in IOS XR.

- **ip multicast vrf <vrfName> route-limit**. The reason for not supporting this is that the command to set the route limit per VRF is not available on IOS XR devices.
- **ip pim vrf <vrfName> sparse-dense-mode**. Sparse-dense mode is not supported by IOS XR. Only sparse mode and bidirectional modes are supported.

The following IOS commands are enabled on the IOS XR device by default when the multicast routing is enabled. They cannot be disabled.

- **ip pim vrf <vrfName> sparse-mode**
- **ip pim vrf <vrfName> ssm default**
- **ip pim vrf <vrfName> autorp listener**

Multicast Support for IPv6 (IOS XR Only)

Multicast on IPv6 is only supported on IOS XR devices. Specifically, in this release this feature is only supported on Cisco 12000 series routers. Prime Fulfillment allows the following on supported PE devices and versions of IOS XR:

- A multicast VPN to be deployed on an IPv6 PE-CE link.
- Multicast to be enabled during the creation of the VRF object.

When creating a VPN or a VRF object, you can enable multicast for IPv4, IPv6, or both. You can enter IPv6 addresses as static Rendezvous Point (RP) addresses if IPv6 multicast is enabled during the creation of a VPN or VRF object.

You can also modify an existing VPN or VRF object to enable multicast for IPv4, IPv6, or both. When IPv4 multicast is enabled, all deployed service requests containing IPv4 links of the same VPN or VRF are moved into Requested state.

In addition, you can specify within the MPLS service request whether you want to enable multicast for IPv4, IPv6, or both on a given MPLS link.

When IPv6 multicast is enabled, all deployed service requests containing IPv6 links of the same VPN or VRF are moved into Requested state. If IPv4 is previously configured and only IPv6 multicast is enabled in a VPN, only the service requests with IPv6 links are moved into Requested state.

You can modify an existing VPN or VRF object and add IPv6 static RP addresses when IPv6 multicast is enabled. Any service requests already in Deployed state are then moved to the Requested state.

You can create a service policy or an MPLS VPN link in the service request with IPv6 Numbered or IPv4+IPv6 Numbered as the IP addressing scheme and a multicast VPN or a VRF with multicast enabled.

DCPL Properties Updated for IOS 6VPE Support

Two DCPL properties have been updated to support certain IOS commands that require a delay after being downloaded to a device. This may cause a delay when deploying MPLS VPN service requests on IOS devices containing IPv6 configuration commands.

- The DCPL property GTL/CSL/ios/delayAfterDownloadingCmd has been added to Prime Fulfillment to support IOS commands that require a delay after they are downloaded via a terminal session protocol such as Telnet. The List element format is:

  cmd_regex:delay_in_seconds; no vrf definition *:105

  After the "no vrf definition" command is pushed to the device, there is a delay of 105 seconds before it takes effect on the device.

- The DCPL property GTL/CSL/ios/delayBeforeDownloadingCmd has been added to Prime Fulfillment to support certain IOS commands that require a delay before they are downloaded via a terminal session protocol such as Telnet. The List element format is:

  cmd_regex:delay_in_seconds;
  vrf definition *:70;

  After the "vrf definition" command is pushed to the device, there is a delay of 70 seconds before it takes effect on the device.

MPLS Reports

MPLS VPN reports support IPv6 addresses and 6VPE devices. See Generating MPLS Reports, page 10-40 for information on generating MPLS VPN reports for IPv6 and 6VPE.

Upgrading an Existing IPV4 VRF to Be a Dual-Stack (IPV4+IPV6) VRF

This section describes VRF upgrading on IOS 6VPE devices using MPLS service requests. Key points to keep in mind are as follows:

- This feature is only supported for IOS 12.2(33) SRE2 version and above.
- Any IPv4 deployment on a VRF always generates the command "ip vrf vrf-name" on the device. When it is upgraded to dual stack (IPv4+Ipv6) or IPv6, then:
  - Any links sharing the same VRF on the same device are upgraded to "vrf definition vrf-name" in the device.
  - All the related service requests sharing the same VRF on the same device are moved to the Requested state.
  - All service requests have to be redeployed for an audit pass.
- The VRF upgrade scenarios from Prime Fulfillment work for IOS 6VPE devices only if the "vrf upgrade-cli multi-af-mode non-common-policies vrf vrf-name force" command is supported in the device. If not the service request results in FAILED-DEPLOYED state. This command is available in IOS version 12.2 (33) SRE2.
- Most upgrade scenarios will likely involve starting with existing IPv4 service requests, rather than starting from scratch with IOS-based IPv6. The scenarios below cover various upgrade scenarios for the typical cases.

The following are typical VRF modification scenarios:

- IPv4 to Dual-Stack (IPv4+IPv6). Configlets are generated for the IPv6 link. The command “ip vrf vrf-name ” is upgraded to “vrf definition vrf-name” by using the command “vrf upgrade-cli multi-af-mode non-common-policies vrf vrf-name force”.
- IPv4 to IPv4. There is no change in the configlets.
- IPv4 to IPv6. “No” commands (“no ip vrf vrf-name”) are generated on the IPv4 link, and new configlets (“vrf definition vrf-name”) get deployed on the IPv6 link.
- IPv6 to IPv4. “No” commands (“no vrf definition vrf-name”) are generated on the IPv6 link, and new configlets (“ip vrf vrf-name”) are issued for the IPv4 link.
- Rehoming (that is, moving from one PE to another) issues “no” commands on the old device and new commands on the rehomed PE.

An example VRF modification scenario is provided below for reference.

An IPv4 link has VRF configured as:

```
ip vrf V8:stellavpn8
 rd 64512:1572
 route-target export 64512:15870
 route-target import 64512:15870
 route-target import 64512:15871
!
```

An IPv6 link has VRF configured as:

```
vrf definition V4:stellavpn4
 rd 64512:1568
 !
 address-family ipv6
 route-target export 64512:15862
 route-target import 64512:15862
 exit-address-family
!
```

An IPv4+IPv6 link (which has been upgraded from IPv4 to dual-stack) has VRF configured as:

```
vrf upgrade-cli multi-af-mode non-common-policies vrf V9:stellavpn9 force !
vrf definition V9:stellavpn9
rd 64512:1573
!
address-family ipv4
route-target export 64512:15872
route-target import 64512:15872
route-target import 64512:15873
exit-address-family
!
address-family ipv6
route-target export 64512:15872
route-target import 64512:15872
route-target import 64512:15873
exit-address-family
```

Unsupported IPv6 and 6VPE Features

The following features are **not** supported for IPv6 and 6VPE:

- Discovery of existing IPv6 VPN services on the device.
- IPv6 addressing as part of a CPE device definition and configuration.
- IPv6 address pools.

- IPv6 multicast address pools.
- The IPv4 and IPv6 Unnumbered address schemes are not supported for 6VPE and IOS XR.
- Grey management VPN support for 6VPE and IOS XR.
- Staging service request deployment to support eBGP route maps on IOS XR devices.
- Managed CE services (if the device does not support IPv6 services).
- Multi-VRF CE (MVRFCE) support.
- One-time setup operations on the 6VPE device like enabling IPv6 routing, BGP VPNv6 configuration.
- Tunnel interface. An IPv6 address cannot be specified as the Tunnel Source Address value.

MPLS VPN Service Policies

This section describes how to use the Cisco Prime Fulfillment GUI to define MPLS VPN Service Policies. You can also associate Prime Fulfillment templates and data files with a policy. See Chapter 9, "Managing Templates and Data Files." for more information about using templates and data files in policies.

It is also possible to create user-defined attributes within a policy (and service requests based on the policy). For background information on how to use the additional information feature, see Appendix H, "Adding Additional Information to Services."

Service Policy Overview

Provisioning an MPLS VPN begins with defining a service policy. A service policy can be applied to multiple PE-CE links in a single service request. A *network operator* defines service policies. A *service operator* uses a service policy to create service requests. Each service request contains a list of PE-CE links. When a service operator creates a service request, the operator sees only the policy information required to be completed. All the other necessary information is filled in by the service policy itself (as well as the Auto Discovery process).

Service Policy Editor

When you define a service policy for Prime Fulfillment, you are presented with a series of dialog boxes that allow you to specify the parameters for each major category required to complete an MPLS service request. The Service Policy editor presents three columns: **Attribute**, **Value**, and **Editable**:

- **Attribute**

  The Attribute column displays the names of each parameter that you need to define for each major category (for example, IP addresses or routing protocols).

- **Value**

  The Value column displays the fields and other selectable items that correspond to each parameter and option.

  The type of dialog box that is invoked when you edit an attribute depends on the type of attribute. In some cases, the value is a simple string value or integer value, in which case a single text entry field appears. In other cases, the value is complex or consists of multiple values, such as an IP address. In these cases, a dialog box appears so you can specify the required values. The values you

enter are validated; when invalid values are entered, you receive notification of the invalid values. In other cases, you will be presented with check boxes that will allow you to enable or disable a particular option.

Note In some cases, changing an attribute's value results in invalidating the values of related attributes. For example, changing the PE interface name can result in invalidating the PE encapsulation value. When this occurs, the service policy editor removes the invalid values and you will need to reset them appropriately.

There is a parent-child relationship between some attributes. In these cases, changing the value of a parent attribute can enable or disable the child attributes. For example, changing the value of the PE encapsulation could result in enabling or disabling the DLCI (data link connection identifier), VLAN ID, ATM circuit identifiers, and the tunnel source and destination address attributes.

- **Editable**

  The Editable column allows the network operator to indicate the attributes that are likely to change across multiple service requests. When attributes are checked as editable, only those attributes will be made available to the service operator when creating or modifying service requests with that service request policy.

  When an attribute category is set to be editable, all the related and child attributes are also editable attributes.

About IP Addresses in Cisco Prime Fulfillment

Within a VPN (or extranet), all IP addresses must be unique. Customer IP addresses are not allowed to overlap with provider IP addresses. Overlap is possible only when two devices cannot see each other; that is, when they are in isolated, non-extranet VPNs.

The Prime Fulfillment MPLS VPN software assumes that it has an IP address pool to draw addresses from. The only way to guarantee that the product can use these addresses freely is if they are provider IP addresses.

Predefining a unique section (or sections) of IP address space for the PE-CE links is the only way to ensure stable security. Thus, because of the security and maintenance issues, we do not recommend using customer IP addresses on the PE-CE link.

Defining an MPLS VPN Service Policy

The remaining sections in this section provide an extended example of defining an MPLS service policy for a PE-CE link. This is to demonstrate the various steps involved in defining an MPLS service policy. The steps can be used as the basis for defining other types of MPLS VPN service policies. Additional types of MPLS VPN policies are described in other chapters in this guide.

To begin defining an MPLS VPN service policy for PE-CE link, perform the following steps:

Step 1 Choose the **Service Design > Policies > MPLS**.

The MPLS Policy Editor - Policy Type window appears.

Step 2 Enter a **Policy Name** for the MPLS policy.

Step 3 Choose the **Policy Owner.**

There are three types of MPLS policy ownership:

- Customer ownership
- Provider ownership
- Global ownership: Any service operator can make use of this MPLS policy.

This ownership has relevance when the Prime Fulfillment Role-Based Access Control (RBAC) comes into play. For example, an MPLS policy that is customer-owned can only be seen by operators who are allowed to work on this customer-owned policy.

Similarly, operators who are allowed to work on a provider's network can view, use, and deploy a particular provider-owned policy.

Note For Cable (PE-NoCE), policy ownership should be set to Provider.

Step 4 Click **Select** to choose the owner of the MPLS policy. (If you choose Global ownership, the Select function is not available.)

The Select Customer window or the Select Provider window appears and you can choose an owner of the policy and click **Select**.

Step 5 Choose the **Policy Type** of the MPLS policy.

There are two policy types for MPLS policies:

- Regular PE-CE: PE-to-CE link
- MVRFCE PE-CE: PE to CE link using the Multi-VRF feature for the PE

Step 6 Check the **CE Present** check box if you want Prime Fulfillment to ask the service operator who uses this MPLS policy to provide a CE router and interface during service activation. The default is CE present in the service.

If you do not check the **CE Present** check box, Prime Fulfillment asks the service operator, during service activation, only for the PE-CLE or the PE-POP router and customer-facing interface.

Step 7 Click **Next**.

To continue with the example, see the following section, Specifying PE and CE Interface Parameters, page 5-42.

Specifying PE and CE Interface Parameters

To specify the PE, UNI Security, and CE interface information for this MPLS policy follow these steps:

Tip You do not have to choose a specific interface type for the PE and CE at this point. Notice that the fields are set by default to **Editable**. With the interface parameters set to **Editable**, the service operator can specify the exact interface type and format when he or she creates the service request.

If you want to specify the device interface information for this service policy when the service request is created, leave the fields as they are currently set by default, then click **Next**.

PE Information

Step 1 **Interface Type:** From the drop-down list, choose the interface type for the PE.

Cisco IP Solution Center supports the following interface types (for both PEs and CEs):

- Any
- ATM (Asynchronous Transfer Mode)
- BRI (Basic Rate Interface)
- Bundle-Ether. (For additional information, see Step 2Interface Format: Optionally, you can specify the slot number and port number for the PE interface., page 5-43.)
- Ethernet
- Fast Ethernet
- FDDI (Fiber Distributed Data Interface)
- GE-WAN (Gigabit Ethernet WAN)
- Gigabit Ethernet
- HSSI (High Speed Serial Interface)
- Loopback
- MFR
- MultiLink
- PoS (Packet over Sonet)
- Port-Channel
- Serial
- Switch
- Tunnel
- VLAN

Step 2 **Interface Format:** Optionally, you can specify the slot number and port number for the PE interface.

Specify the format in the standard nomenclature: **slot number/port number** (for example, **1/0** indicates that the interface is located at slot 1, port 0).

This is especially useful to specify here if you know that the link will always go through a particular interface's slot/port location on all or most of the network devices in the service. If this parameter is left editable, it can be changed when the service operator creates the service request.

You can also specify the Interface Format as a Channelized Interface:

- **slot/subSlot/port** (for example, **2/3/4** indicates that the interface is located at Serial 2/3/4)
- **slot/subSlot/port/T1#:channelGroup#** (for example, **2/0/4/6:8** indicates that the interface is located at Serial 2/0/4/6:8)
- **slot/subSlot/port.STS-1Path/T1#:channelGroup#** (for example, **2/0/0.1/6:8** indicates that the interface is located at Serial 2/0/0.1/6:8)

Step 3 **Interface Description:** Optionally, you can enter a description of the PE interface.

Step 4 **Shutdown Interface:** When you check this check box, the specified PE interface is configured in a shut down state.

Step 5 **Encapsulation:** Choose the encapsulation used for the specified PE interface type.

When you choose an interface type, the Encapsulation field displays a drop-down list of the supported encapsulation types for the specified interface type.

Table 5-2 shows the protocol encapsulations available for each of the supported interface types.

Table 5-2 Interface Types and Their Corresponding Encapsulations

Interface Type	Encapsulations
ATM	AAL5SNAP
BRI	Frame-Relay, Frame-Relay-ietf, HDLC (High-Level Data Link Control), PPP (Point-to-Point Protocol). **Frame-Relay-ietf** sets the encapsulation method to comply with the Internet Engineering Task Force (IETF) standard (RFC 1490). Use this method when connecting to another vendor's equipment across a Frame Relay network.
Bundle-Ether	Default frame, dot1q (802.1Q)
Ethernet	Default frame, dot1q (802.1Q)
Fast Ethernet	Default frame, ISL (Inter-Switch Link), dot1q (802.1Q)
FDDI (Fiber Distributed Data Interface)	None
Gibabit Ethernet	Default frame, ISL (Inter-Switch Link), dot1q (802.1Q)
Gigabit Ethernet WAN	Default frame, ISL (Inter-Switch Link), dot1q (802.1Q)
HSSI (High Speed Serial Interface)	Frame-Relay, Frame-Relay-ietf, HDLC (High-Level Data Link Control), PPP (Point-to-Point Protocol)
Loopback	None.
MFR	Frame-Relay, Frame-Relay-ietf, HDLC (High-Level Data Link Control), PPP (Point-to-Point Protocol).
MultiLink	PPP (Point-to-Point Protocol)
Port-Channel	Default frame, ISL (Inter-Switch Link), dot1q (802.1Q) NOTE: [Andrew to provide content]
POS (Packet Over Sonet)	Frame-Relay, HDLC (High-Level Data Link Control), PPP (Point-to-Point Protocol)
Serial	Frame-Relay, Frame-Relay-ietf, HDLC (High-Level Data Link Control), PPP (Point-to-Point Protocol)
Switch	AAL5SNAP
Tunnel	GRE (Generic Routing Encapsulation) **- GRE is not supported in this release. -**
VLAN	None

Note MLFR interfaces are supported on IOS and IOS XR devices. Prime Fulfillment does not set up the MLFR interface. Prime Fulfillment provisions the Layer 3 service on the MLFR interface.

Step 6 **Auto-Pick VLAN ID:** Check this check box to have Prime Fulfillment automatically pick the VLAN ID.

Note If Auto-Pick VLAN ID is unchecked, you are prompted to enter the VLAN ID during the creation of the service request based on the policy.

Step 7 **Use SVI:** Check this check box to have Prime Fulfillment terminate VRF on SVI.

Step 8 **ETTH Support:** Check this check box to configure Ethernet-To-The-Home (ETTH). For an explanation of ETTH, see Ethernet-To-The-Home (ETTH), page 5-147.

Step 9 Standard UNI Port: Check this check box to access UNI Security Parameters:

UNI Security Information

Step 10 **Disable CDP:** Check this check box to disable CDP.

Step 11 **Filter BPDU:** Check this check box to filter BPDU.

Step 12 **Use existing ACL Name:** Check this check box to use existing ACL name.

Step 13 **UNI MAC Addresses:** Click **Edit** to modify or create a MAC address record.

Step 14 **UNI Port Security:** Check this check box to access UNI Port Security parameters:

a. **Maximum MAC Address:** Enter a valid value.

b. **Aging (in minutes):** Enter a valid value.

c. **Violation Action:** From the drop-down list, choose one of the following:

- PROTECT
- RESTRICT
- SHUTDOWN

d. **Secure MAC Address:** Click **Edit** to modify or create a secure MAC address record.

CE Interface Information

Step 15 **Interface Type:** From the drop-down list, choose the interface type for the CE.

Step 16 **Interface Format:** Optionally, you can specify the slot number and port number for the CE interface.

Step 17 **Interface Description:** Optionally, you can enter a description of the CE interface.

Step 18 *Encapsulation:* Choose the encapsulation used for the specified CE interface type.

Step 19 When satisfied with the interface settings, click **Next**.

To continue with the example, see the following section, Specifying the IP Address Scheme, page 5-45.

Specifying the IP Address Scheme

To specify the IP address scheme you want to use for this service policy, perform the following steps:

Step 1 Define the IP addressing scheme that is appropriate for the PE-CE link.

IP Numbering Scheme

You can choose from the following options.

- **IPv4 Numbered**

  If you choose **IPv4 Numbered** and also check the **Automatically Assign IP Address** check box, Prime Fulfillment: MPLS checks for the presence of the corresponding IP addresses in the router's configuration file. If the addresses are present and they are in the same subnet, Prime Fulfillment uses those addresses (and does not allocate them from the address pool). If the IP addresses are not present in the configuration file, Prime Fulfillment picks IPv4 addresses from a /30 subnet point-to-point IP address pool.

- **IPv4 Unnumbered**

  IPv4 addresses are drawn from the loopback IPv4 address pool. An unnumbered IPv4 address means that each interface "borrows" its address from another interface on the router (usually the loopback interface). Unnumbered addresses can only be used on point-to-point WAN links (such as Serial, Frame, and ATM), not on LAN links (such as Ethernet). If using IP unnumbered, then both the PE and CE must use the same IP unnumbered addressing scheme. When you choose **IPv4 Unnumbered**, Prime Fulfillment: MPLS creates a static route for the PE-CE link.

  When you choose **IPv4 Unnumbered**, Prime Fulfillment: MPLS automatically creates a loopback interface (unless a loopback interface already exists with the correct attributes). For related information, see Using Existing Loopback Interface Number, page 5-47.

- **IPv6 Numbered**

  This addressing scheme is provided to support a 6VPE router. See IPv6 and 6VPE Support in MPLS VPN, page 5-30 for more information on IPv6 and 6VPE support in MPLS VPN management.

  Note This option only appears if the policy type is a regular PE-CE policy.

- **IPv4+IPv6 Numbered**

  In the case of a 6VPE device, the PE interface can be "dual stacked," meaning it can contain both IPv4 and IPv6 addresses. In later steps, you will be able to enter the routing information independently for both IPv4 and IPv6. See IPv6 and 6VPE Support in MPLS VPN, page 5-30 for more information on IPv6 and 6VPE support in MPLS VPN management.

  Note This option only appears if the policy type is a regular PE-CE policy.

Step 2 Indicate whether an extra loopback interface is required for the CE.

Extra CE Loopback Required

Even though a numbered IP address does not require a loopback address, Prime Fulfillment software provides the option to specify than an extra CE loopback interface is required. This option places an IP address on a CE router that is not tied to any physical interface.

If you enable **Extra CE Loopback Required**, you can enter the CE loopback address.

Step 3 Specify whether you want to automatically assign IP addresses.

Automatically Assign IP Address

If you choose **IPv4 Unnumbered** and also check the **Automatically Assign IP Address** check box, Prime Fulfillment picks two IP addresses from a /32 subnet point-to-point IP address pool.

If you choose **IPv4 Numbered** and also check the **Automatically Assign IP Address** check box, Prime Fulfillment checks for the presence of the corresponding IP addresses in the router's configuration file. If the addresses are present and they are in the same subnet, Prime Fulfillment uses those addresses (and does not allocate them from the address pool). If the IP addresses are not present in the configuration file, Prime Fulfillment picks IP addresses from a /30 subnet point-to-point IP address pool.

Note This option is not supported for the **IPv6 Numbered** and **IPv4+IPv6 Numbered** address schemes.

Step 4 Specify the IP address pool and its associated Region for this service policy.

IP Address Pool

The IP Address Pool option gives the service operator the ability to have Prime Fulfillment automatically allocate IP addresses from the IP address pool attached to the Region. Prior to defining this aspect of the service policy, the Region must be defined and the appropriate IP address pools assigned to the Region.

You can specify IP address pool information for point-to-point (IP numbered) PE-CE links.

IP unnumbered addresses are drawn from the loopback IP address pool. An unnumbered IP address means that each interface "borrows" its address from another interface on the router (usually the loopback interface). Unnumbered addresses can only be used on point-to-point WAN links (such as Serial, Frame, and ATM), not on LAN links (such as Ethernet). If using IP unnumbered, then both the PE and CE must use the same IP unnumbered addressing scheme.

Note This option is not supported for the IPv6 Numbered and IPv4+IPv6 Numbered address schemes.

Step 5 When satisfied with the IP address scheme, click **Next**.

Using Existing Loopback Interface Number

On each PE, there is usually only one loopback interface number per VRF for interfaces using IP unnumbered addresses. However, if provisioning an interface using IP unnumbered addresses and manually assigned IP addresses, it is possible to have more than one loopback interface number under the same VRF. When using automatically-assigned IP addresses for provisioning IP unnumbered addresses, Prime Fulfillment associates the first loopback number with the same VRF name to the interface. If no loopback number already exists, Prime Fulfillment creates one.

If a service provider wants Prime Fulfillment to use an existing loopback interface number (for example, Loopback0), the service provider must modify the loopback interface description line in the configuration files for the pertinent routers (PE or CE).

To use the existing loopback interface number, you must modify the loopback interface description line so that it includes the keyword **VPN-SC**, as shown in the following example of a router configuration file.

Note When using an existing loopback interface number on a PE, an additional command line with the **ip vrf forwarding** *VRF_name* command must be included directly after the "description" line.

```
interface Loopback0
description by VPN-SC
```

```
ip vrf forwarding <VRF_name> ; This line is required on the PE only
ip address 209.165.202.129 255.255.255.224
```

You can use an existing loopback interface number only when the interface configuration meets these conditions: it must be a WAN serial interface using IP unnumbered addresses.

Prime Fulfillment selects loopback interface numbers by sequence. Prime Fulfillment uses the first loopback interface number that meets the requirement—for a CE, it is inclusion of the VPN-SC keyword; for a PE, it is the matching VRF name.

For example, if loopback1 and loopback2 include the VPN-SC keyword, but loopback3 does not, adding the VPN-SC keyword to loopback3 will not force Prime Fulfillment to choose loopback3 for the unnumbered interface when using automatically assigned addresses. Loopback1 will be chosen instead. The only way to choose a specific loopback interface number is to use a manually assigned IP address that matches the desired loopback interface number.

Note Unlike standard interfaces, when loopback interfaces are provisioned in Prime Fulfillment, the resulting configuration file does not include a service request (SR) ID number. This is because multiple interfaces or service requests can use the same loopback interface.

To continue with the example, see the following section, Specifying the Routing Protocol for a Service, page 5-48.

Specifying the Routing Protocol for a Service

You can now specify the routing protocol information for this service policy.

Note IPv4 and IPv6 routing are independent. The Prime Fulfillment GUI allows you to input the same or different routing protocols for IPv4 and IPv6, depending upon which addressing scheme you selected. Not all routing protocols are supported for IPv6. See IPv6 and 6VPE Support in MPLS VPN, page 5-30 for more information IPv6 and supported routing protocols.

The routing protocol you choose must run on both the PE and the CE. You can choose any one of the following protocols:

- Static—Specifies a static route (see Static Protocol Chosen, page 5-49).
- RIP—Routing Information Protocol (see RIP Protocol Chosen, page 5-51).
- BGP—Border Gateway Protocol (see BGP Protocol Chosen, page 5-54).
- OSPF—Open Shortest Path First (see OSPF Protocol Chosen, page 5-59).
- EIGRP—Enhanced Interior Gateway Routing Protocol (see EIGRP Protocol Chosen, page 5-67).
- None—Specifies parameters for cable services (see None Chosen: Cable Services, page 5-71).

To specify a routing protocol for the PE-CE link, perform the following steps:

Step 1 Choose the appropriate protocol from the Routing Protocol drop-down list.

Note In the case of IPv6 addressing, only a subset of routing protocols are supported. For IOS XR devices, only Static, BGP, EIGRP and None are supported. For IOS devices, only Static, BGP, and None are supported.

When you choose a particular routing protocol, the related parameters for that protocol are displayed.

Step 2 Enter the required information for the selected routing protocol, then click **Next**.

Step 3 Define the MPLS Policy VRF and VPN Selection parameters as described in Defining VRF and VPN Information, page 5-72.

Redistribution of IP Routes

Route redistribution is the process of taking routing information from one source and importing that information into another source. Redistribution should be approached with caution. When you perform route redistribution, you lose information. Metrics must be arbitrarily reset. For example, if a group of RIP routes with a metric of five hops is redistributed into iGRP, there is no way to translate the five hop RIP metric into the composite metric of IGRP. You must arbitrarily choose a metric for the RIP routes as they are redistributed into IGRP. Also, when redistribution is performed at two or more points between two dynamic routing protocol domains, routing loops can occur.

CSC Support

To define a Service Policy with Carrier Supporting Carrier (CSC), choose the CSC Support check box from the MPLS Policy Editor - Routing Information. When CSC Support is checked, the CSC functionality is enabled to the MPLS VPN service. Provisioning CSC is explained in Provisioning Carrier Supporting Carrier, page 5-139

Giving Only Default Routes to CE

When you enable the **Give only default routes to CE** option, you indicate whether the site needs *full routing* or *default routing*. Full routing is when the site must know specifically which other routes are present in the VPN. Default routing is when it is sufficient to send all packets that are not specifically for your site to the VPN.

If you choose this option, Prime Fulfillment configures **the default-info originate** command on the PE router under the running protocol (for RIP, OSPF, or EIGRP). For Static, Prime Fulfillment configures an **ip route 0.0.0.0 0.0.0.0 <out-going interface name>** command on the CE router.

A device can only have one default route. Therefore, the VPN can use a default route, but only on condition that the customer site does not already have a different one. The most common reason to already have a default route is that the site has an Internet feed that is independent of the VPN.

If the CE site already has Internet service, the CE can either route all packets to unknown destinations to the Internet or learn all the routes in the Internet. The obvious choice is to route all packets to unknown destinations to the Internet. If a site has an Internet feed, it might already have a default route. Under such conditions, setting the VPN as the default route is incorrect; the VPN should only route packets meant for other VPN sites.

Static Protocol Chosen

Static routing refers to routes to destinations that are listed manually in the router. Network reachability in this case is not dependent on the existence and state of the network itself. Whether a destination is up or down, the static routes remain in the routing table and traffic is still sent to that destination.

When you choose **Static** as the protocol, four options are enabled: **CSC Support**, **Give Only Default Routes to CE**, **Redistribute Connected (BGP only)**, and **Default Information Originate (BGP only)**.

Note Two other options (**AdvertisedRoutes** and **Default Routes - Routes to reach other sites**) are available when you create the service request. See Setting Static Routing Protocol Attributes (for IPv4 and IPv6), page 5-90.

To specify Static as the routing protocol for the service policy, perform the following steps:

Step 1 **CsC Support:** To define a Service Policy with Carrier Supporting Carrier (CSC), choose the CSC Support check box from the MPLS Policy Editor - Routing Information.

When CSC Support is checked, the CSC functionality is enabled to the MPLS VPN service. Provisioning CSC is explained in Provisioning Carrier Supporting Carrier, page 5-139

This attribute is not available if the IP addressing scheme was set to IPv6 in previous steps.

Step 2 **Give Only Default Routes to CE:** Specify whether this service policy should give only default routes to the CE when provisioning with static routes.

When you enable the **Give only default routes to CE** option with static route provisioning on the PE-CE link, Prime Fulfillment creates a default route on the CE that points to the PE. The VRF static route to the CE site is redistributed into BGP to other sites in the VPN.

When you choose this option, the default route (0.0.0.0/32) is automatically configured; the site contains no Internet feed or any other requirement for a default route. When the site encounters a packet that does not route locally, it can send the packet to the VPN.

If you choose this option, Prime Fulfillment configures **the default-info originate** command on the PE router under the running protocol (for RIP, OSPF, or EIGRP). For Static, Prime Fulfillment configures an **ip route 0.0.0.0 0.0.0.0 <out-going interface name>** command on the CE router.

Step 3 **Redistribute Connected (BGP Only):** Indicate whether this service policy should redistribute the connected routes to the other CEs in the VPN.

When you enable the **Redistribute Connected** option, the connected routes (that is, the routes to the directly connected PEs or CEs) are distributed to all the other CEs in that particular VPN. This option is meant for iBGP if the routing protocol between PE-CE is a non-BGP protocol. For example, if the routing protocol is RIP, OSPF, EIGRP, or Static, the option is meant for the router BGP that is configured on the PE for the MPLS core. On the PE router, there is one router BGP process running at all times for MPLS. This option is also for BGP.

Tip You must enable the **Redistribute Connected** option when joining the management VPN and you are also using IP numbered addresses.

Step 4 **Default Information Originate (BGP Only):** When you enable this option, Prime Fulfillment issues a **default-information-originate** command under the iBGP address family for the currently specified VRF.

The **Default Information Originate** option is required, especially in the hub and spoke topology because each spoke must be able to communicate with every other spoke (by injecting a default route in the hub PE to the spoke PEs).

Step 5 When finished defining static routing for this service policy, click **Next**.

The MPLS Policy VRF and VPN Membership dialog box appears. To proceed, see Defining VRF and VPN Information, page 5-72.

RIP Protocol Chosen

The Routing Information Protocol (RIP) is a distance-vector protocol that uses hop count as its metric. RIP is an Interior Gateway Protocol (IGP), which means that it performs routing within a single autonomous system. RIP sends routing-update messages at regular intervals and when the network topology changes. When a router receives a routing update that includes changes to an entry, it updates its routing table to reflect the new route. The metric value for the path is increased by one, and the sender is specified as the next hop.

RIP routers maintain only the best route to a destination—that is, the route with the lowest possible metric value. After updating its routing table, the router immediately begins transmitting routing updates to inform other network routers of the change. These updates are sent independently of the regularly scheduled updates that RIP routers transmit.

To specify RIP as the routing protocol for the service policy, perform the following steps:

Step 1 Choose **RIP** from the Routing Protocol drop-down list.

The RIP Routing Protocol window appears.

Step 2 **CSC Support:** To define a Service Policy with Carrier Supporting Carrier (CSC), choose the CSC Support check box from the MPLS Policy Editor - Routing Information.

When CSC Support is checked, the CSC functionality is enabled to the MPLS VPN service. Provisioning CSC is explained in Provisioning Carrier Supporting Carrier, page 5-139

Step 3 **Give Only Default Routes to CE:** Specify whether you want to give only the default routes to the CE.

When an internetwork is designed hierarchically, default routes are a useful tool to limit the need to propagate routing information. Access-level networks, such as branch offices, typically have only one connection to headquarters. Instead of advertising all of an organization's network prefixes to a branch office, configure a default route. If a destination prefix is not in a branch office's routing table, forward the packet over the default route. The Cisco IP routing table displays the default route at the top of the routing table as the "Gateway of Last Resort." RIP automatically redistributes the 0.0.0.0 0.0.0.0 route.

If you choose this option, Prime Fulfillment configures **the default-info originate** command on the PE router under the running protocol (for RIP, OSPF, or EIGRP). For Static, Prime Fulfillment configures an **ip route 0.0.0.0 0.0.0.0 <out-going interface name>** command on the CE router.

When you enable the **Give Only Default Routes to CE** option for RIP, Prime Fulfillment creates a default RIP route on the PE; the default RIP route points to the PE and is sent to the CE. The provisioning request gives you the option of redistributing any other routing protocols in the customer network into the CE RIP routing protocol. The RIP routes on the PE to the CE site are redistributed into BGP to other VPN sites.

When you choose this option for RIP routing, the PE instructs the CE to send any traffic it cannot route any other way to the PE. Do *not* use this option if the CE site needs a default route for any reason, such as having a separate Internet feed.

Step 4 **Redistribute Static:** (BGP and RIP) Specify whether you want to redistribute static routes into the core BGP network.

When you enable the **Redistribute Static** option for RIP, the software imports the static routes into the core network (running BGP) and to the CE (running RIP).

Step 5 **Redistribute Connected:** (BGP only) Specify whether you want to redistribute the connected routes to the CEs in the VPN.

When you enable the **Redistribute Connected** option for BGP, the software imports the connected routes (that is, the routes to the directly connected PEs or CEs) to all the other CEs in that particular VPN.

When you enable the Redistribute Connected option, the connected routes (that is, the routes to the directly connected PEs or CEs) are distributed to all the other CEs in that particular VPN. This option is meant for iBGP if the routing protocol between PE-CE is a non-BGP protocol. For example, if the routing protocol is RIP, OSPF, EIGRP, or Static, the option is meant for the router BGP that is configured on the PE for the MPLS core. On the PE router, there is one router BGP process running at all times for MPLS. This option is also for BGP.

Step 6 **RIP Metrics:** (BGP only) Enter the appropriate RIP metric value. The valid metric values are **1** through **16**.

The metrics used by RIP are hop counts. The hop count for all directly connected interfaces is **1**. If an adjacent router advertises a route to another network with a hop count of 1, then the metric for that network is 2, since the source router must send a packet to that router to get to the destination network.

As each router sends its routing tables to its neighbors, a route can be determined to each network within the AS. If there are multiple paths within the AS from a router to a network, the router selects the path with the smallest hop count and ignores the other paths.

Step 7 **Redistributed Protocols on PE:** Specify whether you want to redistribute the routing protocols into the PE.

Redistribution allows routing information discovered through another routing protocol to be distributed in the update messages of the current routing protocol. With redistribution, you can reach all the points of your IP internetwork. When a RIP router receives routing information from another protocol, it updates all of its RIP neighbors with the new routing information already discovered by the protocol it imports redistribution information from.

To specify the protocols that RIP needs to import routing information to the PE:

a. From the **Redistribute Protocols on PE** option, click **Edit**.

The PE Redistributed Protocol dialog box appears.

b. Click **Add**.

The PE Redistributed Protocols dialog box appears.

c. From the Protocol Type drop-down list, choose the protocol you want to import into the PE.

You can choose one of the following: **Static**, **OSPF**, or **EIGRP**.

- Redistribute Static. When you choose **Static** routes for redistribution into RIP, Prime Fulfillment imports the static routes into the PE that is running RIP.

  There are no parameters or metrics required for redistributing Static routes into the PE.

- Redistribute OSPF (Open Shortest Path First). When you choose the **OSPF** protocol for redistribution into RIP, Prime Fulfillment imports the OSPF routes into the PE that is running RIP.

  Parameter: OSPF process number

  Metric: Any numeral from 1 to 16

- Redistribute EIGRP (Enhanced IGRP). When you choose the **EIGRP** protocol for redistribution into RIP, Prime Fulfillment imports the EIGRP routes into the PE that is running RIP.

  Parameter: EIGRP autonomous system (AS) number

  Metric: Any numeral from 1 to 16

d. Choose the protocol you want to redistribute into RIP on the PE.

e. Enter the appropriate parameter for the protocol selected.

f. Click **Add**.

g. Repeat these steps for any additional protocols you want to redistribute into RIP on the PE, then click **OK**.

Step 8 **Redistribute Protocols on CE:** Specify whether you want to redistribute the routing protocols into the CE.

To specify the protocols that RIP needs to import routing information to the CE:

a. From the **Redistribute Protocols on CE** option, click **Edit**.

The CE Redistributed Protocol dialog box appears.

b. Click **Add**.

The CE Redistributed Protocols dialog box appears.

c. From the Protocol Type drop-down list, choose the protocol you want to import into the CE.

You can choose one of the following protocols: **Static**, **BGP**, **Connected (routes)**, **IGRP**, **OSPF**, **EIGRP**, or **IS-IS**.

- Redistribute Static. When you choose **Static** routes for redistribution into RIP, Prime Fulfillment imports the static routes into the CE that is running RIP.

  There are no parameters required for redistributing Static routes into the CE.

- Redistribute BGP (Border Gateway Protocol). When you choose the **BGP** protocol for redistribution into RIP, Prime Fulfillment imports the BGP routes into the CE that is running RIP.

  Parameter: BGP autonomous system (AS) number

- Redistribute Connected routes. When you choose the **Connected** routes for redistribution into RIP, Prime Fulfillment imports all the routes to the interfaces connected to the current router. Use the **Connected** option when you want to advertise a network, but you don't want to send routing updates into that network. Note that redistributing connected routes indiscriminately redistributes all connected routes into the routing domain.

  Parameter: No parameter required

- Redistribute IGRP (Interior Gateway Routing Protocol). When you choose the **IGRP** (Interior Gateway Routing) protocol for redistribution into RIP, Prime Fulfillment imports the IGRP routes into the CE that is running RIP.

  Parameter: IGRP autonomous system (AS) number

- Redistribute EIGRP (Enhanced IGRP). When you choose the **EIGRP** protocol for redistribution into RIP, Prime Fulfillment imports the EIGRP routes into the PE that is running RIP.

  Parameter: EIGRP autonomous system (AS) number

- Redistribute OSPF (Open Shortest Path First). When you choose the **OSPF** protocol for redistribution into RIP, Prime Fulfillment imports the OSPF routes into the CE that is running RIP.

  Parameter: OSPF process number

- Redistribute IS-IS (Intermediate System-to-Intermediate System. When you choose the **IS-IS** protocol for redistribution into RIP, Prime Fulfillment imports the IS-IS routes into the CE that is running RIP.

  Parameter: IS-IS tag number

d. Choose the protocol you want to redistribute into RIP on the CE.

e. Enter the appropriate parameter for the selected protocol.

f. Click **Add**.

g. Repeat these steps for any additional protocols you want to redistribute into RIP on the CE, then click **OK**.

Step 9 When you are satisfied with the RIP protocol settings for this service policy, click **Next**.

The MPLS Policy VRF and VPN Membership dialog box appears. To proceed, see Defining VRF and VPN Information, page 5-72.

Note If a PE link is initially configured to use the RIP routing protocol and subsequently modified to use another routing protocol (or static routing), Prime Fulfillment does not remove all of the RIP CLI commands associated with the interface from the PE configuration file. Specifically, Prime Fulfillment does not remove the address family subcommands under the RIP command unless the VRF associated with the service request is removed. This is because Prime Fulfillment configures the RIP protocol using a network class (that is, network a.0.0.0) based under address-family. Later, if the routing protocol is changed, Prime Fulfillment does not remove any other services under the same network.

BGP Protocol Chosen

BGP (Border Gateway Protocol) operates over TCP (Transmission Control Protocol), using port 179. By using TCP, BGP is assured of reliable transport, so the BGP protocol itself lacks any form of error detection or correction (TCP performs these functions). BGP can operate between peers that are separated by several intermediate hops, even when the peers are not necessarily running the BGP protocol.

BGP operates in one of two modes: Internal BGP (iBGP) or External BGP (eBGP). The protocol uses the same packet formats and data structures in either case. iBGP is used between BGP speakers within a single autonomous system, while eBGP operates over inter-AS links.

eBGP extensions are supported for IPv6 and dual stacked services. The eBGP extensions are configured per BGP neighbor. Thus, the IPv4 and IPv6 neighbors for the same VRF can be configured with a different set of values. Prime Fulfillment facilitates this by allowing these parameters to be configured per BGP neighbor.

To specify BGP as the routing protocol for the service policy, perform the following steps:

Step 1 Choose **BGP** from the Routing Protocol drop-down list.

The BGP Routing Protocol window appears.

Step 2 **CsC Support:** To define a Service Policy with Carrier Supporting Carrier (CSC), check the CSC Support check box from the MPLS Policy Editor - Routing Information.

When CSC Support is checked, the CSC functionality is enabled to the MPLS VPN service. Provisioning CSC is explained in Provisioning Carrier Supporting Carrier, page 5-139

This attribute is not available if the IP addressing scheme was set to IPv6 in previous steps.

Step 3 **Redistribute Static (BGP Only):** Indicate whether you want to redistribute static routes into BGP.

If you are importing static routes into BGP, choose this check box.

Step 4 **Redistribute Connected Routes (BGP Only):** Indicate whether you want to redistribute the directly connected routes into BGP.

Enabling the **Redistribute Connected** option imports all the routes to the interfaces connected to the current router. Use the **Redistribute Connected** option when you want to advertise a network, but you don't want to send routing updates into that network. Note that redistributing connected routes indiscriminately redistributes all connected routes into the routing domain.

When you enable the **Redistribute Connected** option, the connected routes (that is, the routes to the directly connected PEs or CEs) are distributed to all the other CEs in that particular VPN. This option is meant for iBGP if the routing protocol between PE-CE is a non-BGP protocol. For example, if the routing protocol is RIP, OSPF, EIGRP, or Static, the option is meant for the router BGP that is configured on the PE for the MPLS core. On the PE router, there is one router BGP process running at all times for MPLS. This option is also for BGP.

Step 5 **Default Information Originate:** Choose an appropriate option from the drop-down list to cause the BGP speaker (local router) to send a default route to a neighbor.

This inserts the default-originate command under the per-neighbor configuration.

The drop-down list has three choices:

- **None.** This is the default choice. The default-origination command is not added to the per-neighbor configuration. The default route is not advertised to BGP neighbors.
- **Enable.** Allows you to specify the name of a route policy in the Route-Policy (Default Information Origination) field, which dynamically appears in the Prime Fulfillment GUI. The route policy allows route 0.0.0.0 to be injected conditionally. See the usage notes below for further details.
- **Disable.** Prevents the default-originate command characteristics from being inherited from a parent group.

Usage notes:

- Entering a route policy in the Route-Policy (Default Information Origination) field is optional.
- Any route policy that is specified must be pre-existing on the device. If not, Prime Fulfillment will generate an error message when a service request based on the policy is created.
- The default-originate command does not require the presence of the default route (0.0.0.0/0 for IPv4 or ::/0 for IPv6) in the local router. When the default-originate command is used with a route policy, the default route is advertised if any route in the BGP table matches the policy.
- The Default Information Originate attribute is supported in MPLS policies and service requests for both IPv4 and IPv6 address families. It is only supported for MPLS PE_CE and PE_No_CE policies and service requests. It is not supported in MVRFCE policies and service requests.
- The Default Information Originate attribute is only supported on IOS XR devices.
- The following Prime Fulfillment template variables support this feature:
  - For IPv4: PE_CE_NBR_DEFAULT_INFO_ORIGINATE_ROUTE_POLICY
  - For IPv4: PE_CE_NBR_DEFAULT_INFO_ORIGINATE
  - For IPv6: PE_CE_NBR_DEFAULT_INFO_ORIGINATE_ROUTE_POLICY_IPV6
  - For IPv6: PE_CE_NBR_DEFAULT_INFO_ORIGINATE_IPV6
- For sample configlets showing the use of the Default Information Originate option, see PE L3 MPLS VPN (BGP, Default Information Originate, IOS XR), page 5-212.

Step 6 **CE BGP AS ID:** Enter the BGP autonomous system (AS) number for the customer's BGP network.

The autonomous number assigned here to the CE must be different from the BGP AS number for the service provider's core network.

2-byte integer values are supported as valid AS number values. In addition, Prime Fulfillment supports a remote 4-byte AS number in the format [0-65535].[0-65535]. As an example: 100.65535. This remote 4-byte AS number is supported as a CE BGP AS number in a service policy and in a service request. If the platform does not support a remote 4-byte AS number, the service deployment fails. The remote 4-byte AS number is not supported on IOS platforms, but is supported on IOS XR (for both IPv4 and IPv6 services).

Step 7 **Neighbor Allow-AS In:** If appropriate, enter the **Neighbor Allow-AS-in** value.

When you enter a **Neighbor Allow-AS-in** value, you specify a maximum number of times (up to 10) that the service provider autonomous system (AS) number can occur in the autonomous system path.

Step 8 **Neighbor AS Override:** If required for this VPN, enable the **Neighbor AS Override** option.

The AS Override feature allows the MPLS VPN service provider to run the BGP routing protocol with a customer even if the customer is using the same AS number at different sites. This feature can be used if the VPN customer uses either a private or public autonomous system number.

When you enable the **Neighbor AS-Override** option, you configure VPN Solutions Center to reuse the same AS number on all the VPN's sites.

Step 9 **Route Map/Policy In:** Enter a route map (IOS devices) or route policy (IOS XR devices) to apply to inbound routes.

See the usage notes following Step 10 for more information on this attribute.

Note This attribute is not supported for use with MVRFCE policies and service requests.

Step 10 **Route Map/Policy Out:** Enter a route map (IOS devices) or route policy (IOS XR devices) to apply to outbound routes.

Note This attribute is not supported for use with MVRFCE policies and service requests. It is also not supported for IPv6 on IOS devices in service requests.

Usage notes for IOS devices (BGP route map):

- The Route Map/Policy In and Route Map/Policy Out attributes are available to support **route-map** commands for IOS devices with BGP as the PE-CE protocol. They are used to apply a route map to inbound or outbound routes for the purpose of route filtering.
- The value entered in the text field translates to the **neighbor route-map** command in address family or router configuration mode, as shown in the following example configuration:

```
neighbor x.x.x.x route-map slmpls-in in
neighbor x.x.x.x route-map no-routes out
```

- These attributes are optional. For IOS devices, no default value is required.
- The following Prime Fulfillment template variables support BGP route map for IOS devices:
  - PE_CE_NBR_ROUTE_MAP_IN_NAME
  - PE_CE_NBR_ROUTE_MAP_OUT_NAME
- At the service request level, the Route Map/Policy In attribute is disabled and cleared if Site of Origin is enabled. The Site of Origin attribute does not show up at the policy level, but only in the service request workflow (and only in the case of an IOS device and a configuration consisting of a PE with no CE). For additional information on this behavior, see the usage notes for the Site of Origin attribute on page 5-96.

Usage notes for IOS XR devices (route policy):

- The Route Map/Policy In and Route Map/Policy Out attributes are available to support **route-policy** commands for IOS XR devices. They provide a way to apply a routing policy to updates advertised to or received from a Border Gateway Protocol (BGP) neighbor. The policy filters routes or modifies route attributes.You specify the name of a routing policy for an inbound or outbound route.
- There are globally defined route policies that can be referred to (for example, “pass all”), but the Route Map/Policy In and Route Map/Policy Out attributes provide a means for you to override these with your own specific route policies.
- The actual route policy must be configured externally on the device, prior to creating a service request based on the policy.
- The in/out values from the GUI are inserted into the IOS XR device configuration, as follows:

```
route-policy <IN param> in
route-policy <OUT param> out
```

- These attributes are optional. For IOS XR devices, if no values are supplied, they default to the DEFAULT value.
- The following Prime Fulfillment template variables support Prime Fulfillment route policy commands for IOS XR devices:
  - PE_CE_BGP_Neighbor _Route_Map_Or_Policy_In
  - PE_CE_BGP_Neighbor _ Route_Map _Or_Policy_Out

Step 11 **Neighbor Send Community:** Choose one of the following from the drop-down list to send a communities attribute to a BGP neighbor:

- **None.** Do not send a community attribute to a BGP neighbor.
- **Standard.** Send only standard communities to a BGP neighbor.
- **Extended.** Send only extended communities to a BGP neighbor.
- **Both.** Send both standard and extended communities to a BGP neighbor.

This option is only available when the PE-CE routing protocol is BGP. It is applicable for both IOS and IOS XR devices. It is available for both IPv4 and IPv6 external BGP (eBGP) neighbors.

Note This attribute is not supported for use with MVRFCE policies and service requests.

Step 12 Specify whether you want to redistribute routing protocols into the CE.

Redistributed Protocols on CE: The redistribution of routes into MP-iBGP is necessary only when the routes are learned through any means other than BGP between the PE and CE routers. This includes connected subnets and static routes. In the case of routes learned via BGP from the CE, redistribution is not required because it’s performed automatically.

To specify the protocols that BGP needs to import routing information to the CE:

a. From the **Redistribute Protocols on CE** option, click **Edit**.

   The CE Redistributed Protocol dialog box appears.

b. Click **Add**.

   The CE Redistributed Protocols dialog box appears.

c. From the Protocol Type drop-down list, choose the protocol you want to import into the CE.

You can choose one of the following protocols: **Static**, **RIP**, **Connected (routes)**, **IGRP**, **OSPF**, **EIGRP**, or **IS-IS**.

- Redistribute Static. When you choose **Static** routes for redistribution into BGP, Prime Fulfillment imports the static routes into the CE that is running BGP.

  Parameter: No parameter required

- Redistribute RIP (Routing Information Protocol). When you choose the **RIP** protocol for redistribution into BGP, Cisco Prime Fulfillment imports the RIP routes into the CE that is running BGP.

  Parameter: No parameter required

- Redistribute Connected routes. When you choose the **Connected** routes for redistribution into BGP, Prime Fulfillment imports all the routes to the interfaces connected to the current router. Use the **Connected** option when you want to advertise a network, but you do not want to send routing updates into that network. Note that redistributing connected routes indiscriminately redistributes all connected routes into the routing domain.

  Parameter: No parameter required

- Redistribute IGRP (Interior Gateway Routing Protocol). When you choose the **IGRP** protocol for redistribution into BGP, IP Solution Center imports the IGRP routes into the CE that is running BGP.

  Parameter: IGRP autonomous system (AS) number

- Redistribute EIGRP (Enhanced IGRP). When you choose the **EIGRP** protocol for redistribution into BGP, Prime Fulfillment imports the EIGRP routes into the CE that is running BGP.

  Parameter: EIGRP autonomous system (AS) number

- Redistribute OSPF (Open Shortest Path First). When you choose the **OSPF** protocol for redistribution into BGP, Prime FulfillmentPrime Fulfillment imports the OSPF routes into the CE that is running BGP.

  Parameter: OSPF process number

- Redistribute IS-IS (Intermediate System-to-Intermediate System). When you choose the **IS-IS** protocol for redistribution into BGP, Prime Fulfillment imports the IS-IS routes into the CE that is running BGP.

  Parameter: IS-IS tag number

d. Choose the protocol you want to redistribute into BGP on the CE.

e. Enter the appropriate parameter for the selected protocol.

f. Click **Add**.

g. Repeat these steps for any additional protocols you want to redistribute into BGP on the PE, then click **OK**.

Step 13 **Advertise Interval:** Enter the eBGP advertisement interval.

The value is an integer ranging from 0 to 600, specifying the number of seconds of the advertisement interval. The default setting is 30 seconds for the eBGP peer, if it is not explicitly configured. This eBGP extension is available to configure for both IOS and IOS XR PE devices.

Step 14 **Max Prefix Number:** Enter the maximum number of prefixes that can be received from a neighbor.

Usage notes:

- This feature allows a router to bring down a peer when the number of received prefixes from that peer exceeds the limit.

- The range is:
  - 1–2147483647 for IOS devices
  - 1–4294967295 for IOS XR devices
- This and the related options are supported for both IPv4 and IPv6 address families.
- For sample configlets showing the use of the Max Prefix Number, Max Prefix Threshold, Max Prefix Warning Only, and Max Prefix Restart options, see PE L3 MPLS VPN (BGP, Maximum Prefix/Restart, IOS XR), page 5-207.

Step 15 **Max Prefix Threshold:** Enter a value that specifies at what percentage Max Prefix Number is configured.

The range is from 1 to 100 percent, with the default being 75 percent. When this threshold is reached, the router generates a warning message. For example, if the Max Prefix Number is 20 and the Max Prefix Threshold is 60, the router generates warning messages when the number of BGP learned routes from the neighbor exceeds 60 percent of 20, or 12 routes.

Step 16 **Max Prefix Warning Only:** Check this check box if you want to allow the router to generate a log message when the maximum prefix limit is exceeded, instead of terminating the peering session.

Step 17 **Max Prefix Restart:** Enter a value, in minutes, specifying when the router will automatically re-establish a peering session that has been brought down because the configured maximum prefix limit has been exceeded.

The range is from 1 to 65535. No intervention from the network operator is required when this feature is enabled. This feature attempts to re-establish a disabled peering session at the configured time interval that is specified. However, the configuration of the restart timer alone cannot change or correct a peer that is sending an excessive number of prefixes. The network operator will need to reconfigure the maximum prefix limit or reduce the number of prefixes that are sent from the peer. A peer that is configured to send too many prefixes can cause instability in the network, where an excessive number of prefixes are rapidly advertised and withdrawn. In this case, the Max Prefix Warning Only attribute can be configured to disable the restart capability, while the network operator corrects the underlying problem.

Step 18 When you are satisfied with the BGP protocol settings for this service policy, click **Next**.

The MPLS Policy VRF and VPN Membership dialog box appears. To proceed, see Defining VRF and VPN Information, page 5-72.

OSPF Protocol Chosen

The MPLS VPN backbone is not a genuine OSPF area 0 backbone. No adjacencies are formed between PE routers—only between PEs and CEs. MP-iBGP is used between PEs, and all OSPF routes are translated into VPN IPv4 routes. Thus, redistributing routes into BGP does not cause these routes to become external OSPF routes when advertised to other member sites of the same VPN.

To specify OSPF as the routing protocol for the service policy, perform the following steps:

Step 1 Choose **OSPF** from the Routing Protocol drop-down list.

The OSPF Routing Protocol window appears.

Step 2 **CSC Support:** To define a Service Policy with Carrier Supporting Carrier (CSC), choose the CSC Support check box from the MPLS Policy Editor - Routing Information.

When CSC Support is checked, the CSC functionality is enabled to the MPLS VPN service. Provisioning CSC is explained in Provisioning Carrier Supporting Carrier, page 5-139

Step 3 **Give Only Default Routes to CE:** Specify whether you want to give only the default routes to the CE.

When you enable the **Give only default routes to CE** option, you indicate whether the site needs *full routing* or *default routing*. Full routing is when the site must know specifically which other routes are present in the VPN. Default routing is when it is sufficient to send all packets that are not specifically for your site to the VPN.

If you choose this option, Prime Fulfillment configures the **default-info originate** command on the PE router under the running protocol RIP or EIGRP and the **default-info originate always** command on the PE router under the running protocol OSPF for Static and configures an **ip route 0.0.0.0 0.0.0.0 <out-going interface name>** command on the CE router.

Step 4 **Redistribute Static (BGP only):** Indicate whether you want to redistribute static routes into OSPF.

If you are importing static routes into OSPF, check this check box.

Step 5 **Redistribute Connected Routes (BGP only):** Indicate whether you want to redistribute the directly connected routes into OSPF.

Enabling the **Redistribute Connected** option imports all the routes to the interfaces connected to the current router. Use the **Redistribute Connected** option when you want to advertise a network, but you don't want to send routing updates into that network. Note that redistributing connected routes indiscriminately redistributes all connected routes into the routing domain.

This option is meant for iBGP if the routing protocol between PE-CE is a non-BGP protocol. For example, if the routing protocol is RIP, OSPF, EIGRP, or Static, the option is meant for the router bgp that is configured on the PE for the MPLS core. On the PE router, there is one router bgp process running at all times for MPLS. This option is also for BGP.

Step 6 **Default Information Originate:** Indicate if you want to generate a default external route into an OSPF routing domain.

By checking the Default Information Originate check box, other options dynamically appear in the GUI.

a. Check **OSPF Default Information Originate Always** to advertise the default route regardless of whether the routing table has a default route.

b. For **Metric Value**, enter an OSPF metric to be used for generating the default route. Range is 1–16777214.

c. For **Metric Type**, choose one of the following from the drop-down list to specify the link type associated with the default route:

   - **None**
   - **Type-1 External Route**
   - **Type-2 External Route**

d. For **Default Info Route Policy**, enter the name of a route policy.

Usage notes:

- Default Information Originate is available in MPLS policy and service request workflows.
- All suboptions are optional.
- The route policy, if specified, must be pre-existing on the device. If not, an error is generated when a service request is created based on the policy using this feature.
- This feature is only supported for IOS XR devices.
- This feature is only available for IPv4 address family.

- The following Prime Fulfillment template variables support this feature:
  - PE_CE_OSPF_ METRIC_VALUE
  - PE_CE_OSPF_METRIC_TYPE
  - PE_CE_OSPF_ROUTE_POLICY
- For sample configlets showing the use of the Default Information Originate option, see L3 MPLS VPN (OSPF, Default Information Originate, IOS XR), page 5-222.

Step 7 **OSPF Route Policy:** Enter a route policy.

Usage notes:

- This is an optional attribute.
- This attribute is only supported with IPv4 routing on IOS and IOS XR PE devices.
- This attribute is used to support redistribution of an OSPF route policy. It provides a means to take values from the GUI and insert them into a device configuration, as shown in the examples below.
- Example IOS XR configuration following deployment of a service request based on a policy using this attribute:

```
vrf edn
rd 11.31.128.80:300
address-family ipv4 unicast
redistribute connected
redistribute ospf 3000 route-policy 'xxxx'
```

- Example IOS configuration:

```
address-family ipv4 vrf edn
redistribute connected
redistribute ospf 3000 route-map <route-map>
```

- Characters are taken from the GUI as is. No validation is performed.
- If no valid route policy is supplied, the default route policy is used.
- The actual route policy must be configured externally on the device prior to creating a service request based on this policy.
- The following Prime Fulfillment template variables support the redistribution of the OSPF route policy:
  - PE_CE_Ospf_Route_Policy
  - PE_MVRFCE_Ospf_ Route_Policy

Step 8 **OSPF Redistribute Match Internal/External (BGP only):** To set the match criteria by which OSPF routes are redistributed into other routing domains, choose one of the following from the drop-down list:

- None—Do not specify match criteria for route redistribution. This is the default.
- Internal only—Match routes that are internal to the autonomous system (AS).
- External only—Match routes that are external to the AS.
- Both—Match routes that are internal and external to the AS.

Usage notes:

- This attribute is only supported with IPv4 routing on IOS and IOS XR PE devices.
- Example IOS XR configuration for redistribute OSPF match internal:

```
vrf edn
rd 11.31.128.80:300
```

```
address-family ipv4 unicast
redistribute connected
redistribute ospf 3000 match internal
```

- Example IOS configuration for redistribute OSPF match internal:

```
address-family ipv4 vrf edn
redistribute connected
redistribute ospf 3000 match internal
```

- Example IOS XR configuration for redistribute OSPF match external:

```
vrf edn
rd 11.31.128.80:300
address-family ipv4 unicast
redistribute connected
redistribute ospf 3000 match external
```

- Example IOS configuration for redistribute OSPF match external:

```
address-family ipv4 vrf edn
redistribute connected
redistribute ospf 3000 match external 1 external 2
```

- Example IOS XR configuration when Both option is chosen:

```
redistribute ospf 3000 match internal external
```

- Example IOS configuration when Both option is chosen:

```
redistribute ospf 3000 match internal external 1 external 2
```

- There is no support for **external type 1** or **external type 2** in the IOS XR variation of this command, but the support exists in IOS. In the Prime Fulfillment GUI, there is no option to specify **external type 1** or **external type 2**. The only option is External only. The generated configlets will differ based on whether the device is IOS or IOS XR.
- The Prime Fulfillment template variable PE_CE_Ospf_Match_Internal_External support this attribute.

Step 9 **OSPF Process ID on PE:** Enter the OSPF process ID for the PE.

The OSPF process ID is a unique value assigned for each OSPF routing process within a single router—this process ID is internal to the PE only. You can enter this number either as any decimal number from 1 to 65535, or a number in dotted decimal notation.

Note For additional information on how the OSPF process ID is handled in Prime Fulfillment, see OSPF Process ID for the IGP (IOS XR Only), page 5-65.

Step 10 **Use VRF or VPN Domain ID:** Check this check box to use an OSPF domain ID from a VRF or VPN.

Usage notes:

- If you do not check this check box, you can enter a value for the OSPF domain ID on the PE in the text field of the OSPF Domain ID on PE attribute (the next attribute in the GUI).
- When you check the Use VPN or VRF Domain ID check box, the fields in the OSPF Domain ID on PE attribute are disabled.
- The OSPF domain ID feature is supported only for PE-CE and PE- NoCE policies. The OSPF Domain ID and OSPF Domain ID on PE attributes only show up in the GUI if the policy type is PE-CE or PE-NoCE.

- The OSPF domain ID feature is not supported for MultiVRF-CE policies.
- OSPF domain ID is supported only on IOS XR devices. In the case of IOS devices, Prime Fulfillment ignores the this attribute if you use a VRF object or VPN with an OSPF domain ID specified.
- The OSPF domain ID attribute uniquely identifies the OSPF domain from which a route is redistributed. This domain ID should be unique per customer. For IOS devices, because IOS allows only one VRF per process, the default behavior is that the OSPF process ID is considered as the OSPF domain ID. IOS XR supports multiple VRFs per process. Therefore, for IOS XR devices, you need to explicitly configure a unique OSPF domain ID for each VRF. You can configure one VRF per OSPF process, but it is not a scalable solution.
- Only OSPF domain ID configuration of type 0005 is supported.
- Note the following points in the case of a service request created based on the policy:
  - OSPF domain ID configuration is optional. When Use VPN or VRF Domain ID is not enabled and no value is supplied in the OSPF Domain ID field, Prime Fulfillment ignores the OSPF domain ID configuration.
  - If Use VPN or VRF Domain ID is enabled, at the time of provisioning Prime Fulfillment gets the OSPF domain ID from the selected VPN object. If an OSPF domain ID is not configured in the VPN object, Prime Fulfillment ignores the OSPF domain ID configuration. No error message is generated.
  - When Use VPN or VRF Domain ID is enabled and multiple VPNs are joined for the link (extranet), Prime Fulfillment ignores the OSPF domain configuration.

Step 11 **OSPF Domain ID on PE:** Enter an OSPF domain ID in decimal format.

Usage notes:

- This field is disabled if the Use VPN or VRF Domain ID check box is checked. See notes in the previous step.
- Enter the value in decimal format. The Hex value: field is a non-editable text field that displays the equivalent hex value. The hex value is what actually gets displayed on the device.
- OSPF domain ID is supported only on IOS XR devices. In the case of IOS devices, Prime Fulfillment ignores the this attribute if you use a VRF object or VPN with an OSPF domain ID specified.

Step 12 **OSPF Process ID on CE:** Enter the OSPF process ID for the CE.

The OSPF process ID is a unique value assigned for each OSPF routing process within a single router—this process ID is internal to the CE only. You can enter this number either as any decimal number from 1 to 65535, or a number in dotted decimal notation.

Note For additional information on how the OSPF process ID is handled in Prime Fulfillment, see OSPF Process ID for the IGP (IOS XR Only), page 5-65.

Step 13 **OSPF Process Area Number:** Enter the OSPF process area number.

You can enter the OSPF area number for the PE either as any decimal number in the range specified, or a number in dotted decimal notation.

Step 14 **Redistributed Protocols on PE:** If necessary, specify the redistributed protocols into the PE.

Note Restricting the amount of redistribution can be important in an OSPF environment. Whenever a route is redistributed into OSPF, it is done so as an external OSPF route. The OSPF protocol floods external routes across the OSPF domain, which increases the protocol's overhead and the CPU load on all the routers participating in the OSPF domain.

To specify the protocols that OSPF needs to import to the PE, follow these steps.

a. From the **Redistribute Protocols on PE** option, click **Edit**.

   The PE Redistributed Protocol dialog box appears.

b. Click **Add**.

   The PE Redistributed Protocols dialog box appears.

c. From the Protocol Type drop-down list, choose the protocol you want to import into the PE.

   You can choose one of the following: **Static**, **EIGRP**, or **RIP**.

   - Redistribute Static. When you choose **Static** routes for redistribution into OSPF, Prime Fulfillment imports the static routes into the PE that is running OSPF.

     There are no parameters or metrics required for redistributing Static routes into the PE.

   - Redistribute EIGRP (Enhanced IGRP). When you choose the **EIGRP** protocol for redistribution into OSPF, Prime Fulfillment imports the EIGRP routes into the PE that is running OSPF.

     Parameter: EIGRP autonomous system (AS) number

     Metric: Any numeral from 1 to 16777214

   - Redistribute RIP. When you choose the **RIP** protocol for redistribution into OSPF, Prime Fulfillment imports the RIP routes into the PE that is running OSPF.

     Parameter: No parameter required.

     Metric: Any numeral from 1 to 16777214.

d. Choose the protocol you want to redistribute into OSPF on the PE.

e. Enter the appropriate parameter for the protocol selected.

f. Click **Add**.

g. Repeat these steps for any additional protocols you want to redistribute into OSPF on the PE, then click **OK**.

Step 15 Specify whether you want to redistribute the routing protocols into the CE.

Redistribute Protocols on CE: To specify the protocols that OSPF needs to import routing information to the CE, follow these steps.

a. From the **Redistribute Protocols on CE** option, click **Edit**.

   The CE Redistributed Protocol dialog box appears.

b. Click **Add**.

   The CE Redistributed Protocols dialog box appears.

c. From the Protocol Type drop-down list, choose the protocol you want to import into the CE.

   You can choose one of the following protocols: **Static**, **RIP**, **BGP**, **Connected (routes)**, **IGRP**, **EIGRP**, or **IS-IS**.

   - Redistribute Static. When you choose **Static** routes for redistribution into OSPF, Prime Fulfillment imports the static routes into the CE that is running OSPF.

There are no parameters required for redistributing Static routes into the CE.

- Redistribute RIP. When you choose the **RIP** protocol for redistribution into OSPF, Prime Fulfillment imports the RIP routes into the CE that is running OSPF.

  Parameter: No parameter required

- Redistribute BGP (Border Gateway Protocol). When you choose the **BGP** protocol for redistribution into OSPF, Prime Fulfillment imports the BGP routes into the CE that is running OSPF.

  Parameter: BGP autonomous system (AS) number

- Redistribute Connected routes. When you choose the **Connected** routes for redistribution into OSPF, Prime Fulfillment imports all the routes to the interfaces connected to the current router. Use the **Connected** option when you want to advertise a network, but you don't want to send routing updates into that network. Note that redistributing connected routes indiscriminately redistributes all connected routes into the routing domain.

  Parameter: No parameter required

- Redistribute IGRP (Interior Gateway Routing Protocol). When you choose the **IGRP** (Interior Gateway Routing) protocol for redistribution into OSPF, IP Solution Center imports the IGRP routes into the CE that is running OSPF.

  Parameter: IGRP autonomous system (AS) number

- Redistribute EIGRP (Enhanced IGRP). When you choose the **EIGRP** protocol for redistribution into OSPF, Prime Fulfillment imports the EIGRP routes into the CE that is running OSPF.

  Parameter: EIGRP autonomous system (AS) number

- Redistribute IS-IS (Intermediate System-to-Intermediate System). When you choose the **IS-IS** protocol for redistribution into OSPF, Prime Fulfillment imports the IS-IS routes into the CE that is running OSPF.

  Parameter: IS-IS tag number

d. Choose the protocol you want to redistribute into OSPF on the CE.

e. Enter the appropriate parameter for the selected protocol.

f. Click **Add**.

g. Repeat these steps for any additional protocols you want to redistribute into OSPF on the CE, then click **OK**.

Step 16 When you are satisfied with the OSPF protocol settings for this service policy, click **Next**.

The MPLS Policy VRF and VPN Membership dialog box appears. To proceed, see Defining VRF and VPN Information, page 5-72.

OSPF Process ID for the IGP (IOS XR Only)

Note The information in this section only applies to IOS XR devices, since IOS XR supports a virtual OSPF process. It is not applicable to IOS devices.

For IOS XR devices, Prime Fulfillment keeps the OSPF process for the Interior Gateway Protocol (IGP) as a separate process. By default, the OSPF for all PE-CE links is another process. For further OSPF processes, the PE-CE VRFs are under that parent.

The user is responsible for determining and tracking the OSPF process ID. Prime Fulfillment checks that the PE-CE process ID is different from the IGP process ID and provides a warning message if the process ID is already in use.

If the user provides an OSPF process ID that is already in use for IGP purposes, Prime Fulfillment generates a warning message during deployment of the service request. An OSPF process is considered to be in use if it references a VRF. If it does so, then it is regarded as a non-IGP process; otherwise, it is regarded as an IGP process.

Prime Fulfillment provides a DCPL property to set the maximum number of OSPF processes. The DCPL property is Provisioning\Service\mpls\ospfProcessLimit. The default for this value is 2.
Prime Fulfillment keeps track of how many OSPF processes have been configured. If the limit is exceeded or reached, a warning message is generated during the deployment of the service request. Aside from the warning message, there are no side effects from exceeding the limit.

Note The DCPL limit represents the total of all OSPF processes (IGP or otherwise). No warning is generated if the OSPF process ID is already present as an VRF-based OSPF process. A warning is generated if there is more than one VRF-based OSPF process (assuming a default value of 2 for ospfProcessLimit).

See the following configuration examples.

Example: Core IGP (90)

```
router ospf 90
nsr
log adjacency changes
router-id 11.31.128.77
bfd minimum-interval 200
bfd multiplier 3
network point-to-point
nsf cisco
auto-cost reference-bandwidth 100000
redistribute rip metric 3 metric-type 1
redistribute isis ntt metric 10 metric-type 1
address-family ipv4 unicast
area 51
mpls traffic-eng
interface Loopback0
!
interface GigabitEthernet0/0/0/0
network broadcast
!
!
area 0.0.0.0
mpls traffic-eng
interface GigabitEthernet0/0/0/1
!
interface GigabitEthernet0/0/0/2
network point-to-point
!
interface GigabitEthernet0/0/0/4
network point-to-point
!
interface TenGigE0/3/0/0
!
!
mpls traffic-eng router-id Loopback0
mpls traffic-eng multicast-intact
```

Example: PE-CE VRFs (3000)

```
router ospf 3000
vrf edn
log adjacency changes detail
router-id 1.1.1.77
domain-tag 77
area 0.0.0.100
bfd minimum-interval 250
bfd fast-detect
bfd multiplier 3
network point-to-point
stub
interface GigabitEthernet0/0/5/7.101
!
!
!
vrf regus
log adjacency changes detail
router-id 2.2.2.1
domain-tag 3177
network point-to-point
address-family ipv4 unicast
area 51
bfd minimum-interval 250
bfd fast-detect
bfd multiplier 3
network point-to-point
interface Loopback9000
```

Note If **route-policy** is used on the router, matching is not applicable.

EIGRP Protocol Chosen

Enhanced IGRP (EIGRP) is a hybrid routing protocol that discovers a network like a distance vector protocol (namely IGRP), but maintains a topological database for rapid reconvergence. EIGRP supports variable length subnet masks and discontinuous subnets. When configured for IP, it automatically redistributes routes with IGRP processes defined in the same autonomous system. By default, EIGRP autosummarizes subnets at the classful network boundaries.

EIGRP performs the same metric accumulation as IGRP. However, if you examine the metric calculation between IGRP and EIGRP, you will see that the EIGRP value is much greater. If you divide the EIGRP metric by 256, you get the same IGRP metric value.

EIGRP allows all routers involved in a topology change to synchronize at the same time. Routers that are not affected by topology changes are not involved in the recomputation. The result is very fast convergence time.

To specify EIGRP as the routing protocol for the service policy, perform the following steps:

Step 1 Choose **EIGRP** from the Routing Protocol drop-down list.

The EIGRP Routing Protocol window appears.

Step 2 **CSC Support:** To define a Service Policy with Carrier Supporting Carrier (CSC), choose the CSC Support check box from the MPLS Policy Editor - Routing Information.

When CSC Support is checked, the CSC functionality is enabled to the MPLS VPN service. Provisioning CSC is explained in Provisioning Carrier Supporting Carrier, page 5-139

This attribute is not available if the IP addressing scheme was set to IPv6 in previous steps.

Step 3 **Redistribute Static:** (BGP only) If appropriate, enable the **Redistribute Static (BGP only)** option.

When you enable the Redistribute Static option for BGP, the software imports the static routes into the core network (running BGP).

Step 4 **Redistribute Connected:** (BGP only) If appropriate, enable the **Redistribute Connected (BGP only)** option.

When you enable the **Redistribute Connected** option, the connected routes (that is, the routes to the directly connected PEs or CEs) are distributed to all the other CEs in that particular VPN. This option is meant for iBGP if the routing protocol between PE-CE is a non-BGP protocol. For example, if the routing protocol is RIP, OSPF, EIGRP, or Static, the option is meant for the router BGP that is configured on the PE for the MPLS core. On the PE router, there is one router PCP process running at all times for MPLS. This option is also for BGP.

Note Redistributing connected routes can be problematic because all the connected routes are redistributed indiscriminately into a specified routing domain. If you do not want all connected routes to be redistributed, use a *distribute-list out* statement to identify the specific connected routes that should be redistributed.

Step 5 **EIGRP Authentication KeyChain Name:** Enter a keychain name to authenticate all EIGRP protocol traffic on one or more interfaces.

Usage notes:

- No space characters and backslash (\) characters are allowed in the keychain name.
- If no name is specified, EIGRP keychain authentication is not deployed.
- This option is supported for both IPv4 and IPv6 address families.
- This option is available only for IOS XR devices.
- For sample configlets showing the use of the EIGRP Authentication KeyChain Name option, see PE L3 MPLS VPN (EIGRP, Authentication Keychain Name, IOS XR), page 5-227.

Step 6 **EIGRP AS ID on PE:** Enter the EIGRP autonomous system ID on the PE.

This is a unique 16-bit number.

Step 7 **EIGRP AS ID on CE:** Enter the EIGRP autonomous system ID on the CE.

This is a unique 16-bit number.

Step 8 Enter the values for the EIGRP metrics as described below.

EIGRP Metrics

EIGRP uses metrics in the same way as IGRP. Each route in the route table has an associated metric. EIGRP uses a composite metric much like IGRP, except that it is modified by a multiplier of 256. Bandwidth, Delay, Load, Reliability, and MTU are the submetrics. Like IGRP, EIGRP chooses a route based primarily on bandwidth and delay, or the composite metric with the lowest numerical value. When EIGRP calculates this metric for a route, it calls it the feasible distance to the route. EIGRP calculates a feasible distance to all routes in the network.

Bandwidth Metric: Bandwidth is expressed in units of Kilobits. It must be statically configured to accurately represent the interfaces that EIGRP is running on. For example, the default bandwidth of a 56-kbps interface and a T1 interface is 1,544 kbps.

Delay Metric: Delay is expressed in microseconds. It, too, must be statically configured to accurately represent the interface that EIGRP is running on. The delay on an interface can be adjusted with the delay **time_in_microseconds** interface subcommand.

Reliability Metric: Reliability is a dynamic number in the range of 1 to 255, where 255 is a 100 percent reliable link and 1 is an unreliable link.

Loading Metric: Load is the number in the range of 1 to 255 that shows the output load of an interface. This value is dynamic and can be viewed using the **show interfaces** command. A value of 1 indicates a minimally loaded link, whereas 255 indicates a link loaded 100 percent.

MTU Metric: The maximum transmission unit (MTU) is the recorded smallest MTU value in the path, usually 1500.

Note Whenever you are influencing routing decisions in IGRP or EIGRP, use the Delay metric over Bandwidth. Changing bandwidth can affect other routing protocols, such as OSPF. Changing delay affects only IGRP and EIGRP.

Step 9 **Redistributed Protocols on PE:** If necessary, specify the redistributed protocols on the PE.

When configured for IP, it automatically redistributes routes with IGRP processes defined in the same autonomous system. By default, EIGRP autosummarizes subnets at the classful network boundaries.

To specify the protocols that EIGRP needs to import to the PE:

a. From the **Redistribute Protocols on PE** option, click **Edit**.

   The PE Redistributed Protocol dialog box appears.

b. Click **Add**.

   The PE Redistributed Protocols dialog box appears.

c. From the Protocol Type drop-down list, choose the protocol you want to import into the PE.

   You can choose one of the following: **Static**, **RIP**, or **OSPF**.

   - Redistribute Static. When you choose **Static** routes for redistribution into EIGRP, Prime Fulfillment imports the static routes into the PE that is running OSPF.

     There are no parameters or metrics required for redistributing Static routes into the PE.

   - Redistribute RIP. When you choose the **RIP** protocol for redistribution into EIGRP, Prime Fulfillment imports the RIP routes into the PE that is running EIGRP.

     Parameter: No parameter required

     Metric: Any numeral from 1 to 16777214

   - Redistribute OSPF (Open Shortest Path First). When you choose the **OSPF** protocol for redistribution into EIGRP, Prime Fulfillment imports the OSPF routes into the PE that is running EIGRP.

     Parameter: OSPF process number

     Metric: Any numeral from 1 to 16

d. Choose the protocol you want to redistribute into EIGRP on the CE.

e. Enter the appropriate parameter for the protocol selected.

f. Click **Add**.

g. Repeat these steps for any additional protocols you want to redistribute into EIGRP on the PE, then click **OK**.

Step 10 **Redistribute Protocols on CE:** Specify whether you want to redistribute the routing protocols into the CE.

To specify the protocols that EIGRP needs to import routing information to the CE:

a. From the **Redistribute Protocols on CE** option, click **Edit**.

The CE Redistributed Protocol dialog box appears.

b. Click **Add**.

The CE Redistributed Protocols dialog box appears.

c. From the Protocol Type drop-down list, choose the protocol you want to import into the CE.

You can choose one of the following protocols: **Static**, **BGP**, **Connected (routes)**, **IGRP**, **RIP**, **OSPF**, or **IS-IS**.

- Redistribute Static. When you choose **Static** routes for redistribution into EIGRP, Prime Fulfillment imports the static routes into the CE that is running OSPF.

  There are no parameters required for redistributing Static routes into the CE.

- Redistribute BGP (Border Gateway Protocol). When you choose the **BGP** protocol for redistribution into EIGRP, Prime Fulfillment imports the BGP routes into the CE that is running OSPF.

  Parameter: BGP autonomous system (AS) number

- Redistribute Connected routes. When you choose the **Connected** routes for redistribution into EIGRP, Prime Fulfillment imports all the routes to the interfaces connected to the current router. Use the **Connected** option when you want to advertise a network, but you don't want to send routing updates into that network. Note that redistributing connected routes indiscriminately redistributes all connected routes into the routing domain.

  When you enable the **Redistribute Connected** option, the connected routes (that is, the routes to the directly connected PEs or CEs) are distributed to all the other CEs in that particular VPN. This option is meant for iBGP if the routing protocol between PE-CE is a non-BGP protocol. For example, if the routing protocol is RIP, OSPF, EIGRP, or Static, the option is meant for the router BGP that is configured on the PE for the MPLS core. On the PE router, there is one router BGP process running at all times for MPLS. This option is also for BGP.

  Parameter: No parameter required

- Redistribute IGRP (Interior Gateway Routing Protocol). When you choose the **IGRP** (Interior Gateway Routing) protocol for redistribution into EIGRP, IP Solution Center imports the IGRP routes into the CE that is running EIGRP.

  Parameter: IGRP autonomous system (AS) number

- Redistribute RIP. When you choose the **RIP** protocol for redistribution into EIGRP, Cisco Prime Fulfillment imports the RIP routes into the CE that is running EIGRP.

  Parameter: No parameter required

- Redistribute OSPF (Open Shortest Path First). When you choose the **OSPF** protocol for redistribution into EIGRP, Prime Fulfillment imports the OSPF routes into the CE that is running EIGRP.

  Parameter: OSPF process number

- Redistribute IS-IS (Intermediate System-to-Intermediate System). When you choose the **IS-IS** protocol for redistribution into EIGRP, Prime Fulfillment imports the IS-IS routes into the CE that is running EIGRP.

  Parameter: IS-IS tag number

d. Choose the protocol you want to redistribute into EIGRP on the CE.

e. Enter the appropriate parameter for the selected protocol.

f. Click **Add**.

g. Repeat these steps for any additional protocols you want to redistribute into EIGRP on the CE, then click **OK**.

Step 11 When you are satisfied with the EIGRP protocol settings for this service policy, click **Next**.

The MPLS Policy VRF and VPN Membership dialog box appears. To proceed, see Defining VRF and VPN Information, page 5-72.

None Chosen: Cable Services

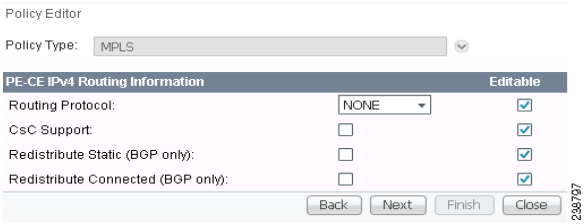
When operating a cable link, the link does not run a routing protocol. The **None** option in the service policy routing protocol dialog box is provided to allow for configuring a service over a cable link without having to unnecessarily specify a routing protocol.

If this service policy is for cable services, perform the following steps:

Step 1 Choose **None** from the list of routing protocols.

The following dialog box appears, as shown in Figure 5-4.

Figure 5-4 No Routing Protocol Selected



Step 2 **CSC Support:** To define a Service Policy with Carrier Supporting Carrier (CSC), choose the CSC Support check box from the MPLS Policy Editor - Routing Information.

When CSC Support is checked, the CSC functionality is enabled to the MPLS VPN service. Provisioning CSC is explained in Provisioning Carrier Supporting Carrier, page 5-139

Step 3 **Redistribute Static:** If you want to distribute static routes into the provider core network (which runs BGP), check the **Redistribute Static (BGP only)** check box.

Step 4 **Redistribute Connected:** Because there is no routing protocol on the cable link, we recommend that you redistribute the connected routes to all the other CEs in the VPN. To do so, check the **Redistribute Connected (BGP only)** check box.

When you enable the **Redistribute Connected** option, the connected routes (that is, the routes to the directly connected PEs or CEs) are distributed to all the other CEs in that particular VPN. This option is meant for iBGP if the routing protocol between PE-CE is a non-BGP protocol. For example, if the routing protocol is RIP, OSPF, EIGRP, or Static, the option is meant for the router BGP that is configured on the PE for the MPLS core. On the PE router, there is one router BGP process running at all times for MPLS. This option is also for BGP.

Step 5 When finished specifying the necessary settings, click **Next**.

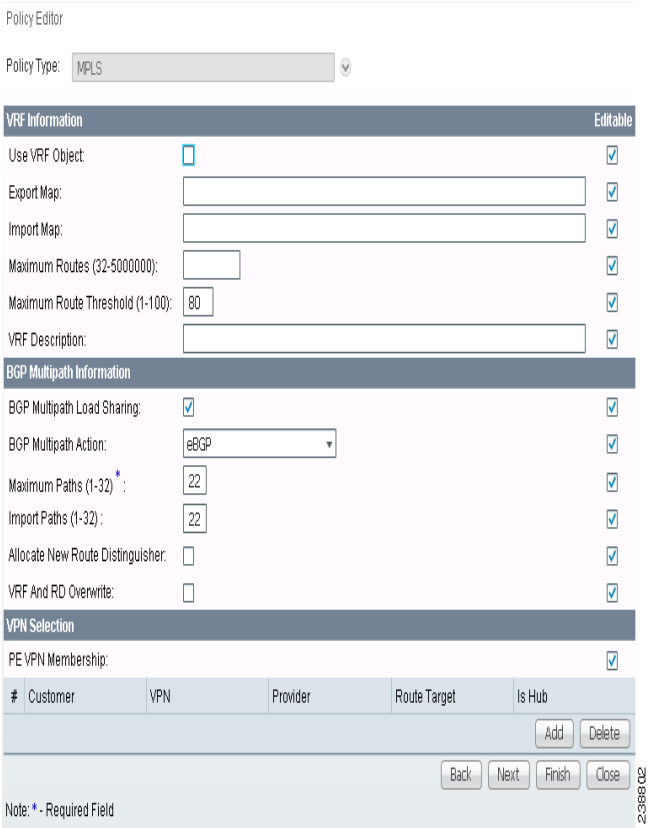
The MPLS Policy VRF and VPN Membership dialog box appears. To proceed, see Defining VRF and VPN Information, page 5-72.

Defining VRF and VPN Information

When you are finished defining the routing protocol(s) for the service policy, you must then specify the VRF and VPN information for this service policy. To do this, perform the following steps:

Step 1 The MPLS Policy VRF and VPN Membership dialog box appears, as shown in Figure 5-5.

Figure 5-5 Specifying the VRF Information



Step 2 If you want to set the VRF and VPN attributes via a previously defined VRF object, check the **Use VRF Object** check box.

For more information on this feature, see Independent VRF Management, page 5-14. That section describes how to use independent VRF objects in MPLS VPN service policies and service requests.

If you are not using the VRF object feature, then define the VRF and VPN attributes as described in the following steps:

Step 3 **Export Map:** If necessary, enter the name of the export route map.

The name of the export route map you enter here must be the name of an existing export route map on the PE.

Note IOS supports only one export route map per VRF. Therefore, there can be only one export route map per VPN.

When you use the Prime Fulfillment software to define a management VPN, Prime Fulfillment automatically generates an export route map for the management VPN. Because the Cisco IOS supports only one export route map per VRF and that route map is reserved for the management VPN, the Export Map field is not available if the VRF is part of the management VPN.

An export route map does not apply a filter; it can be used to override the default set of route targets associated with a route.

Step 4 **Import Map:** Enter the name of the import route map.

The name of the import route map you enter here must be the name of an existing import route map on the PE.

Note IOS supports only one import route map per VRF. Therefore, there can be only one import route map per VPN.

An import route map does apply a filter. Therefore, if you want to exclude a particular route from the VRF on this PE, you can either set an export route map on the sending router to make sure it does not have any route targets that can be imported into the current VRF, or create an import route map on the PE to exclude the route.

Step 5 **Maximum Routes:** Specify the maximum number of routes that can be imported into the VRF on this PE.

Note Prime Fulfillment will not allow provisioning of another value for Maximum Routes after it is configured with a value. Because a VRF might be used by multiple interfaces (links), after this value is configured for a link, it is recommended that you do not manually change it. Prime Fulfillment generates an error if you try to change the maximum routes value for an existing or new service request using this VRF.

Step 6 **Maximum Route Threshold:** Specify the threshold value for the number of maximum routes.

When the specified number of maximum routes is exceeded, Prime Fulfillment sends a warning message.

Step 7 **VRF Description:** Optionally, you can enter a description of the VRF for the current VPN.

Step 8 **BGP Multipath Load Sharing:** Check this check box to enable BGP multipath load sharing and maximum path configuration.

See BGP Multipath Load Sharing and Maximum Path Configuration, page 5-75, for details on using this option.

Step 9 **Allocate New Route Distinguisher:** A route distinguisher (RD) is a 64-bit number appended to each IPv4 route that ensures that IP addresses that are unique in the VPN are also unique in the MPLS core. This extended address is also referred to as a VPN-IPv4 address.

When **Allocate New Route Distinguisher** is enabled, create a new VRF if there is no matching VRF configuration on that PE; otherwise, reuse it.

When **Allocate New Route Distinguisher** is disabled, find the first matching VRF configuration across the entire range of PEs, regardless of the PE. If this VRF is found on the PE being configured, reuse it. If it is not found on the PE, create it.

Note The service request might get a VRF that has already been configured on another PE router.

Prime Fulfillment automatically sets the route target (RT) and RD values, but you can assign your own values by checking the VRF and RD check box instead.

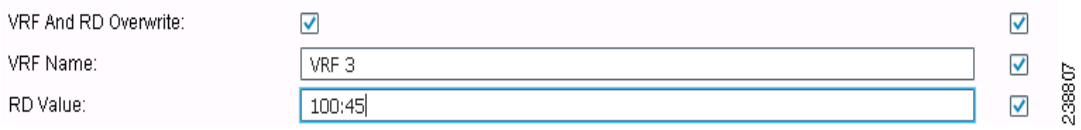
Note The **Allocate New Route Distinguisher** option is disabled if you enabled the unique route distinguisher feature when the VPN was created. For information, see **Enabling a Unique Route Distinguisher for a VPN, page 5-11**.

Step 10 **VRF and RD Overwrite:** When you enable the **VRF and RD Overwrite** option, this dialog box presents two new fields, as shown in Figure 5-6, that allow you to overwrite the default VRF name and route distinguisher values.

Caution If not done correctly, changing the default values for the VRF name and the route distinguisher value can alter or disable service requests that are currently running. Please make these changes with caution and only when absolutely necessary.

Note The **VRF and RD Overwrite** option is disabled if you enabled the unique route distinguisher feature when the VPN was created. For information, see **Enabling a Unique Route Distinguisher for a VPN, page 5-11**.

Figure 5-6 No Routing Protocol Selected a



a. **VRF Name:** Enter the new VRF name. It is recommended not to use special characters (' ` " < > ( ) [ ] { } / \ & ^ ! ? ~ * % = , . + |), as this may cause misconfiguration of the VRF name for certain devices.

b. **RD Value:** Enter the new RD value.

Note In MPLS service requests, once you specify values to sub-attributes under the VRF and RD Overwrite attribute (that is, the VRF Name and RD Value attributes) and save the service request, both of these fields are disabled and are no longer editable. This behavior was introduced because changing the default values for the VRF Name and RD Value can alter or disable currently running service requests. Therefore, if these values need to be changed on a deployed service request, the workaround is that you must decommission and purge the service request and create a new service request. In the case of a new service request that has not yet been deployed, you must force purge the service request and then create a new service with new values.

Step 11 **PE VPN Membership:** In the check box, specify the VPN associated with this service policy.

The PE VPN Membership information includes the customer name, VPN name, service provider name, CE routing community name, and whether the CERC type is a hub-and-spoke CERC or a fully meshed CERC.

If the **Is Hub** check box is checked, it indicates that the CERC type is hub-and-spoke.

Using the **Add** and **Delete** buttons, you can add a VPN to this list or delete a VPN from this list.

Step 12 If you would like to enable template and data file support for the policy, click the **Next** button to access the Template Association window, and then see Enabling Template Association for a Policy, page 5-78 for details on working with templates and data files.

Step 13 If you are satisfied with the VRF and VPN selections, click **Finish**.

The Policies window appears.

Now that you have defined a service policy for an MPLS PE-to-CE service, the service operator can now use this policy to create and deploy a service request for a PE-CE link. For details, see MPLS VPN Service Requests, page 5-78

BGP Multipath Load Sharing and Maximum Path Configuration

Prime Fulfillment supports the configuration of Border Gateway Protocol (BGP) multipath load sharing for external BGP (eBGP), internal BGP (iBGP), and external and internal BGP (eiBGP). As additional support for BGP multipath load sharing, MPLS also allows setting a unique route distinguisher (RD) per provider edge (PE) router for a virtual private network (VPN) and virtual route forwarding (VRF) table. The **BGP Multipath Load Sharing** option allows you to enable or disable BGP multipath load sharing, as shown in Figure 5-7.

Figure 5-7 Multipath Configuration Options of the VRF and VPN Membership Window

When the **BGP Multipath Load Sharing** check box is checked, additional fields are displayed for the BGP multipath action, maximum paths, import paths, and unequal cost routes. The additional fields appear dynamically in the GUI based on the **BGP Multipath Action** option you choose.

If there is no existing BGP multipath configuration, specifying multipath load sharing through these fields creates a new multipath BGP configuration for the VRF of the PE. If a BGP multipath configuration already exists, this action overwrites the existing configuration with the new multipath values. A remove option allows you to delete all existing BGP multipath configurations of a particular type for the VRF of the PE. If the **BGP Multipath Load Sharing** check box is unchecked, no BGP multipath actions are taken. See Removing a Multipath Configuration, page 5-77, for how multipath settings defined in a service request can be removed.

When a BGP multipath configuration is edited on an existing MPLS service request, all MPLS service requests on the same device with the same VPN membership are moved to the Requested state. This keeps the IPv4 and IPv6 multipath configuration synchronized.

Note For information on BGP multipath support for IOS XR devices, see BGP Multipath Support for IOS XR Devices, page 5-77.

BGP multipath is supported for IPv6 and dual stacked services. The BGP multipath configuration is configured for the VPN routing/forwarding instance (VRF). Thus, it is possible to set only one set of parameters for both IPv4 and IPv6 services.

The following sections describe each of the multipath scenarios, as determined by the type of BGP multipath selected in the **BGP Multipath Action** drop-down list. The options available in the drop-down list are:

- eBGP—Specifies the multipath configuration for eBGP. This is the default selection.
- iBGP—Specifies the multipath configuration for iBGP.
- eiBGP—Specifies the multipath configuration for both eBGP and iBGP. This option allows you to set a common shared value for maximum paths and import paths for both eBGP and iBGP.
- eBGP+iBGP—Specifies the multipath configuration for both eBGP and iBGP. This option allows you to set the maximum paths and import paths separately for both eBGP and iBGP.
- Remove—Deletes all existing BGP multipath configurations for the VRF of the PE.

Each of these scenarios is covered below.

Note When creating service requests, in the MPLS Link Editor - VPN and VRF window, an additional BGP attribute called **Force Modify Shared Multipath Attributes** appears in the GUI when the **BGP Multipath Load Sharing** check box is checked. The purpose of this attribute is to enable forced modification of the shared VRF attributes used by other links. This field is not persisted. This attribute only appears when creating service requests, not when creating policies.

eBGP Multipath

When you select the **eBGP** option, the **Maximum Paths** and **Import Paths** fields appear. Where:

- Maximum Paths—Specifies the maximum number of routes to allow in the routing table.
- Import Paths—Specifies the number of redundant paths that can be configured as backup multipaths for a VRF.

Note When setting up an eBGP multipath configuration, you must set a value for either **Maximum Paths** or **Import Paths**. Both fields cannot be blank.

iBGP Multipath

When you select the **iBGP** option, the **Maximum Paths**, **Import Paths**, and **Unequal Cost** fields appear. Where:

- Maximum Paths—Specifies the maximum number of routes to allow in the routing table. You must specify a value when setting up an iBGP multipath configuration.
- Import Paths—Specifies the number of redundant paths that can be configured as backup multipaths for a VRF.
- Unequal Cost—Enables/disables unequal-cost multipath. Unequal-cost multipath allows traffic to be distributed among multiple unequal-cost paths to provide greater overall throughput and reliability.

eiBGP Multipath

When you select the **eiBGP** option, the **Maximum Paths** and **Import Paths** fields appear. Where:

- Maximum Paths—Specifies the maximum number of routes to allow in the routing table. You must specify a value when setting up an eiBGP multipath configuration.
- Import Paths—Specifies the number of redundant paths that can be configured as backup multipaths for a VRF.

eiBGP+iBGP Multipath

When you select the **eiBGP+iBGP** option, the **Maximum Paths**, **Import Paths**, and **Unequal Cost** fields appear. Where:

- Maximum Paths—Specifies the maximum number of routes to allow in the routing table. The number of routes can be specified separately for eBGP and iBGP.
- Import Paths—Specifies the number of redundant paths that can be configured as backup multipaths for a VRF. The number of paths can be specified separately for eBGP and iBGP.
- Unequal Cost—Enables/disables unequal-cost multipath. Unequal-cost multipath allows traffic to be distributed among multiple unequal-cost paths to provide greater overall throughput and reliability.

Note The support for multipath load sharing requires unique route distinguishers (RDs) for each PE router for a VPN (VRF). This is to prevent the same RDs from being allocated to different customers. This allows the use of the same RD for the same VRF. That is, all sites in the PE can have the same unique RD. The unique RD feature is optional. It is enabled at both a global VPN level or a service request level. To enable the unique RD per PE for a VPN, the Create VPN window contains a new **Enable Unique Route Distinguisher** field. For more information on using this feature, see Enabling a Unique Route Distinguisher for a VPN, page 5-11.

BGP Multipath Support for IOS XR Devices

The following attributes are supported in Prime Fulfillment for BGP multipath configuration on IOS XR devices:

- Maximum Paths—This attribute has a range from 2 to 8 for IOS XR. When an out-of-range value is specified, the service request cannot be saved and an error is displayed. The service request will not move to an Invalid state (which occurs if a deployment is carried out).
- Unequal Cost—This attribute is supported for iBGP only.

The **Import Paths** attribute is supported in IOS but not in IOS XR.

Removing a Multipath Configuration

A multipath configuration can be removed by selecting the **Remove** option in drop-down list of the BGP Multipath Action attribute. The Remove option removes the multipath configuration for the VRF on the PE, if it is previously configured.

If a service request is saved with a multipath configuration and the configuration has to be removed, you should use the Remove option.

Note A multipath configuration cannot be removed by simply unchecking the BGP Multipath Load Sharing check box. It must be removed by setting the BGP Multipath Action attribute to Remove, and then saving the service request. You should uncheck the BGP Multipath Load Sharing check box only after removing the multipath configuration.

Enabling Template Association for a Policy

The Prime Fulfillment template feature gives you a means to download free-format CLIs to devices configured for links within an MPLS service request. If you enable templates, you can use templates and data files to download commands that are not currently supported by Prime Fulfillment.

Step 1 To enable template association for the policy, click the **Next** button in MPLS Policy Editor - VRF and VPN Membership window.

Note An additional window appears in the policy workflow before the Template Association window. This window allows you to create user-defined attributes within the policy (and service requests based on the policy). For background information on how to use the additional information feature, see Appendix H, "Adding Additional Information to Services." If you are not using this feature, click **Next** to proceed to the Template Association window, or else click **Finish** to save the policy.

The Template Association window appears. In this window, you can enable template support and, optionally, associate templates and data files with the policy. For instructions about associating templates with policies and how to use the features in this window, see Chapter 9, "Managing Templates and Data Files."

Step 2 When you have completed setting up templates and data files for the policy per the instructions in the appendix, click **Finish** in the Template Association window to close it.

The Policies window appears.

Now that you have defined a service policy for an MPLS PE-to-CE service, the service operator can now use this policy to create and deploy a service request for a PE-CE link. For details, see Chapter 5, "MPLS VPN Service Requests."

MPLS VPN Service Requests

This section contains the following sections:

To apply MPLS VPN policies to network devices, you must deploy the service request. When you deploy a service request, Prime Fulfillment compares the device information in the Repository (the Prime Fulfillment database) with the current device configuration and generates a configlet. Additionally, you can perform various monitoring and auditing tasks on service requests. These common task that apply to all types of Prime Fulfillment service requests are covered in Chapter 8, "Managing Service Requests". See that section for more information on these tasks.

Service Enhancements

With this release of MPLS VPN Management, a number of enhancements to the service function are available:

- A service is no longer limited to a single PE-CE link at a time. Under Prime Fulfillment, a service can be comprised of multiple PE-CE links per service request.
- Multicast MPLS VPNs

  A multicast address is a single address that represents a group of machines. Unlike a broadcast address, however, the machines using a multicast address have all expressed a desire to receive the messages sent to the address. A message sent to the broadcast address is received by all IP-speaking machines, whether they care what it contains or not. For example, some routing protocols use multicast addresses as the destination for their periodic routing messages. This allows machines that have no interest in routing updates to ignore them.

  To implement multicast routing, Prime Fulfillment employs the concept of a multicast domain (MD), which is a set of VRFs associated with interfaces that can send multicast traffic to each other. A VRF contains VPN routing and forwarding information for unicast. To support multicast routing, a VRF also contains multicast routing and forwarding information; this is called a Multicast VRF.
- Site of Origin support

  Although a route target provides the mechanisms to identify which VRFs should receive routes, a route target does not provide a facility that can prevent routing loops. These routing loops can occur if routes learned from a site are advertised back to that site. To prevent this, the Site of Origin (SOO) feature identifies which site originated the route, and therefore, which site should *not* receive the route from any other PE routers.

  Note The Prime Fulfillment graphical user interface (GUI) previously supported eBGP Site of Origin for IOS devices. In this release, eBGP Site of Origin is additionally supported for IPv4 eBGP neighbors on IOS XR PE devices.
- Layer 2 access into MPLS VPNs
- Provisioning PE-Only service requests

How Prime Fulfillment Accesses Network Devices

When Prime Fulfillment attempts to access a router, it uses the following algorithm:

1. Checks to see if a terminal server is associated with the device, and if this is the case, Prime Fulfillment uses the terminal server to access the device.
2. If there is no terminal server, Prime Fulfillment looks for the management interface on the device.

3. If there is no management interface, Prime Fulfillment tries to access the device using the fully-qualified domain name (host name plus domain name).

If any step in the VPN Solutions Center device-access algorithm fails, the entire device access operation fails—there is no retry or rollover operation in place. For example, if there is a terminal server and Prime Fulfillment encounters an error in attempting to access the target device through the terminal server, the access operation fails at that point. With the failure of the terminal server access method, Prime Fulfillment does not attempt to find the management interface to access the target device.

Examples of Creating MPLS VPN Service Requests

A service request is an instance of service contract between a customer edge router (CE) and a provider edge router (PE). The service request user interface asks you to enter several parameters, including the specific interfaces on the CE and PE routers, routing protocol information, and IP addressing information.You can also integrate an Prime Fulfillment template with a service request, and associate one or more templates to the CE and the PE. To create a service request, a service policy must already be defined, as described in MPLS VPN Service Policies, page 5-40

Note Subsequent chapters in this guide provide additional examples of setting up these and other MPLS VPN service requests. See also Provisioning Regular PE-CE Links, page 5-98 and Provisioning Multi-VRFCE PE-CE Links, page 5-109

MPLS VPN Topology Example

Figure 5-8 shows the topology for the network used to define the service requests in this section.

PE-CE Example

In the PE-CE example, the service provider needs to create an MPLS service for a CE (mlce1) in their customer site Acme_NY (in New York).

Multi-VRF Example

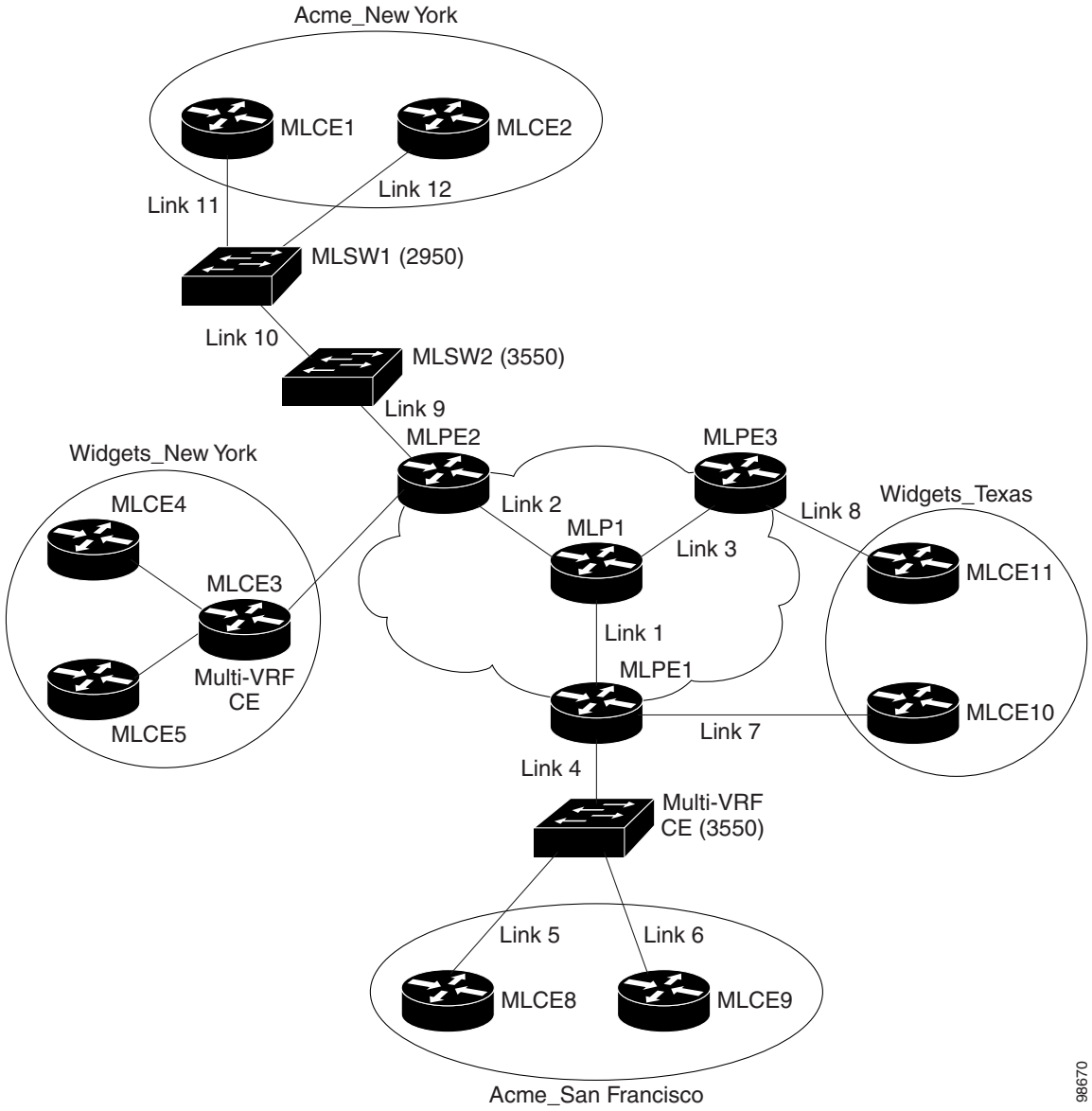
In the Multi-VRF example, the service provider needs to create an MPLS service between a CE (mlce4) in their customer site Widgets_NY (in New York) and a Multi-VRFCE (mlce3) located in their customer site Widgets_NY (in New York).

The goal is to create a single service request that defines a link between the customer site in New York and the PE (mlpe2).

PE-Only Example

In the PE-Only example, the service provider needs to create an MPLS service for a PE (mlpe2.)

Figure 5-8 Example Network Topology



Creating an MPLS VPN PE-CE Service Request

For an example of creating an MPLS VPN PE-CE service request, perform the following steps:

Step 1 Choose **Operate> Service Requests > MPLS**.

Step 2 Choose the policy of choice, then click **OK**.

The MPLS Service Request Editor appears.

Step 3 Click **Add Link**.

The MPLS Service Request Editor now displays a set of fields. Notice that the Select CE field is enabled. Specifying the CE for the link is the first task required to define the link for this service.

Step 4 **CE:** Click **Select CE**.

The Select CPE Device window appears.

a. From the "Show CPEs with" drop-down list, you can display CEs by Customer Name, by Site, or by Device Name.

b. You can use the **Find** button to either search for a specific CE, or to refresh the display.

c. You can set the "Rows per page" to **5**, **10**, **20**, **30**, **40**, or **All**.

d. This dialog box displays the first page of the list of currently defined CE devices. The number of pages of information is displayed in the lower right corner of the dialog box. To go to the another page of CE devices, click the number of the page you want to go to.

Step 5 In the Select column, choose the name of the CE for the MPLS link, then click **Select**.

You return to the Service Request Editor window, where the name of the selected CE is now displayed in the CE column.

Step 6 **CE Interface:** Choose the CE interface from the drop-down list.

Note that in the PE column, the **Select PE** option is now enabled.

Note on Using Bundle-Ether Interfaces

The following usage notes apply to Bundle-Ether interfaces:

- You can select a Bundle-Ether interface for an IOS XR device based on the interface type specified in the corresponding policy.
- Bundle-Ether interfaces are only visible in the service request if one or more Bundle-Ether interfaces are pre-configured on the selected PE device. That is, port channel must be preconfigured on the device prior to creating the service request. Port channel interfaces are used for VRF termination.
- Links can be IPv4 and/or IPv6. Note the following points:
  - On the Cisco Carrier Routing System One (CRS-1) router, both IPv4 and IPv6 links are supported. Multicast is not supported for IPv6. See the following link for more information:

    http://www.cisco.com/en/US/docs/ios_xr_sw/iosxr_r3.8/interfaces/command/reference/hr38lbun.html#wp1410649

    http://www.cisco.com/en/US/docs/ios_xr_sw/iosxr_r3.8/multicast/configuration/guide/mc38mcst.html#wp1168111

    http://www.cisco.com/en/US/docs/ios_xr_sw/iosxr_r3.8/multicast/configuration/guide/mc38mcst.html#wp1290965
  - On the Cisco 12000 (also known as a Gigabit Switch Router or GSR), only IPv4 links are supported; this is a device restriction. See the following link for more information:

    http://www.cisco.com/en/US/docs/ios/12_0s/feature/guide/lnkbndl.html
- The multiple neighbor and peering with bundled physical interface feature is not supported for MVRFCE service requests.

Step 7 **PE:** Click **Select PE**.

The Select PE Device dialog box appears.

a. From the "Show PEs with" drop-down list, you can display PEs by Customer Name, by Site, or by Device Name.

b. You can use the **Find** button to either search for a specific PE, or to refresh the display.

c. You can set the "Rows per page" to **5**, **10**, **20**, **30**, **40**, or **All**.

d. This dialog box displays the first page of the list of currently defined PE devices. The number of pages of information is displayed in the lower right corner of the dialog box.

   To go to the another page of PE devices, click the number of the page you want to go to.

Step 8 In the Select column, choose the name of the PE for the MPLS link, then click **Select**.

You return to the Service Request Editor window, where the name of the selected PE is now displayed in the PE column.

Step 9 **PE Interface:** Choose the PE interface from the drop-down list.

Note that the Link Attribute **Add** option is now enabled.

See the section Note on Using Bundle-Ether Interfaces, page 5-82, for information on specifying Bundle-Ether interfaces.

Step 10 In the Link Attribute column, click **Add**.

The MPLS Link Attribute Editor window appears, showing the fields for the interface parameters.

The field values displayed in this dialog box reflect the values specified in the service policy associated with this service. For details on each of the PE and CE interface fields, see Specifying PE and CE Interface Parameters, page 5-42.

Notes on the VLAN ID and Second VLAN ID Attributes

The VLAN ID is shared between the PE and CE, so there is one VLAN ID for both.

The Second VLAN ID is an optional attribute that provides a method to match the Q-in-Q second VLAN tag of incoming frames on the PE interface.

Usage notes:

- This attribute is not available for service requests based on MVRFCE policies.
- This attribute does not exist at the policy level and must be set while creating the service request. There is no corresponding autopick option for the second VLAN ID, so a value must be supplied. It must be an integer from 1 to 4094.
- This attribute is only applicable for regular PE-CE links. It is supported both when the CE is present and when it is not present. It is supported for both managed and unmanaged CE devices.
- This attribute is only applicable when the encapsulation type for the PE interface is dot1q. For all other encapsulation types, this attribute does not appear in GUI.
- This feature is available for limited platforms (only those that support Q-in-Q matching). If service requests with second VLAN ID are deployed on unsupported platforms it results in a deployment failure. In such cases, the operator can remove the second VLAN ID and redeploy the service. This would be a service-affecting operation, since the IP address is also removed and redeployed during the change.
- A service request created with a second VLAN ID results in the following command on the IOS device:

  encapsulation dot1q *VLAN_ID* **second-dot1q** *SECOND_VLAN_ID*
- A service request created with a second VLAN ID results in the following command on the IOS XR device:

  dot1q vlan *VLAN_ID SECOND_VLAN_ID*
- Prime Fulfillment does not apply the second VLAN. It only supports the second VLAN matching on the PE interface.
- The second VLAN ID attribute is available for use as a template variable (*Second_PE_Vlan_ID*).

- For additional information on second VLAN ID and Q-in-Q support, see the following sections:
  - CE-PE L3 MPLS VPN (Q-in-Q/Second VLAN ID, IOS), page 5-182
  - CE-PE L3 MPLS VPN (Q-in-Q/Second VLAN ID, IOS XR), page 5-184
  - Frequently Asked Questions, page 5-247

Step 11 Edit any interface values that must be modified for this particular link, then click **Next**.

The MPLS Link Attribute Editor for the IP Address Scheme appears. The field values displayed in this dialog box reflect the values specified in the service policy associated with this service. For details on the IP address scheme fields, see Specifying the IP Address Scheme, page 5-45.

Step 12 Edit any IP address scheme values that must be modified for this particular link, then click **Next**.

The MPLS Link Attribute Editor for Routing Information window appears.

The field values displayed in this dialog box reflect the values specified in the service policy associated with this service. For details on the routing information for the PE and CE, see Specifying the Routing Protocol for a Service, page 5-48.

Because the service policy used for this service specified the routing protocol as editable, you can change the routing protocol for this service request as needed.

Note For the Static routing protocol, there are two additional attributes that you can add via the Link Attribute Editor. See Setting Static Routing Protocol Attributes (for IPv4 and IPv6), page 5-90.

Step 13 Edit any routing protocol values that must be modified for this particular link, then click **Next**.

Note If this interface is dual stacked (IPv4 and IPv6), you will be prompted to enter the routing information for both IPv4 and IPv6 independently.

The MPLS Link Attribute Editor for the VRF and VPN attributes appears. The field values displayed in this dialog box reflect the values specified in the service policy associated with this service. For details on the VRF and VPN information, see Defining VRF and VPN Information, page 5-72.

Note If you want to set the VRF and VPN attributes via a previously defined VRF object, check the **Use VRF Object** check box. For more information on this feature, see Chapter 5, "Independent VRF Management." That section describes how to use independent VRF objects in MPLS VPN service policies and service requests.

Step 14 If multicast is enabled, choose the PIM (Protocol Independent Multicast) Mode:

- SPARSE_MODE
- SPARCE_DENSE_MODE

Tip Multicast routing architecture allows the addition of IP multicast routing on existing IP networks. PIM is an independent unicast routing protocol. It can be operated in two modes: dense and sparse.

Step 15 Edit any VRF and VPN values that must be modified for this particular link.

Note Most of the attributes available in the MPLS Link Attribute Editor - VRF and VPN window are covered in the VRF and VPN Member window of the policy workflow. For information on the common attributes, see Defining VRF and VPN Information, page 5-72. However, there are some differences when defining the VRF and VPN attributes in service requests. See Defining VRF and VPN Attributes in an MPLS Service Request, page 5-85 for information on defining VRF and VPN attributes during service request creation.

Step 16 Click the **Next** button if you want to associate templates or data files to the service request.

The Template Association window appears. In this window, you can associate templates and data files with a device by clicking the **Add** button in Template/Data File column for the device. When you click the **Add** button, the Add/Remove Templates window appears.

For instructions about associating templates with service requests and how to use the features in this window, see Chapter 9, "Managing Templates and Data Files." When you have completed setting up templates and data files for the device(s), click **Finish** in the Template Association window to close it and return to the Service Request Editor window.

Step 17 If you did not add templates, click **Finish** in the MPLS Link Editor – VRF and VPN window.

You return to the MPLS Service Request Editor. You can define multiple links in this service request, following the steps outlined in previous steps.

Step 18 To save your work on this first link in the service request, click **Save**.

You return to the Service Requests window, where the information for the link you just defined is now displayed.

As you can see, the service request is in the Requested state. When all the links for this service have been defined, you must deploy the service, as described in Migrating PE Devices from IOS to IOS XR, page 5-98.

Note By default, all service requests in the Prime Fulfillment system are shown in the Service Request window. You can filter the list of service requests to be displayed by choosing different selections from the **Show Services with**, **matching**, and **of type** drop-down lists and clicking the **Find** button.

Note If you have only ACTIVATION, L3MPLSVPN, and VPN licenses installed for Prime Fulfillment, you cannot display all service requests based on the VPN used (by choosing **VPN Name** in the **Show Services with** drop-down list, where **Type** is **All**). The workaround for this is to display the service requests based the MPLS VPN type (by choosing **MPLS VPN** in the **of type** drop-down list). This problem does not occur if all Prime Fulfillment licenses are installed.

Defining VRF and VPN Attributes in an MPLS Service Request

Most of the attributes available in the MPLS Link Attribute Editor - VRF and VPN window are described in the discussion of the VRF and VPN Member window of the MPLS policy workflow. For information on defining and using these common attributes, see Defining VRF and VPN Information, page 5-72 in MPLS VPN Service Policies, page 5-40 However, there are some differences when defining the VRF and

VPN attributes in service requests. There are two cases to consider, depending on whether the MPLS service request is using a VPN or if it is using an in independent VRF object. These cases are covered in separate sections below.

Case 1: Using a VPN

If the service request is using a VPN, you can create an MPLS VPN link in the service request with the RD Format and RD Overwrite attributes.

Perform the following steps:

Step 1 **Use VRF Object:** Leave this check box unchecked.

Checking this check box causes most of the attributes to disappear from the window. This case is covered in the next section, Case 2: Using an Independent VRF Object, page 5-89.

Step 2 **RD Format:** Choose an RD format from the drop-down list. The choices are:

- RD_AS—Route distinguisher in AS format. This is the default.
- RD_IPADDR—Route distinguisher in IP address format.

Usage notes:

- If you select RD_IPADDR as the RD format, the GUI refreshes and displays a new attribute: RD IP Address Value.
- You must either manually enter the RD IP Address Value in the provided text field or else select a loopback IP address of the PE device used in the service request. To do the latter, click the **Select Loopback IP** button and choose the desired loopback interface in the dialogue box.
- Prime Fulfillment validates the IP address entered.
- Only basic IPv4 addresses are allowed. No network prefixes are permitted.
- The RD is formed by appending to the IP address the VPN ID picked from the RD pool of the respective provider.

Note If you select RD_IPADDR as the RD format and use a VPN with a VPN ID greater than 65535, the service request goes to the **Failed Deploy** state. The reason is that if the first part of the RD value is an IP address (which is 32 bits), the second part of the RD can be only16 bits (which equates to a value from 1 to 65535).

- The RD options are disabled when subsequently editing the service request.
- When multiple service requests with the same VPN having “manual/loopback IP” entry for RD IP Address are deployed on multiple PEs, new VRFs with unique RDs are created. This is because RD IP Address (manual/loopback IP) might differ for different devices.
- The following Prime Fulfillment template variables support RD Format:
  - RD_FORMAT
  - RD_IPADDRESS

Step 3 Check the **Unique Route Distinguisher:** and **Allocate New Route Distinguisher:** check boxes based on the RD Format selection.

Step 4 **PE VPN Membership: Specify the VPN associated with this service policy.**

Usage notes:

- The PE VPN Membership information includes the customer name, VPN name, service provider name, Route Targets name, Route Targets type, and whether the Route Targets type is a hub-and-spoke Route Targets or a fully meshed Route Targets.
- If you choose a VPN that is already being used in a service request using the same PE, the same RD Format and RD IP Address Value is picked for the new service request and the RD Format and RD IP Address Value attributes are disabled.
- If you choose an IPv4, IPv6, or "dual-stacked" (both IPv4 and IPv6) VPN, additional attributes (Enable IPv4 Multicast and Enable IPv6 Multicast) appear in the VRF and VPN window.
- For details on using the CERC Type attribute, see the section Adding Independent IPv4 and IPv6 Route Targets for MPLS Service Requests, page 5-87.

Migrating Existing Service Requests to the New RD Format

To migrate existing service requests to be able to use the RD format, you must do the following:

- Decommission the service request.
- Redeploy the service request using RD Format, or check the **VRF and RD Overwrite:** check box to overwrite the RD Value using the new format (*ip_address:vpn_id*).

Note Once you specify values to sub-attributes under the VRF and RD Overwrite attribute (that is, the VRF Name and RD Value attributes) and save an MPLS service request, both of these fields are disabled and are no longer editable. This behavior was introduced in because changing the default values for the VRF Name and RD Value can alter or disable currently running service requests. Therefore, if these values need to be changed on a deployed service request, the workaround is that you must decommission and purge the service request and create a new service request. In the case of a new service request that has not yet been deployed, you must force purge the service request and then create a new service with new values.

Adding Independent IPv4 and IPv6 Route Targets for MPLS Service Requests

Prime Fulfillment supports independent IPv4 and IPv6 route targets (RTs) for Route Targets. You can configure this feature using the Route Targets Type attribute.

Usage notes:

- During service request creation, you can specify the RT type of a Route Target in the PE VPN Membership section of the VRF and VPN window. It is specified in a drop-down list in the Route Targets Type column. The list choices are:
  - IPv4. If you select IPv4, the corresponding Route Targets are applied to the **ipv4 address-family** CLI in the device configuration.
  - IPv6. If you select IPv6, the corresponding Route Targets are applied to the **ipv6 address-family** CLI in the device configuration.
  - IPv4 and IPv6 (dual-stacked). If you select IPv4 and IPv6, the same RTs are applied for both address families.
- The choices available in the Route Targets Type drop-down list depend on the IP addressing scheme selected for the service request. This is determined by the IP Number Scheme attribute in the IP Addressing Scheme window of the MPLS Link Editor workflow.
- If you select IPV4 and IPV6 address family, the Route Targets type should be one of the following:
  - Single Route Target: IPV4 and IPV6

  - Two (or more) individual Route Targets: At least one of type IPv4 and the other(s) of type IPv6

  If you do not do this, Prime Fulfillment generates an error.

- If an existing service request is deployed only for IPv4 and you later modify the service request as dual-stacked (IPv4 and IPv6), Prime Fulfillment changes the tagging for the Route Targets added based on the address family. This also applies to a case in which the service request is modified from IPv6 to dual-stacked (IPv4 and IPv6).
- When modifying a service request, if the Route Targets type is changed, you can add or remove Route Targets/VPNs also.
- If VPN association is set up at the policy level and specified as non-editable, then while creating a service request using this policy, the tagging of the Route Targets types is decided based on the address family that was chosen in the policy.
- If an existing dual-stacked (IPv4 and IPv6) service request is modified to the IPv4 or IPv6 address family, Prime Fulfillment automatically changes the Route Targets tagging to the selected address family.
- Prime Fulfillment checks for other service requests on the same PE that are using the same VPN, to make sure that RTs being used by other service requests are not modified or removed.
- The independent RTs for IPv4 and IPv6 feature is supported with the VRF and RD Overwrite option.
- The independent RTs for IPv4 and IPv6 feature is not supported for MVRFCE service requests.
- The independent RTs for IPv4 and IPv6 feature is not supported for independent VRF service requests and MPLS service requests using an independent VRF.
- This feature is controlled through the DCPL property GUI\MplsVPN\UniqueRTFeatureEnable. The default value for this property is false. To use the independent RTs for IPv4 or IPv6 feature, you must set the DCPL property to true. Controlling the feature through a DCPL property ensures that other customers' flows are not affected (that is, those who do not want to use this feature). Customers who desire to use this feature can enable it through the DCPL property.
- The following template variables are supported for independent RTs:
  - MPLSExportRouteTargets—Template variable for export RTs under IPv4 address family.
  - MPLSImportRouteTargets—Template variable for import RTs under IPv4 address family.
  - MPLSExportRouteTargets_IPV6—Template variable for export RTs under IPv6 address family.
  - MPLSImportRouteTargets_IPV6—Template variable for import RTs under IPv6 address family.
- The following example shows how the template variables might be used in a template file.

```
vrf MyVRF2
address-family ipv4 unicast
import route-target
#foreach($name in $MPLSImportRouteTargets)
$name
#end
export route-target
#foreach($name in $MPLSExportRouteTargets)
$name
#end
address-family ipv6 unicast
import route-target
#foreach($name in $MPLSImportRouteTargets_IPV6 )
$name
#end
export route-target
#foreach($name in $MPLSExportRouteTargets_IPV6 )
```

```
$name
#end
```

- For example configlets of this feature, see PE L3 MPLS VPN (Outgoing Interface + Next Hop IP Address, Static Route Configuration, IOS XR and IOS), page 5-244.

Case 2: Using an Independent VRF Object

If the service request is using an independent VRF object, you can specify the RD attributes as described in this section. For general coverage of creating VRF objects, working with VRF service requests, and using VRF objects in MPLS VPN policies and service requests, see Independent VRF Management, page 5-14

Perform the following steps:

Step 1 **Use VRF Object:** Check the check box for this attribute.

Checking this check box causes most of the attributes to disappear from the window.

Step 2 **VRF Object:** Click the **Select** button to select a previously created VRF object.

The Select Independent VRF window appears.

Step 3 Click a radio button to choose a VRF object.

Step 4 **Unique RD:** Check this check box to assign a unique RD and to ensure a unique RD allocation for each VRF on all PEs of the VPN.

Note For more information on the unique RD feature in Prime Fulfillment, see Enabling a Unique Route Distinguisher for a VPN, page 5-11.

Step 5 Click **Select** to confirm the VRF object selection.

The VRF and VPN window reappears showing the selected VRF object in the VRF Object field.

Usage notes:

- If you select a VRF object with RD in IP address format (RD_IPADDR) and with Autopick RD enabled, then the RD Value while selecting the VRF shows up in the form *IP:vpn_id*. And if a manual RD is entered, it would be in the form *ip_address:vpn_id*, where *ip_address* is an IPv4 address and *vpn_id* is a 4-byte integer value.
- If during the creation of the independent VRF object you selected RD_IPADDR as the RD format and enabled Autopick RD, either you can manually enter the RD IP Address Value in the text field provided or you can click the **Select Loopback IP** button to choose a loopback IP address of the PE device used in the service request.
- Prime Fulfillment validates the IP address entered. Only basic IPv4 addresses are allowed. No network prefixes are permitted.
- The RD is formed by appending to the IP address the VPN ID picked from the RD pool of the respective provider.
- After the VRF service request is deployed with the RD using the IP address entered, the RD IP Address Value field is disabled and cannot be edited.
- If you choose a VRF which is already used in a service request using the same PE, the same RD IP Address Value is picked for the existing service request. The RD IP Address Value options are disabled.

- If you want to change the RD Format to a new format in the case of a VRF object that is already deployed on a device, it is only possible under the following conditions:
  - All related MPLS service requests are decommissioned and purged.
  - The VRF service request is decommissioned, purged, and redeployed.
- Unique RD can be enabled for the VRF.

Step 6 Click **Next** to continue setting the MPLS link attributes.

Viewing Configlets Generated by the MPLS VPN Service Request

To view configlets generated on the PE and CE device by the MPLS VPN service request, perform the following steps:

Step 1 To view the PE and CE configlets for a service request that has been successfully deployed, from the Service Request window, choose the service request you want to see, then click **Details**.

The Service Request Details window appears for the associated job number.

Step 2 From Service Request Details window, click **Configlets**.

The Service Request Configlets window appears.

Step 3 Choose the IP address for the desired configlet, then click **View Configlet**.

For additional information about viewing device configlets for a deployed service request, see Viewing Service Request Configlets, page 8-6. For sample configlets, see Sample Configlets, page 5-165

Setting Static Routing Protocol Attributes (for IPv4 and IPv6)

For the static routing protocol, in addition to the attributes that you can specify in the service policy, there are additional attributes that you can add via the Link Attribute Editor.

- **Advertised Routes for CE:** allows you to add a list of IP addresses, static routes to put on the PE, that describes all the address space in the CE's site.
- **Routes to Reach other Sites:** allows you to add a list of IP addresses, static routes to put on the CE, that describes all the address space throughout the VPN.

IPv4 Routing Information

For configuring IPv4 routing information, perform the following steps:

Step 1 When you perform Step 12 in the section Creating an MPLS VPN PE-CE Service Request, page 5-81 for static routing protocols, the MPLS Link Attribute Editor for Routing Information appears.

You can edit **Advertised Routes for CE:** and **Routes to Reach other Sites:** for this service request.

Step 2 To edit **Advertised Routes for CE:,** click **Edit.**

The Advertised Routes window appears.

Step 3 Click **Add** to add IP addresses.

The Advertised Routes window appears again.

Step 4 Enter an IP address and a metric.

Step 5 Click **Add** to add another IP address or click **OK**.

Step 6 To edit **Routes to Reach Other Sites:**, click **Edit.**

The Routes to reach other sites window appears.

Step 7 Click **Add** to add IP addresses.

The Routes to reach other sites window appears again.

Step 8 Enter an IP address and a metric.

Step 9 Click **Add** to add another IP address or click **OK**.

Step 10 Choose a **Next Hop Option:**

- USE_OUT_GOING_INTF_NAME
- USE_NEXT_HOP_IPADDR
- OUTGOING_INTF_NAME+NEXT_HOP_IPADDR

  For additional information on this choice, see Outgoing Interface Name + Next Hop IP Address Support for Static Route Configuration, page 5-92.

Step 11 Enter an IP address (in IPv4 format) in the **Next Hop IP Address:** field, if applicable.

IPv6 Routing Information

For configuring IPv6 routing information, perform the following steps:

Step 1 When you perform Step 12 in the section Creating an MPLS VPN PE-CE Service Request, page 5-81 for static routing protocols, the MPLS Link Attribute Editor for Routing Information appears.

You can edit **Advertised Routes for CE:** for this service request.

Step 2 To edit **Advertised Routes for CE:,** click **EDIT**.

The Advertised Routes window appears.

Step 3 Click **Add** to add IP addresses.

The Advertised Routes window appears again.

Step 4 Enter an IP address and a metric.

Step 5 Click **Add** to add another IP address or click **OK**.

Step 6 Click **Add** to add IP addresses.

Step 7 Click **Add** to add another IP address or click **OK**.

Step 8 Choose a **Next Hop Option:**

- USE_OUT_GOING_INTF_NAME
- USE_NEXT_HOP_IPADDR
- OUTGOING_INTF_NAME+NEXT_HOP_IPADDR

  For additional information on this choice, see Outgoing Interface Name + Next Hop IP Address Support for Static Route Configuration, page 5-92.

Step 9 Enter an IP address (in IPv6 format) in the **Next Hop IP Address:** field, if applicable.

For information on formats supported formats for entering IPv6 addresses, see MPLS VPN Policies, page 5-34.

Outgoing Interface Name + Next Hop IP Address Support for Static Route Configuration

Prime Fulfillment provides the ability to specify the outgoing interface name and next hop IP address when creating MPLS service requests for STATIC routing protocol. You do this by choosing OUTGOING_INTF_NAME+NEXT_HOP_IPADDR from the drop-down list of the Next Hop Option attribute in the MPLS Link Attribute Editor - IPv4/IPv6 Routing Information window in the MPLS service creation workflow.

When you create a service request, you set the routing protocol attributes in the MPLS Link Attribute Editor - IPv4/IPv6 Routing Information window. When you set the Routing Protocol attribute to STATIC, the window displays related attributes, including the Next Hop Option.

Usage notes:

- The OUTGOING_INTF_NAME+NEXT_HOP_IPADDR selection in the Next Hop Option drop-down list enables you to provide an outgoing interface name and next hop IP address. Prime Fulfillment supports this format for static route configuration in the following form:

  network_address + *outgoing_interface_name* + *next_hop_address*

  Example: 69.82.224.99/32 GigabitEthernet0/0/0/0 66.174.25.0.

- This format is supported for:
  - PE_CE and PE_NO_CE service requests
  - IPv4 and IPv6 addressing
  - IOS and IOS XR devices
- This feature is configured only on the PE device.
- You can configure the network address by clicking the Edit button of Advertise Routes for CE attribute.
- The following template variables are supported.
  - IPv4 address family:

    Advr_Routes_IP_Address—Network IPv4 address for IPv4 address family.

    Advr_Routes_Metric—Metric value for IPv4 address family.

    STATIC_NEXT_HOP_IP_ADDR—Next hop IPv4 IP address for IPv4 address family.
  - IPv6 address family:

    Advr_Routes_IPV6_Address—Network IPv6 address for IPv6 address family.

    Advr_Routes_Metric_IPV6—Metric value for IPv6 address family.

    STATIC_NEXT_HOP_IPV6_ADDR—Next hop IPv6 IP address for IPv6 address family.
- The following example shows how the template variables might be used in a template file for an IOS device:

```
ip route vrf V2:TempIOS $Advr_Routes_IP_Address 255.255.255.255 $PE_Intf_Name
$STATIC_NEXT_HOP_IP_ADDR $Advr_Routes_Metric
```

- The following example shows how the template variables might be used in a template file for an IOS XR device:

```
router static
```

```
 vrf V21:TempIOSXR
  address-family ipv4 unicast
   $Advr_Routes_IP_Address $PE_Intf_Name $STATIC_NEXT_HOP_IP_ADDR
$Advr_Routes_Metric
  !
  address-family ipv6 unicast
   $Advr_Routes_IPV6_Address $PE_Intf_Name $STATIC_NEXT_HOP_IPV6_ADDR
$Advr_Routes_Metric_IPV6
```

- For example configlets of this feature, see PE L3 MPLS VPN (Outgoing Interface + Next Hop IP Address, Static Route Configuration, IOS XR and IOS), page 5-244.

Creating a Multi-VRF Service Request

MPLS-VPNs provide security and privacy as traffic travels through the provider network. The CE router has no mechanism to guarantee private networks across the traditional LAN network. Traditionally to provide privacy, either a switch needed to be deployed and each client be placed in a separate VLAN or a separate CE router is needed per each client's organization or IP address grouping attaching to a PE. These solutions are costly to the customer as additional equipment is needed and requires more network management and provisioning of each client site.

Multi-VRF, introduced in Cisco IOS release 12.2(4)T, addresses these issues. Multi-VRF extends limited PE functionality to a CE router in an MPLS-VPN model. A CE router now has the ability to maintain separate VRF tables in order to extend the privacy and security of an MPLS-VPN down to a branch office rather than just at the PE router node.

CE routers use VRF interfaces to form a VLAN-like configuration on the customer side. Each VRF on the CE router is mapped to a VRF on the PE router. With Multi-VRF, the CE router can only configure VRF interfaces and support VRF routing tables. Multi-VRF extends some of the PE functionality to the CE router—there is no label exchange, there is no LDP adjacency, there is no labeled packet flow between PE and CE. The only PE-like functionality that is supported is the ability to have multiple VRFs on the CE router so that different routing decisions can be made. The packets are sent toward the PE as IP packets.

To create a Multi-VRFCE PE-CE service request, perform the following steps:

Step 1 Choose **Operate > Service Requests >Service Request Manager**.

Step 2 Choose the MPLS Policy and click **OK**.

The MPLS Service Request Editor window appears.

Step 3 Click **Add Link**.

Step 4 Click **Select CE**.

The Select CPE Device - CE window appears.

Step 5 Choose the **CPE** Device (mlce4) and then click **Select**.

The MPLS Service Request Editor - CE Interface window appears.

Step 6 Choose the **CE Interface** from the drop-down box.

Step 7 Click **Select MVRFCE**.

The Select CPE Device - MVRFCE window appears.

Step 8 Choose the **MVRFCE** and then click **Select**.

The MPLS Service Request Editor - MVRFCE CE Facing Interface window appears.

Step 9 Choose the **MVRFCE CE Facing Interface** from the drop-down box.

The MPLS Service Request Editor - Choose MVRFCE PE Facing Interface window appears.

Step 10 Click **Select PE**.

The Select PE Device window appears.

Step 11 Choose the **PE** and then click **Select**.

The MPLS Link Attribute Editor - Interface window appears.

Step 12 Choose the **PE Interface** from the drop-down box.

Step 13 Click **Add** in the **Link Attribute** cell.

The MPLS Link Attribute Editor - Interface window appears.

Step 14 Enter the VLAN ID for the PE. (**510**)

Step 15 Click **Next**.

The MPLS Link Attribute Editor - Interface window appears.

Step 16 Enter the VLAN ID for the MVRFCE (**530**).

Step 17 Click **Next**.

The MPLS Link Attribute Editor - IP Address Scheme window appears.

Step 18 Keep the defaults, and click **Next**.

The MPLS Link Attribute Editor - IP Address Scheme window appears.

Step 19 Keep the defaults, and click **Next**.

The MPLS Link Attribute Editor - Routing Information window reappears.

Step 20 Keep the defaults and click **Next**.

The MPLS Link Attribute Editor - VRF and VPN window appears.

Note For more information on setting the VRF and VPN attributes in MPLS VPN service requests, see Defining VRF and VPN Attributes in an MPLS Service Request, page 5-85.

Step 21 Click **Add** to choose a VPN.

The Select VPN window appears.

Step 22 Choose a **VPN**.

Step 23 Click **Join as Hub** or **Join as Spoke** to join the CERC.

Step 24 Click **Done**.

The MPLS Link Attribute Editor - VRF and VPN window reappears.

Step 25 Click the **Next** button if you want to associate templates or data files to the service request.

The Template Association window appears. In this window, you can associate templates and data files with a device by clicking the **Add** button in Template/Data File column for the device. When you click the **Add** button, the Add/Remove Templates window appears. For instructions about associating templates with service requests and how to use the features in this window, see Chapter 9, "Managing Templates and Data Files." When you have completed setting up templates and data files for the device(s), click **Finish** in the Template Association window to close it.

The Service Request Editor window appears.

Step 26 If you did not add templates, click **Finish** in the MPLS Link Editor – VRF and VPN window.

The MPLS Service Request Editor window appears.

Step 27 Enter the service request description and then click **Save**.

The MPLS Service Requests window appears showing that the service request is in the Requested state and ready to deploy.

Creating a PE-Only Service Request

To create a PE-only service request, perform the following steps:

Step 1 Choose **Operate > Service Requests >Service Request Manager**.

Step 2 Choose the policy that has CE *not* present, then click **OK**.

The MPLS Service Request Editor appears.

Step 3 Click **Add Link**.

The MPLS Service Request Editor now displays a set of fields. Notice that the Select PE field is enabled. Specifying the PE for the link is the first task required to define the link for this service, unless a CLE switch link is needed. If a CLE switch is needed go to "Adding a CLE to a Service Request" section on page 5-97.

Step 4 PE: Click **Select PE**.

The Select PE Device dialog box appears.

a. From the "Show PEs with" drop-down list, you can display PEs by Provider Name, by Region, or by Device Name.

b. You can use the **Find** button to either search for a specific PE, or to refresh the display.

c. You can set the "Rows per page" to **5**, **10**, **20**, **30**, **40**, or **All**.

d. This dialog box displays the first page of the list of currently defined PE devices. The number of pages of information is displayed in the lower right corner of the dialog box.

To go to the another page of PE devices, click the number of the page you want to go to.

Step 5 In the Select column, choose the name of the PE for the MPLS link, then click **Select**.

You return to the Service Request Editor window, where the name of the selected PE is now displayed in the PE column.

Step 6 **PE Interface:** Choose the PE interface from the drop-down list.

Note that the Link Attribute **Add** option is now enabled.

Step 7 In the Link Attribute column, click **Add**.

The MPLS Link Attribute Editor appears, showing the fields for the interface parameters.

The field values displayed in this window reflect the values specified in the service policy associated with this service. For details on the PE interface fields, see Specifying PE and CE Interface Parameters, page 5-42.

Note For information on setting the VLAN ID and Second VLAN ID attributes, see Notes on the VLAN ID and Second VLAN ID Attributes, page 5-83.

Step 8 Edit any interface values that must be modified for this particular link, then click **Next**.

The MPLS Link Attribute Editor for the IP Address Scheme appears. The field values displayed in this dialog box reflect the values specified in the service policy associated with this service. For details on the IP address scheme fields, see Specifying the IP Address Scheme, page 5-45.

Step 9 Edit any IP address scheme values that must be modified for this particular link, then click **Next**.

The field values displayed in the window reflect the values specified in the service policy associated with this service. For details on the routing information for the PE, see Specifying the Routing Protocol for a Service, page 5-48.

Because the service policy used for this service specified the routing protocol as editable, you can change the routing protocol for this service request as needed.

Step 10 If you check **Site of Origin**, the screen updates to include the required step of selecting a value:

a. Click **Select**.

The Site for SOO Value window appears.

b. From the available list shown, check the check box associated with a site and its SOO value, then click **Select**.

Usage notes:

- The Site of Origin attribute is for IOS devices only. It does not show up at the policy level, but only appears in MPLS Link Attribute Editor window of the service request workflow. In addition, it only shows up in the case of a PE-only service request (that is, PE with no CE present).
- The Prime Fulfillment graphical user interface (GUI) previously supported eBGP Site of Origin for IOS devices. In this release, eBGP Site of Origin is additionally supported for IPv4 eBGP neighbors on IOS XR PE devices.
- There are two use cases to mention:
  1. If Site of Origin is enabled for a customer and the same customer is used to create a VPN used in a service request, the Site of Origin option is visible in the MPLS Link Attribute Editor window (when BGP is selected for the routing protocol). In the case of service request for a PE with no CE, when Site of Origin is enabled, the Route Map/Policy In field is disabled and cleared.
  2. If a customer is enabled for Site of Origin and the CE device uses the same customer and is used in a service request for a PE with a CE, then the Site of Origin field is not visible at the service request level. By default it takes the Site of Origin value into consideration and deploys the Site of Origin configuration to the device. As in the previous case, the Route Map/Policy In field is disabled and cleared.

Step 11 Edit any routing protocol values that must be modified for this particular link.

Note If this interface is dual stacked (IPv4 and IPv6), you will be prompted to enter the routing information for both IPv4 and IPv6 independently. When specifying IPv6 routing protocol information, the MPLS Link Attribute Editor for Routing Information may show a slightly different set of options. For information on formats supported for entering IPv6 addresses, see MPLS VPN Policies, page 5-34.

Step 12 Click **Next**.

The MPLS Link Attribute Editor for the VRF and VPN attributes appears. The field values displayed in this dialog box reflect the values specified in the service policy associated with this service. For details on the VRF and VPN information, see Defining VRF and VPN Information, page 5-72.

Note If you want to set the VRF and VPN attributes via a previously defined VRF object, check the **Use VRF Object** check box. For more information on this feature, see Independent VRF Management, page 5-14. That section describes how to use independent VRF objects in MPLS VPN service policies and service requests.

Note For more information on setting the VRF and VPN attributes in MPLS VPN service requests, see Defining VRF and VPN Attributes in an MPLS Service Request, page 5-85.

Step 13 Edit any VRF and VPN values that must be modified for this particular link.

Step 14 Click the **Next** button, if you want to associate templates or data files to the service request.

The Template Association window appears. In this window, you can associate templates and data files with a device by clicking the **Add** button in Template/Data File column for the device. When you click the **Add** button, the Add/Remove Templates window appears. For instructions about associating templates with service requests and how to use the features in this window, see Chapter 9, "Managing Templates and Data Files." When you have completed setting up templates and data files for the device(s), click **Finish** in the Template Association window.

The Service Request Editor window appears. You can define multiple links in this service request by following the steps outlined in the previous steps.

Step 15 If you did not add templates, click **Finish** in the MPLS Link Editor – VRF and VPN window.

The Service Request Editor window appears.

Step 16 To save your work on this first link in the service request, click **Save**.

You return to the Service Requests dialog box, where the information for the link you just defined is now displayed.

You can add additional links to this service request by choosing **Add Link** and specifying the attributes of the next link in the service. As you can see, the service request is in the Requested state. When all the links for this service have been defined, you must deploy the service, as described in Migrating PE Devices from IOS to IOS XR, page 5-98.

Adding a CLE to a Service Request

To add a CLE device to the service request described in Creating a PE-Only Service Request, page 5-95, perform the following steps:

Step 1 Follow Step 1 through Step 5 of Creating a PE-Only Service Request, page 5-95.

Step 2 Click **Select CLE**. The Select PE Device dialog box appears.

a. From the "Show PEs with" drop-down list, you can display PEs by Provider Name, by Region, or by Device Name.

b. You can use the **Find** button to either search for a specific PE, or to refresh the display.

c. You can set the "Rows per page" to **5**, **10**, **20**, **30**, **40**, or **All**.

d. This dialog box displays the first page of the list of currently defined PE devices. The number of pages of information is displayed in the lower right corner of the dialog box.

To go to the another page of PE devices, click the number of the page you want to go to.

Step 3 In the Select column, choose the name of the CLE for the MPLS link, then click **Select**.

You return to the Service Request Editor window, where the name of the selected CLE is now displayed in the CLE column.

Step 4 **CLE Interface:** Choose the CLE interface from the drop-down list.

Step 5 Continue following Step 4 through Step 16 of "Creating a PE-Only Service Request" section on page 5-95.

Migrating PE Devices from IOS to IOS XR

For assistance in migrating services deployed on IOS devices to IOS XR devices, contact Cisco Advanced Services.

Provisioning Regular PE-CE Links

This section describes how to configure MPLS VPN PE-CE links in the Prime Fulfillment provisioning process.

MPLS VPN PE-CE Link Overview

To provision an MPLS VPN service in Prime Fulfillment, you must first create an MPLS VPN Service Policy. In Prime Fulfillment, a Service Policy is a set of default configurations for creating and deploying a service request.

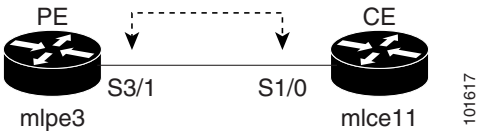
Prime Fulfillment supports two MPLS VPN Service Policy Types: Regular PE-CE and MVRFCE PE-CE. The following scenarios focus on the Regular PE-CE Policy Type.

The Regular PE-CE Policy Type is a normal PE to CE link between two devices. This Policy Type has two options:

- CE Present *enabled* (One PE with one CE; two devices)
- CE Present *disabled* (PE Only with no CE; one device)

Figure 5-9 shows an example of a normal PE to CE link between two devices.

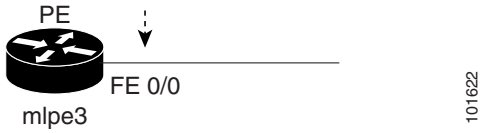
Figure 5-9 PE to CE link with CE Present



In a PE to CE link with CE Present enabled, interfaces S3/1 and S1/0 are configured as an MPLS VPN link in the service request process.

Figure 5-10 shows an example of a PE Only link with no CE.

Figure 5-10 PE to CE link with No CE

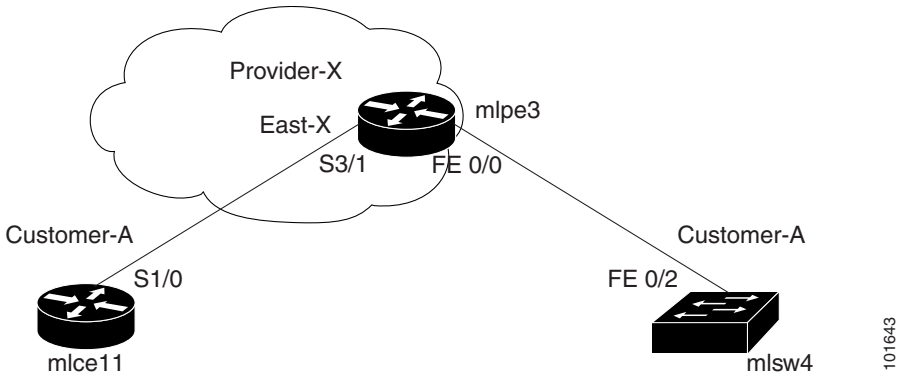


In a PE to CE link with CE Present disabled, interface FE0/0 is configured as an MPLS VPN link in the service request process.

Network Topology

Figure 5-15 shows an overview of the network topology in which the MPLS VPN PE-CE links are created.

Figure 5-11 Network Topology for MPLS VPN PE-CE Scenarios.



The network topology in Figure 5-15 illustrates the lab environment of a service provider (Provider-X) and one customer (Cust-A). There is one Region (East-X) and one PE (mlpe3.cisco.com). Each customer device (one CE and one CLE) represents a Site (mlce11-Site and mlsw4-Site).

Prerequisite Tasks

Before you can create a Service Policy in Prime Fulfillment, you must complete the following Service Inventory tasks:

Step 1 Set up a Customer with a Site (see Managing Customer Premise Devices, page 2-35).

Step 2 Set up a Provider with a Region (see Providers, page 2-15).

Step 3 Import, create, or discover Devices (see Devices, page 2-1).

Step 4 Create CPE and PE (see Providers, page 2-15).

Step 5 Collect Configurations (see Tasks, page 10-23).

Step 6 Create Resource Pools (see Resource Pools, page 2-44).

Step 7 Create Route Target(s) (see Route Targets, page 2-51).

Step 8 Define a MPLS VPN (see Creating an MPLS VPN, page 5-7).

Defining a VPN for the PE-CE Link

During service deployment, Prime Fulfillment generates the Cisco IOS commands to configure the logical VPN relationships. At the beginning of the provisioning process, before creating a Service Policy, a VPN must be defined within Prime Fulfillment.

To define a VPN, perform the following steps:

Step 1 Choose **Inventory > Logical Inventory > VPNs**.

The VPNs window appears.

Step 2 Click **Create** to create a VPN.

The Create New VPN window appears.

Step 3 In the Name field, enter the VPN name.

It is recommended not to use special characters (' ` " < > ( ) [ ] { } / \ & ^ ! ? ~ * % = , . + |) in the VPN name, as this may cause misconfiguration of the VRF name for certain devices, if the VPN name is used to autogenerate a VRF name.

Step 4 In the Customer field, click **Select**.

The Select Customer window appears.

Step 5 Check to choose a Customer and click **Select**.

The VPNs window reappears where the new VPN Name is associated with a Customer in this new VPN definition.

Step 6 Click **Save**.

Note You can also set VRF and VPN attributes via a previously defined independent VRF object. For more information on this feature, see Independent VRF Management, page 5-14

Note For more information on setting the VRF and VPN attributes in MPLS VPN service requests, see Defining VRF and VPN Attributes in an MPLS Service Request, page 5-85.

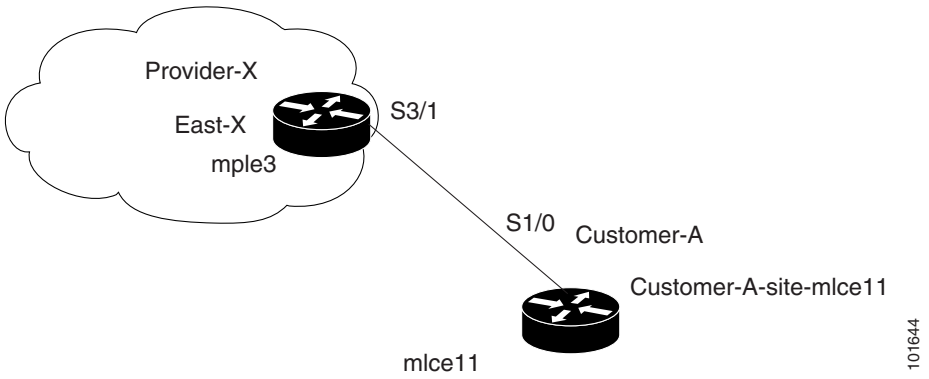
Creating MPLS VPN PE-CE Service Policies

This section contains the following sections:

PE-CE Service Policy Overview

Figure 5-12 shows an example of the PE-CE link that is defined in the PE-CE Service Policy scenario.

Figure 5-12 PE-CE Topology



Creating a PE-CE Service Policy

To create a PE-CE service policy, perform the following steps:

Step 1 Choose **Service Design > Policies > MPLS**.

The MPLS Policy Editor window appears.

Step 2 Edit the following attributes:

- **Policy Name**: Enter the policy name.
- **Policy Owner**: Choose the Policy Owner.
- **Customer**:
  - Click **Select** to specify a Customer.

    The Customer for MPLS Policy window appears.
  - Check to choose a Customer and click **Select**.
- **Policy Type**: Choose the Policy Type. (**Regular PE-CE**)

Step 3 **CE Present**: Check to set CE as present.

Step 4 Click **Next**.

The MPLS Policy Editor - Interface window appears.

Step 5 Click **Next** to accept the defaults.

The MPLS Policy Editor - IP Address Scheme window appears.

Note Make sure the Editable check boxes are checked, so you can edit these attributes in the service request process.

Step 6 Edit all applicable attributes.

Note If you check **Automatically Assign IP Address**, the screen refreshes and adds a forth attribute: **IP Address Pool**.

Step 7 Click **Next**.

The MPLS Policy Editor - Routing Information window appears.

Step 8 Click **Next** to accept the defaults.

The MPLS Policy Editor - VRF and VPN Membership window appears.

Note For information about protocol types, see Specifying the Routing Protocol for a Service, page 5-48.

Note If you want to set the VRF and VPN attributes via a previously defined VRF object, check the **Use VRF Object** check box. For more information on this feature, see Independent VRF Management, page 5-14. That section describes how to use independent VRF objects in MPLS VPN service policies and service requests.

Step 9 To enable template association for the policy, click the **Next** button in MPLS Policy Editor - VRF and VPN Membership window.

Note An additional window appears in the policy workflow before the Template Association window. This window allows you to create user-defined attributes within the policy (and service requests based on the policy). For background information on how to use the additional information feature, see Appendix H, "Adding Additional Information to Services." If you are not using this feature, click **Next** to proceed to the Template Association window, or else click **Finish** to save the policy.

The Template Association window appears. In this window, you can enable template support and, optionally, associate templates and data files with the policy. For instructions about associating templates with policies and how to use the features in this window, see Chapter 9, "Managing Templates and Data Files." When you have completed setting up templates and data files for the policy per the instructions in the appendix, click **Finish** in the Template Association window to close it.

The Policies window appears.

Step 10 If you did not enable templates, click **Finish** in the MPLS Policy Editor – VRF and VPN window.

The Policies window reappears.

The MPLS VPN PE-CE Service Policy is complete.

Creating a PE-NoCE Service Policy

To create a PE-NoCE service policy, perform the following steps:

Step 1 Choose **Service Design > Policies > MPLS**.

The MPLS Policy Editor - Policy Type window appears.

Step 2 Edit the following attributes:

- **Policy Name**: Enter the policy name.
- **Policy Owner**: Choose the Policy Owner.
- **Customer**:

- Click **Select** to specify a Customer.

  The Customer for MPLS Policy window appears.

- Choose a Customer and click **Select.**

- **Policy Type**: Choose the Policy Type. (**Regular PE-CE**)
- **CE Present**: Do ***not*** check to set CE as **not** present (**NoCE**).

Step 3 Click **Next**.

The MPLS Policy Editor - Interface window appears.

Step 4 Click **Next** to accept the defaults.

The MPLS Policy Editor - IP Address Scheme window appears.

Note Make sure the Editable check boxes are checked, so you can edit these attributes in the service request process.

The field values displayed in this dialog box reflect the values specified in the service policy associated with this service.

For details on the IP address scheme fields, see Specifying the IP Address Scheme, page 5-45.

Step 5 Edit all applicable attributes.

Note If you check **Automatically Assign IP Address**, the screen refreshes and adds a forth attribute: **IP Address Pool**.

Step 6 Click **Next**.

The MPLS Policy Editor - Routing Information window appears.

Step 7 Click **Next** to accept the defaults.

The MPLS Policy Editor - VRF and VPN Membership window appears.

Note For information about protocol types, see Specifying the Routing Protocol for a Service, page 5-48.

Note If you want to set the VRF and VPN attributes via a previously defined VRF object, check the **Use VRF Object** check box. For more information on this feature, see Independent VRF Management, page 5-14. That section describes how to use independent VRF objects in MPLS VPN service policies and service requests.

Step 8 To enable template association for the policy, click the **Next** button in MPLS Policy Editor - VRF and VPN Membership window.

Note An additional window appears in the policy workflow before the Template Association window. This window allows you to create user-defined attributes within the policy (and service requests based on the policy). For background information on how to use the additional information feature, see Appendix H, "Adding Additional Information to Services." If you are not using this feature, click **Next** to proceed to the Template Association window, or else click **Finish** to save the policy.

The Template Association window appears. In this window, you can enable template support and, optionally, associate templates and data files with the policy. For instructions about associating templates with policies and how to use the features in this window, see Chapter 9, "Managing Templates and Data Files." When you have completed setting up templates and data files for the policy per the instructions in the appendix, click **Finish** in the Template Association window to close it.

The Policies window appears.

Step 9 If you did not enable templates, click **Finish** in the MPLS Policy Editor – VRF and VPN window.

The Policies window reappears.

The MPLS VPN PE-NoCE Service Policy is complete.

Creating MPLS VPN PE-CE Service Requests

This section contains the following sections:

- Creating MVRFCE PE-CE Service Requests, page 5-115
- Creating MVRFCE PE-NoCE Service Requests, page 5-117

Creating PE-CE Service Requests

To create a PE-CE service request, perform the following steps:

Step 1 Choose **Operate > Service Requests >Service Request Manager**.

The MPLS Policy Selection window appears.

Step 2 Choose an MPLS PE-CE type policy.

Step 3 Click **OK**.

The MPLS Service Request Editor window appears.

Step 4 Click **Add Link**.

The MPLS Service Request Editor window appears.

Step 5 Click **Select CE**.

The CPE for MPLS VPN Link window appears.

Step 6 Choose a CPE device and click **Select**.

The MPLS Service Request Editor window appears.

Step 7 Choose a CE Interface from the drop-down list.

The MPLS Service Request Editor window appears.

Step 8 Click **Select PE**.

The PE for MPLS VPN Link window appears.

Step 9 Choose a PE device and click **Select**.

The MPLS Service Request Editor window appears.

Step 10 Choose a PE Interface from the drop-down list.

The MPLS Service Request Editor window appears.

Step 11 Click **Select PE**.

The PE for MPLS VPN Link window reappears.

Step 12 In the Link Attribute cell, click **Add**.

The MPLS Link Attribute Editor - Interface window appears.

PE Information

Step 13 **Interface Name**: Enter a value to identify the interface.

Step 14 **Interface Description**: Optionally, you can enter a description of the PE interface.

Step 15 **Shutdown Interface**: When you check this check box, the PE interface is configured in a shutdown state.

Step 16 **Encapsulation**: Choose the PE Encapsulation from the drop-down list.

The selections available in the drop-down list are determined by the interface type.

Step 17 **VLAN ID**: Enter the VLAN ID. The VLAN ID is shared between the PE and CE, so there is one VLAN ID for both.

Step 18 **Auto-Pick VLAN ID**: Check this check box if you would like Prime Fulfillment to autopick a VLAN ID from the VLAN pool.

If this box is checked, the VLAN ID field is not visible in the GUI.

Step 19 **Second VLAN ID**: The Second VLAN ID is an optional attribute that provides a method to match the Q-in-Q second VLAN tag of incoming frames on the PE interface.

For usage details about this attribute, see Notes on the VLAN ID and Second VLAN ID Attributes, page 5-83.

Step 20 **Use SVI:** Check this box to have Prime Fulfillment terminate VRF on SVI.

CE Information

Step 21 **Interface Name**: Enter a value from to identify the interface.

Step 22 **Interface Description**: Optionally, you can enter a description of the PE interface.

Step 23 **Encapsulation**: Choose the CE Encapsulation from the drop-down list.

The selections available in the drop-down list are determined by the interface type.

Step 24 Click **Next**.

The MPLS Link Attribute Editor - IP Address Scheme window appears.

Step 25 Accept the defaults and click **Next**.

The MPLS Link Attribute Editor - Routing Information window appears.

Note For information about protocol types, see Specifying the Routing Protocol for a Service, page 5-48.

Step 26 Choose a Next Hop Option:

- USE_OUT_GOING_INTF_NAME
- USE_NEXT_HOP_IPADDR
- OUTGOING_INTF_NAME+NEXT_HOP_IPADDR

Note If this interface is dual stacked (IPv4 and IPv6), you will be prompted to enter the routing information for both IPv4 and IPv6 independently. The fields in the IPv6 Routing Information window are slightly different from the IPv4 version. For information on setting up the routing information for IPv6, see Setting Static Routing Protocol Attributes (for IPv4 and IPv6), page 5-90.

Step 27 To continue, click **Next**.

The MPLS Link Attribute Editor - VRF and VPN window appears.

Note If you want to set the VRF and VPN attributes via a previously defined VRF object, check the **Use VRF Object** check box. For more information on this feature, see Independent VRF Management, page 5-14. That section describes how to use independent VRF objects in MPLS VPN service policies and service requests.

Note For more information on setting the VRF and VPN attributes in MPLS VPN service requests, see Defining VRF and VPN Attributes in an MPLS Service Request, page 5-85.

Step 28 Click **Add** to join a VPN.

The Select CERCs window appears.

Step 29 Choose a Customer from the drop-down list.

Step 30 Choose a VPN from the drop-down list.

Step 31 Check to choose a VPN from the list.

Step 32 Click **Join As Hub** or **Join As Spoke**.

Step 33 Click **Done**.

The MPLS Link Attribute Editor - VRF and VPN window reappears.

Step 34 Click the **Next** button to associate templates or data files to the service request.

The MPLS Link Attribute Editor - Template Association window appears. In this window, you can associate templates and data files with a device by clicking the **Add** button in Template/Data File column for the device. When you click the **Add** button, the Add/Remove Templates window appears. For instructions about associating templates with service requests and how to use the features in this window, see Chapter 9, "Managing Templates and Data Files."

Note The above step assumes the policy on which the service request is based has template association enabled. If not, there will be no **Next** button visible in the GUI. In that case, click **Finish** and return to the MPLS Service Request Editor window and proceed with Step 37, below.

Step 35 When you have completed setting up templates and data files for any device(s), click **Finish** in the Template Association window to close it and return to the MPLS Service Request Editor window.

You can define multiple links in this service request, following the instructions outlined in previous steps.

Step 36 To save your work, click **Save**.

The MPLS Service Requests window reappears showing that the MPLS VPN PE-CE service request is in the Requested state and ready to deploy.

Creating PE-NoCE Service Requests

To create a PE-NoCE service request, perform the following steps:

Step 1 Choose **Operate > Service Requests >Service Request Manager**.

Step 2 Choose an MPLS PE-NoCE type policy.

Step 3 Click **OK**.

The MPLS Service Request Editor window appears.

Step 4 Click **Add Link**.

The MPLS Service Request Editor window appears.

Step 5 Click **Select PE**.

The PE for MPLS VPN Link window appears.

Step 6 Choose a PE device and click **Select**.

The MPLS Service Request Editor window appears.

Step 7 Choose the PE Interface from the drop-down list.

The MPLS Service Request Editor window appears.

Step 8 In the Link Attribute cell, Click **Add**.

The MPLS Link Attribute Editor - Interface window appears.

Step 9 **Interface Name**: Enter a value to identify the interface.

Step 10 **Interface Description**: Optionally, you can enter a description of the PE interface.

Step 11 **Shutdown Interface**: When you check this check box, the PE interface is configured in a shutdown state.

Step 12 **PE Encapsulation**: Choose the PE Encapsulation from the drop-down list.

The selections available in the drop-down list are determined by the interface type. This field is needed for deciding PE/UNI encapsulation.

Step 13 **VLAN ID**: Enter the VLAN ID. The VLAN ID is shared between the PE and CE, so there is one VLAN ID for both.

Step 14 **Auto-Pick VLAN ID**: Check this check box if you would like Prime Fulfillment to autopick a VLAN ID from the VLAN pool.

If this box is checked, the VLAN ID field is not visible in the GUI.

Step 15 **Second VLAN ID**: The Second VLAN ID is an optional attribute that provides a method to match the Q-in-Q second VLAN tag of incoming frames on the PE interface.

For usage details about this attribute, see Notes on the VLAN ID and Second VLAN ID Attributes, page 5-83.

Step 16 **Use SVI:** Check this box to have Prime Fulfillment terminate VRF on SVI.

Step 17 **Standard UNI Port:** Check this box to access additional UNI security parameters.

Step 18 Click **Next**.

The MPLS Link Attribute Editor - IP Address Scheme window appears.

Step 19 Accept the defaults and click **Next**.

Note If this interface is dual stacked (IPv4 and IPv6), you will be prompted to enter the routing information for both IPv4 and IPv6 independently.

The MPLS Link Attribute Editor - Routing Information window appears.

Step 20 Set attributes for the routing information as needed for your configuration.

Note For information about protocol types, see Specifying the Routing Protocol for a Service, page 5-48.

Step 21 Click **Next**.

The MPLS Link Attribute Editor - VRF and VPN window appears.

Note If you want to set the VRF and VPN attributes via a previously defined VRF object, check the **Use VRF Object** check box. For more information on this feature, see Independent VRF Management, page 5-14. That section describes how to use independent VRF objects in MPLS VPN service policies and service requests.

Note For more information on setting the VRF and VPN attributes in MPLS VPN service requests, see Defining VRF and VPN Attributes in an MPLS Service Request, page 5-85.

Step 22 Click **Add** to join the VPN.

The Join VPN dialog box appears.

Step 23 Check to choose the VPN.

Step 24 Click Join as Hub or Join as Spoke.

Step 25 Click **Done**.

The MPLS Service Request Editor window reappears.

Step 26 Click the **Next** button to associate templates or data files to the service request.

The MPLS Link Attribute Editor - Template Association window appears. In this window, you can associate templates and data files with a device by clicking the **Add** button in Template/Data File column for the device. When you click the **Add** button, the Add/Remove Templates window appears. For instructions about associating templates with service requests and how to use the features in this window, see Chapter 9, “Managing Templates and Data Files.”

Note The above step assumes the policy on which the service request is based has template association enabled. If not, there will be no **Next** button visible in the GUI. In that case, click **Finish** and return to the MPLS Service Request Editor window and proceed with Step 30, below.

Step 27 When you have completed setting up templates and data files for any device(s), click **Finish** in the Template Association window to close it and return to the MPLS Service Request Editor window.

You can define multiple links in this service request, following the instructions outlined in previous steps.

Step 28 To save your work, click **Save**.

The MPLS Service Requests window reappears showing that the MPLS VPN PE-NoCE Service Request is in the Requested state and ready to deploy.

Provisioning Multi-VRFCE PE-CE Links

This section describes how to configure MPLS VPN Multi-VRFCE PE-CE links in the Prime Fulfillment provisioning process.

MPLS VPN MVRFCE PE-CE Link Overview

This section contains the following sections:

To provision an MPLS VPN service in Prime Fulfillment, you must first create an MPLS VPN Service Policy. In Prime Fulfillment, a Service Policy is a set of default configurations for creating and deploying a service request. Prime Fulfillment supports two MPLS VPN Service Policy Types: Regular PE-CE an MVRFCE PE-CE. The following scenarios focus on the MVRFCE PE-CE Policy Type. An MVRFCE PE-CE Policy Type is a PE to CE link with three devices:

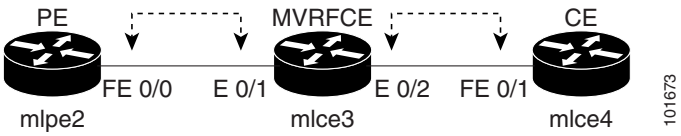
- PE
- Multi-VRF CE
- CE

This Policy Type has two options:

- CE Present *enabled* (One PE with one MVRFCE and one CE; three devices)
- CE Present *disabled* (One PE with one MVRFCE; two devices)

Figure 5-13 shows an example of an MVRFCE PE-CE link with three devices.

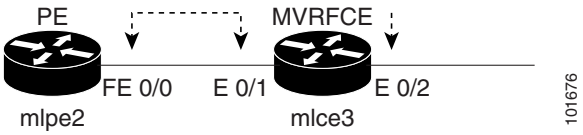
Figure 5-13 MVRFCE PE-CE Link



In an MVRFCE PE-CE link with CE Present enabled, interfaces FE 0/0, E 0/1, E 0/2 and FE 0/1 are configured as an MPLS VPN link in the service request process.

Figure 5-14 shows an example of a PE to MVRFCE link with no CE.

Figure 5-14 MVRFCE PE-CE Link with No CE

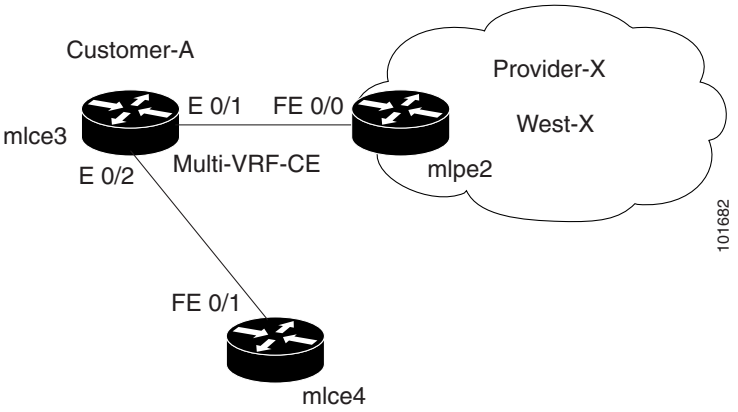


In an MVRFCE PE-CE link with CE Present disabled, interfaces FE 0/0, E 0/1, and E 0/2 are configured as an MPLS VPN link in the service request process.

Network Topology

Figure 5-15 shows an overview of the network topology in which the MPLS VPN MVRFCE PE-CE links are created.

Figure 5-15 Network Topology for MPLS VPN MVRFCE PE-CE Scenarios



The network topology in Figure 5-15 illustrates the lab environment of a service provider (Provider-X) and one customer (Cust-A). There is one Region (West-X) and one PE (mlpe2.cisco.com). Each customer device (one MVRFCE and one CE) represents a Site (mlce3-Site and mlce4-Site).

Prerequisite Tasks

Before you can create a Service Policy in Prime Fulfillment, you must complete the following Inventory Management tasks:

Step 1 Set up a Customer with a Site (see Managing Customer Premise Devices, page 2-35).

Step 2 Setup a Provider with a Region (see Providers, page 2-15).

Step 3 Import, create, or discover Devices (see Chapter 2, "Devices").

Step 4 Create CPE and PE (see Providers, page 2-15).

Step 5 Collect Configurations (see Tasks, page 10-23).

Step 6 Create Resource Pools (see Resource Pools, page 2-44).

Step 7 Create CE routing communities (CERC) (see Route Targets, page 2-51).

Step 8 Define a MPLS VPN (see Creating an MPLS VPN, page 5-7).

Defining VPN for MVRFCE PE-CE Links

During service deployment, Prime Fulfillment generates the Cisco IOS commands to configure the logical VPN relationships.

At the beginning of the provisioning process, before creating a Service Policy, a VPN must be defined within Prime Fulfillment. The first element in a VPN definition is the name of the VPN.

To create a VPN Name, perform the following steps:

Step 1 Choose **Inventory > Logical Inventory > VPNs**.

The VPNs window appears.

Step 2 Click **Create** to create a VPN.

The Create New VPN window appears.

Step 3 Edit the following attributes:

- **Name**: Enter the VPN name.

  It is recommended not to use special characters (' ` " < > ( ) [ ] { } / \ & ^ ! ? ~ * % = , . + |) in the VPN name, as this may cause misconfiguration of the VRF name for certain devices, if the VPN name is used to autogenerate a VRF name.

- **Customer**: Click **Select**.

The Select Customer window appears.

Step 4 Choose a Customer and click **Select**.

Step 5 Click **Save**.

Note Independent VRF association is not supported for MVRFCE-based policies and service requests.

Creating MPLS VPN MVRFCE PE-CE Service Policies

This section contains the following sections:

- Creating MVRFCE PE-CE Service Policies, page 5-112

- Creating PE-NoCE Service Policies, page 5-113

Creating MVRFCE PE-CE Service Policies

To create an MVRFCE PE-CE service policy, perform the following steps:

Note Make sure the Editable check boxes are checked where available, so you can edit these attributes in the service request process.

Step 1 Choose **Service Design > Policies > MPLS**.

The MPLS Policy Editor - Policy Type window appears.

Step 2 Edit the following attributes:

- **Policy Name**: Enter the policy name.
- **Policy Owner**: Choose the Policy Owner.
- **Customer**:
  - Click **Select** to specify a customer.

    The Customer for MPLS Policy window appears.
  - Choose a customer and click **Select.**
- **Policy Type**: Choose the Policy Type. (**MVRFCE: PE-CE**)
- **CE Present**: Check to set CE as present.

Step 3 Click **Next**.

The MPLS Policy Editor - PE Interface window appears.

Step 4 Click **Next**.

The MPLS Policy Editor - Interface window appears.

Step 5 Edit all applicable attributes.

Step 6 Click **Next**.

The MPLS Policy Editor - IP Address Scheme window appears for **PE-MVRFCE**.

Step 7 Edit all applicable attributes.

Step 8 Click **Next**.

Step 9 Another set of MPLS Policy Editor - IP Address Scheme windows appear for **MVRFCE-CE**.

Step 10 Edit all applicable attributes, as above.

Step 11 Click **Next**.

The MPLS Policy Editor - Routing Information window appears for **PE-MVRFCE**.

Note For information about protocol types, see Specifying the Routing Protocol for a Service, page 5-48.

Step 12 Click **Next** to accept the defaults.

The MPLS Policy Editor - Routing Information window appears for **MVRFCE-CE**.

Step 13 Click **Next** to accept the defaults.

The MPLS Policy Editor - VRF and VPN Membership window appears.

Step 14 To enable template association for the policy, click the **Next** button in MPLS Policy Editor - VRF and VPN Membership window.

Note An additional window appears in the policy workflow before the Template Association window. This window allows you to create user-defined attributes within the policy (and service requests based on the policy). For background information on how to use the additional information feature, see Appendix H, "Adding Additional Information to Services." If you are not using this feature, click **Next** to proceed to the Template Association window, or else click **Finish** to save the policy.

The Template Association window appears. In this window, you can enable template support and, optionally, associate templates and data files with the policy. For instructions about associating templates with policies and how to use the features in this window, see Chapter 9, "Managing Templates and Data Files." When you have completed setting up templates and data files for the policy per the instructions in the appendix, click **Finish** in the Template Association window to close it.

The Policies window appears.

Step 15 If you did not enable templates, click **Finish** in the MPLS Policy Editor – VRF and VPN window.

The Policies window reappears showing that the MPLS VPN MVRFCE PE-CE Service Policy is complete.

Creating PE-NoCE Service Policies

To create a PE-NoCE service policy, perform the following steps:

Step 1 Choose **Service Design > Policies** > MPLS.

The MPLS Policy Editor - Policy Type window appears.

Step 2 Edit the following attributes:

- **Policy Name**: Enter the policy name.
- **Policy Owner**: Choose the Policy Owner.
- **Customer**:
  - Click **Select** to specify a customer.

    The Customer for MPLS Policy window appears.
  - Choose a customer and click **Select.**
- **Policy Type**: Choose the Policy Type. (**Regular PE-CE**)
- **CE Present**: Do ***not*** check to set CE as **not** present (**NoCE**).

Step 3 Click **Next**.

The MPLS Policy Editor - Interface window appears.

Step 4 Click **Next** to accept the defaults.

The MPLS Policy Editor - Interface window appears for **MVRFCE-CE Facing Information**.

Step 5 Click **Next** to accept the defaults.

The MPLS Policy Editor - IP Address Scheme window appears for **PE-MVRFCE-CE Interface Address/Mask**.

a. Edit the attributes as indicated:

b. **IP Numbering Scheme**: Choose **IP Numbered** Scheme.

c. **Automatically Assign IP Address**: To have Prime Fulfillment automatically assign IP Addresses, check the check box.

d. **IP Address Pool**: Choose the IP Address Pool.

Step 6 Click **Next.**

The MPLS Policy Editor - IP Address Scheme window appears for **MVRFCE-CE Interface Address/Mask**.

a. Edit the attributes as indicated:

b. **IP Numbering Scheme**: Choose **IP Numbered** Scheme.

c. **Automatically Assign IP Address**: To have Prime Fulfillment automatically assign IP Addresses, check the check box.

d. **IP Address Pool**: Choose the IP Address Pool.

Step 7 Click **Next**.

The MPLS Policy Editor - Routing Information window appears for **PE-MVRFCE Routing Information**.

Note For information about protocol types, see Specifying the Routing Protocol for a Service, page 5-48.

Step 8 Click **Next** to accept the defaults.

The MPLS Policy Editor - Routing Information window appears for **MVRFCE-CE Routing Information**.

Step 9 Click **Next** to accept the defaults.

The MPLS Policy Editor - VRF and VPN Membership window appears.

Step 10 Click **Add** to join a VPN. The VPN dialog box appears.

Step 11 Click **Join as Hub**, then click **Done**.

The MPLS Policy Editor - VRF and VPN Membership window appears.

Step 12 To enable template association for the policy, click the **Next** button in MPLS Policy Editor - VRF and VPN Membership window.

Note An additional window appears in the policy workflow before the Template Association window. This window allows you to create user-defined attributes within the policy (and service requests based on the policy). For background information on how to use the additional information feature, see Appendix H, "Adding Additional Information to Services." If you are not using this feature, click **Next** to proceed to the Template Association window, or else click **Finish** to save the policy.

The Template Association window appears. In this window, you can enable template support and, optionally, associate templates and data files with the policy. For instructions about associating templates with policies and how to use the features in this window, see Chapter 9, "Managing Templates and Data Files." When you have completed setting up templates and data files for the policy per the instructions in the appendix, click **Finish** in the Template Association window to close it.

The Policies window appears.

Step 13 If you did not enable templates, click **Finish** in the MPLS Policy Editor – VRF and VPN window.

The Policies window reappears showing that the MPLS VPN MVRFCE PE-NoCE Service Policy is complete.

Creating MPLS VPN MVRFCE PE-CE Service Requests

This section contains the following sections:

Creating MVRFCE PE-CE Service Requests

To create an MVRFCE PE-CE service request, perform the following steps:

Step 1 Choose **Operate > Service Requests >Service Request Manager**.

Step 2 Choose the MPLS Policy (**mpls-mvrfce-pe-ce**).

Step 3 Click **OK**.

The MPLS Service Request Editor window appears.

Step 4 Click **Add Link**.

The MPLS Service Request Editor window appears.

Step 5 Click **Select CE**.

The CPE for MPLS VPN Link window appears.

Step 6 Choose the CPE Device and click **Select**.

The MPLS Service Request Editor window appears.

Step 7 Choose the CE Interface from the drop-down list.

Step 8 Click **Select MVRFCE**.

The MVRFCE for MPLS VPN Link window appears.

Step 9 Choose the MVRFCE and click **Select**.

The MPLS Service Request Editor window appears.

Step 10 Choose the **MVRFCE PE Facing Interface** from the drop-down list.

Step 11 Click **Add** in the Link Attribute cell.

The MPLS Link Attribute Editor - Interface window appears.

PE Information

Step 12 **Encapsulation**: Choose the PE Encapsulation from the drop-down list. (**DOT1Q**)

Step 13 **VLAN ID:** Enter the PE VLAN ID.

MVRFCE PE Facing Information

Step 14 **Encapsulation**: Choose the PE Encapsulation from the drop-down list. (**DOT1Q**))

Step 15 Click **Next**.

The MPLS Link Attribute Editor - Interface window appears.

MVRFCE CE Information

Step 16 **Encapsulation**: Choose the PE Encapsulation from the drop-down list. (**DOT1Q**)

Step 17 **VLAN ID:** Enter the PE VLAN ID.

MVRFCE PE-Facing Information

Step 18 **Encapsulation**: Choose the PE Encapsulation from the drop-down list. (**DOT1Q**)

Step 19 Click **Next**.

The MPLS Link Attribute Editor - IP Address Scheme window appears for **PE-MVRF-CE interface address/mask**.

Step 20 Accept the defaults and click **Next**.

The MPLS Link Attribute Editor - IP Address Scheme window appears for **MVRFCE-CE interface address/mask**.

Step 21 Accept the defaults and click **Next**.

The MPLS Link Attribute Editor - Routing Information window reappears for **PE-MVRF-CE routing information**.

Note For information about protocol types, see Specifying the Routing Protocol for a Service, page 5-48.

Step 22 Accept the defaults and click **Next**.

The MPLS Link Attribute Editor - Routing Information window reappears for **MVRFCE-CE routing information**.

Step 23 Accept the defaults and click **Next**.

The MPLS Link Attribute Editor - VRF and VPN window appears.

Note For more information on setting the VRF and VPN attributes in MPLS VPN service requests, see Defining VRF and VPN Attributes in an MPLS Service Request, page 5-85.

Step 24 Click **Add** to join a VPN.

The Select CERCs window appears.

Step 25 Choose a Customer from the drop-down list.

Step 26 Choose a VPN from the drop-down list.

Step 27 Check to choose a VPN from the list.

Step 28 Click **Join As Hub** or **Join As Spoke**.

Step 29 Click **Done**.

The MPLS Link Attribute Editor - VRF and VPN window reappears.

Step 30 Click the **Next** button to associate templates or data files to the service request.

The MPLS Link Attribute Editor - Template Association window appears. In this window, you can associate templates and data files with a device by clicking the **Add** button in Template/Data File column for the device. When you click the **Add** button, the Add/Remove Templates window appears. For instructions about associating templates with service requests and how to use the features in this window, see Chapter 9, "Managing Templates and Data Files."

Note The above step assumes the policy on which the service request is based has template association enabled. If not, there will be no **Next** button visible in the GUI. In that case, click **Finish** and return to the MPLS Service Request Editor window and proceed with Step 34, below.

Step 31 When you have completed setting up templates and data files for any device(s), click **Finish** in the Template Association window to close it and return to the MPLS Service Request Editor window.

The MPLS Service Request Editor window reappears.

Step 32 Enter the service request description (**mpls-mvrfce-pe-ce**) and click **Save**.

The MPLS Service Requests window reappears showing that the MPLS VPN MVRFCE PE-CE service request is in the Requested state and ready to deploy.

Creating MVRFCE PE-NoCE Service Requests

To create an MVRFCE PE-NoCE service request, perform the following steps:

Step 1 Choose **Operate > Service Requests >Service Request Manager**.

Step 2 Choose the MPLS Policy (**mpls-mvrfce-pe-noce**).

Step 3 Click **OK**.

The MPLS Service Request Editor window appears.

Step 4 Click **Add Link**.

The MPLS Service Request Editor window appears.

Step 5 Click **Select MVRFCE**.

The CPE for MPLS VPN Link window appears.

Step 6 Choose a MVRFCE and click **Select**.

The MPLS Service Request Editor window appears.

Step 7 Choose the **MVRFCE CE Facing Interface** from the drop-down list.

Step 8 Click **Add** in the Link Attribute cell.

The MPLS Link Attribute Editor - Interface window appears.

PE Information

Step 9 **Encapsulation**: Choose the PE Encapsulation from the drop-down list. (**DOT1Q**)

Step 10 **VLAN ID:** Enter the PE VLAN ID.

MVRFCE PE Facing Information

Step 11 **Encapsulation**: Choose the PE Encapsulation from the drop-down list. (**DOT1Q**))

Step 12 Click **Next**.

The MPLS Link Attribute Editor - Interface window appears.

MVRFCE CE Information

Step 13 **Encapsulation**: Choose the PE Encapsulation from the drop-down list. (**DOT1Q**)

Step 14 **VLAN ID:** Enter the PE VLAN ID.

MVRFCE PE Facing Information

Step 15 **Encapsulation**: Choose the PE Encapsulation from the drop-down list. (**DOT1Q**)

Step 16 Click **Next**.

The MPLS Link Attribute Editor - IP Address Scheme window appears for **PE-MVRF-CE interface address/mask**.

Step 17 Click **Next** to accept the defaults.

The MPLS Link Attribute Editor - IP Address Scheme window appears for **MVRFCE-CE interface address/mask**.

Step 18 Click **Next** to accept the defaults.

The MPLS Link Attribute Editor - Routing Information window reappears for **PE-MVRF-CE routing information**.

Note For information about protocol types, see Specifying the Routing Protocol for a Service, page 5-48.

Step 19 Click **Next** to accept the defaults.

The MPLS Link Attribute Editor - Routing Information window reappears for **MVRFCE-CE routing information**.

Step 20 Click **Next** to accept the defaults.

The MPLS Link Attribute Editor - VRF and VPN window appears.

Note For more information on setting the VRF and VPN attributes in MPLS VPN service requests, see Defining VRF and VPN Attributes in an MPLS Service Request, page 5-85.

Step 21 Click **Add** to join a VPN.

The Select CERCs window appears.

Step 22 Choose a Customer from the drop-down list.

Step 23 Choose a VPN from the drop-down list.

Step 24 Check to choose a VPN from the list.

Step 25 Click **Join As Hub** or **Join As Spoke**.

Step 26 Click **Done**.

The MPLS Link Attribute Editor - VRF and VPN window reappears.

Note For more information on setting the VRF and VPN attributes in MPLS VPN service requests, see Defining VRF and VPN Attributes in an MPLS Service Request, page 5-85.

Step 27 Click the **Next** button to associate templates or data files to the service request.

The MPLS Link Attribute Editor - Template Association window appears. In this window, you can associate templates and data files with a device by clicking the **Add** button in Template/Data File column for the device. When you click the **Add** button, the Add/Remove Templates window appears. For instructions about associating templates with service requests and how to use the features in this window, see Chapter 9, "Managing Templates and Data Files."

Note The above step assumes the policy on which the service request is based has template association enabled. If not, there will be no **Next** button visible in the GUI. In that case, click **Finish** and return to the MPLS Service Request Editor window and proceed with Step 34, below.

Step 28 When you have completed setting up templates and data files for any device(s), click **Finish** in the Template Association window to close it and return to the MPLS Service Request Editor window.

The MPLS Service Request Editor window reappears.

Step 29 Enter the service request description and click **Save**. (**mpls-mvrfce-pe-noce**)

The MPLS Service Requests window reappears showing that the MPLS VPN MVRFCE PE-NoCE service request is in the Requested state and ready to deploy.

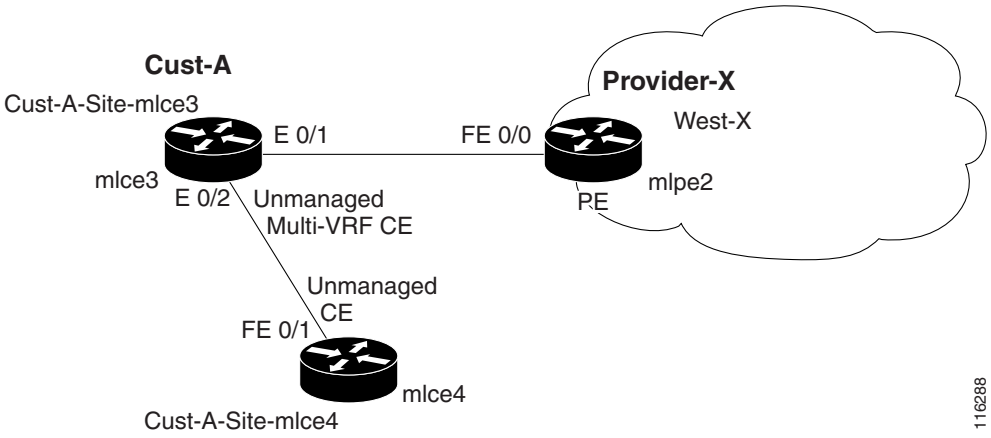
Creating an Unmanaged MVRFCE

The unmanaged MVRFCE feature is similar to the unmanaged CE feature in so far as the service provider does not use Prime Fulfillment to upload or download configurations to the CPE. This feature is similar to the managed MVRFCE feature in so far as Prime Fulfillment creates a link with three devices: a PE, an MVRFCE, and a CE.

In the unmanaged scenarios, the customer configures the CPE manually. To automate the process of configuring the unmanaged MVRFCE, the service provider can use Prime Fulfillment to generate the configuration and then send it to the customer for manual implementation.

Figure 5-16 shows an overview of a network topology with MPLS VPN MVRFCE PE-CE links.

Figure 5-16 Unmanaged MVRFCE PE-CE Network Topology



The network topology in Figure 5-16 shows a service provider (**Provider-X**) and a customer (**Cust-A**). The Provider contains one Region (**West-X**) and one PE (**mlpe2**). The Customer contains an MVRFCE (**mlce3**) and a CE (**mlce4**). Both of these CPEs are unmanaged.

Provisioning Management VPN

This section provides the fundamental concepts and considerations for administering customer edge routers (CEs) in the context of an Prime Fulfillment management subnet. Before Prime Fulfillment can be appropriately deployed to deliver services to customers, the question of whether the CEs are to be managed by the Service Provider or not must be answered.

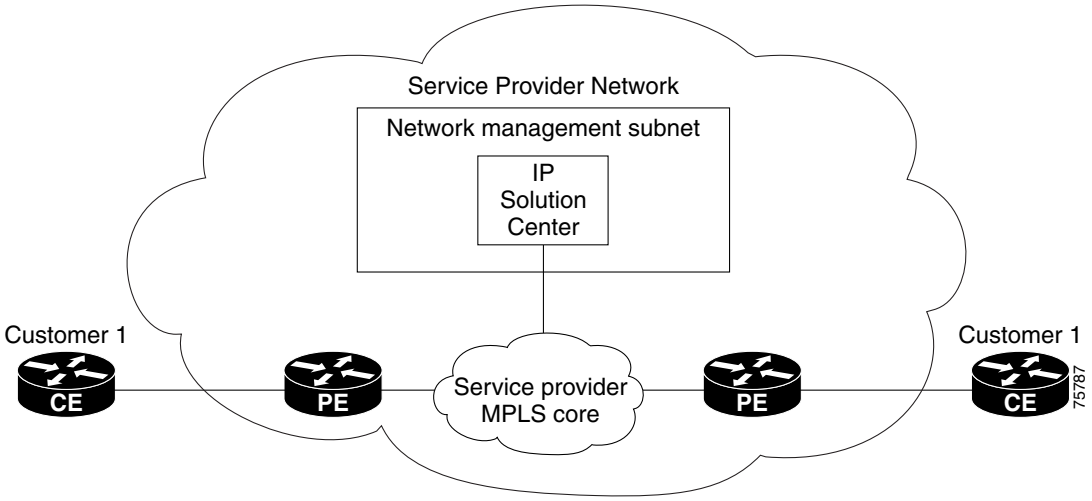
Unmanaged Customer Edge Routers

One of the options available to the Service Provider is to not manage the customer edge routers (CEs) connected to the Service Provider network. For the Service Provider, the primary advantage of an unmanaged CE is administrative simplicity.

If the CEs are unmanaged, the provider can use IPv4 connectivity for all management traffic. Prime Fulfillment is not employed for provisioning or managing unmanaged CEs.

Figure 5-17 shows a basic topology with unmanaged CEs. The network management subnet has a direct link to the Service Provider MPLS core network.

Figure 5-17 Service Provider Network and Unmanaged CEs



Regarding unmanaged CEs, Service Providers should note the following considerations:

- Because unmanaged CEs are outside the Service Provider's administrative domain, the Service Provider does not maintain or configure unmanaged CEs.
- The Service Provider does *not* administer the following elements on the unmanaged CE:
  - IP addresses
  - Host Name
  - Domain Name server
  - Fault management (and timestamp coordination by means of the Network Time Protocol)
  - Collecting, archiving, and restoring CE configurations
  - Access data such as passwords and SNMP strings on the unmanaged CE
- Prototype CE configlets are generated, but they are not automatically downloaded to the router.
- There is no configuration management.
  - With no configuration management, no configuration history is maintained and there is no configuration change management.
  - Changes to a service request (on the PE-CE link) are not deployed to the CE.
- There is no configuration auditing because there is no means to retrieve the current CE configuration.
- You can perform routing auditing.
- You can use the Service Assurance Agent (SA Agent) to measure response times between shadow routers, but you *cannot* use SA Agent to measure response times between CEs.

Managed Customer Edge Routers

The alternative to unmanaged CEs is managed CEs, that is, customer edge routers managed by the Service Provider. Managed CEs can be wholly within the Service Provider's administrative domain or co-managed between the provider and the customer, although CE co-management poses a number of ongoing administrative challenges and is not recommended.

Regarding managed CEs, Service Providers should note the following considerations:

- Managed CEs are within the Service Provider's administrative domain. Thus, some connectivity to the CEs from the Service Provider network is required.
- The Service Provider must administer the following elements on the managed CE:
  - IP addresses
  - Host Name
  - Domain Name server
  - Access data such as passwords and SNMP strings
- The Service Provider should administer fault management (and timestamp coordination by means of the Network Time Protocol)
- The Service Provider can administer collecting, archiving, and restoring CE configurations.
- CE configlets are generated and downloaded to the managed CE.
- Changes to service requests are based on the current CE configuration and automatically downloaded.
- The CE configurations are audited.
- Customer routing and Service Provider routing must interact.
- Access from CEs to the management hosts on the network management subnet is required.
- Configuration auditing and routing auditing are both functional.
- You can use the Service Assurance Agent (SA Agent) to measure response times between CEs and between shadow routers.

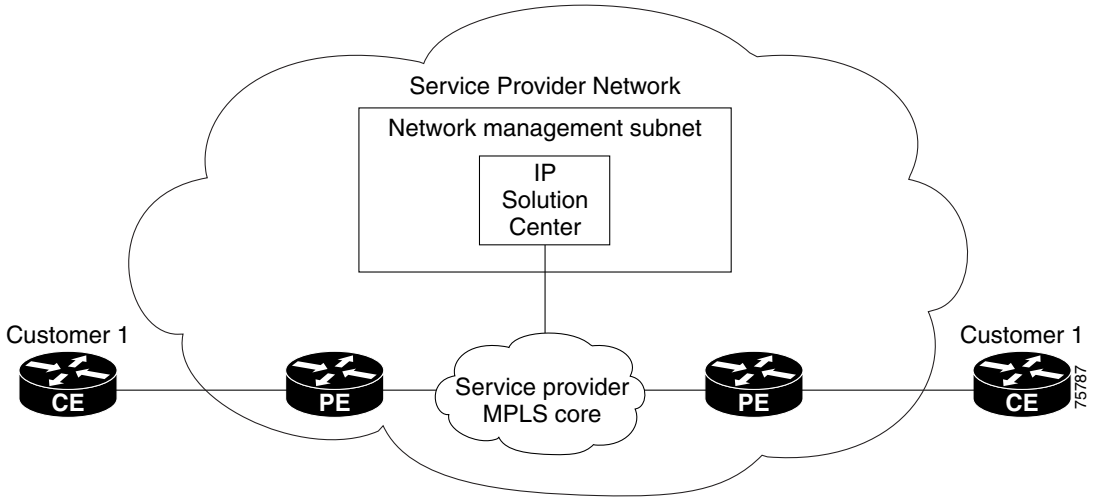
The following sections discuss the concepts and issues required for administering a managed CE environment.

Network Management Subnets

The Network Management Subnet is required when the provider's service offering entails the management of CEs. Once a CE is in a VPN, it is no longer accessible by means of conventional IPv4 routing unless one of the techniques described in this section is employed.

Figure 5-18 shows the Prime Fulfillment network management subnet and the devices that might be required to connect to it:

Figure 5-18 The Prime Fulfillment Network Management Subnet



Issues Regarding Access to VPNs

The core issues with regard to gaining access to VPNs are as follows:

- How to keep provider space “clean” from unnecessary customer routes
- How to keep customer space “clean” from both the provider’s and other customer’s routes
- How to provide effective security
- How to prevent routing loops

Note Prime Fulfillment does not handle any of these responsibilities—doing so must be designed and implemented by the Service Provider.

- Reachability changes as a direct consequence of employing Prime Fulfillment.

  Before you provision a CE in the Prime Fulfillment, you might be able to reach the CE via IPv4 connectivity, but the moment the product deploys a service request, you cannot reach that CE any more—unless you have *first* implemented the network management subnet.

Implementation Techniques

The network management subnet must have access to a Management CE (MCE) and PEs. The network management subnet is appropriate—and necessary—when there is an intent to have managed CEs connected via an in-band connection. *In-band* indicates a single link or permanent virtual circuit (PVC) that carries *both* the customer's VPN traffic, as well as the provider’s network management traffic.

Management CE (MCE)

The network management subnet is connected to the Management CE (MCE). The MCE emulates the role of a customer edge router (CE), but the MCE is in provider space and serves as a network operations center gateway router. The MCE is part of a management site as defined in the Prime Fulfillment. You configure the MCE by identifying the CE as part of the management LAN in Prime Fulfillment.

Management PE (MPE)

The Management PE (MPE) emulates the role of a PE in the provider core network. The MPE connects the MCE to the provider core network. An MPE can have a dual role as both a PE and the MPE.

The MPE needs access to the following devices:

Device	Connectivity	Function
1. Customer Edge Routers (CEs)	Access from the network management subnet into the VPNs	Provision or change configuration and collect SA Agent performance data.
2. Shadow CEs	Access from the network management subnet into the VPNs	A simulated CE used to measure data travel time between two devices. A shadow CE is connected directly to a PE via Ethernet.
3. Provider Edge Routers (PEs)	Standard IP connectivity	Provision or change configuration.

At the current time, Prime Fulfillment recommends two main network management subnet implementation techniques:

- Management VPN Technique

  The MPE-MCE link uses a Management VPN (see Management VPN, page 5-124) to connect to managed CEs. To connect to the PEs, the MPE-MCE link uses a parallel IPv4 link.

- Out-of-Band Technique

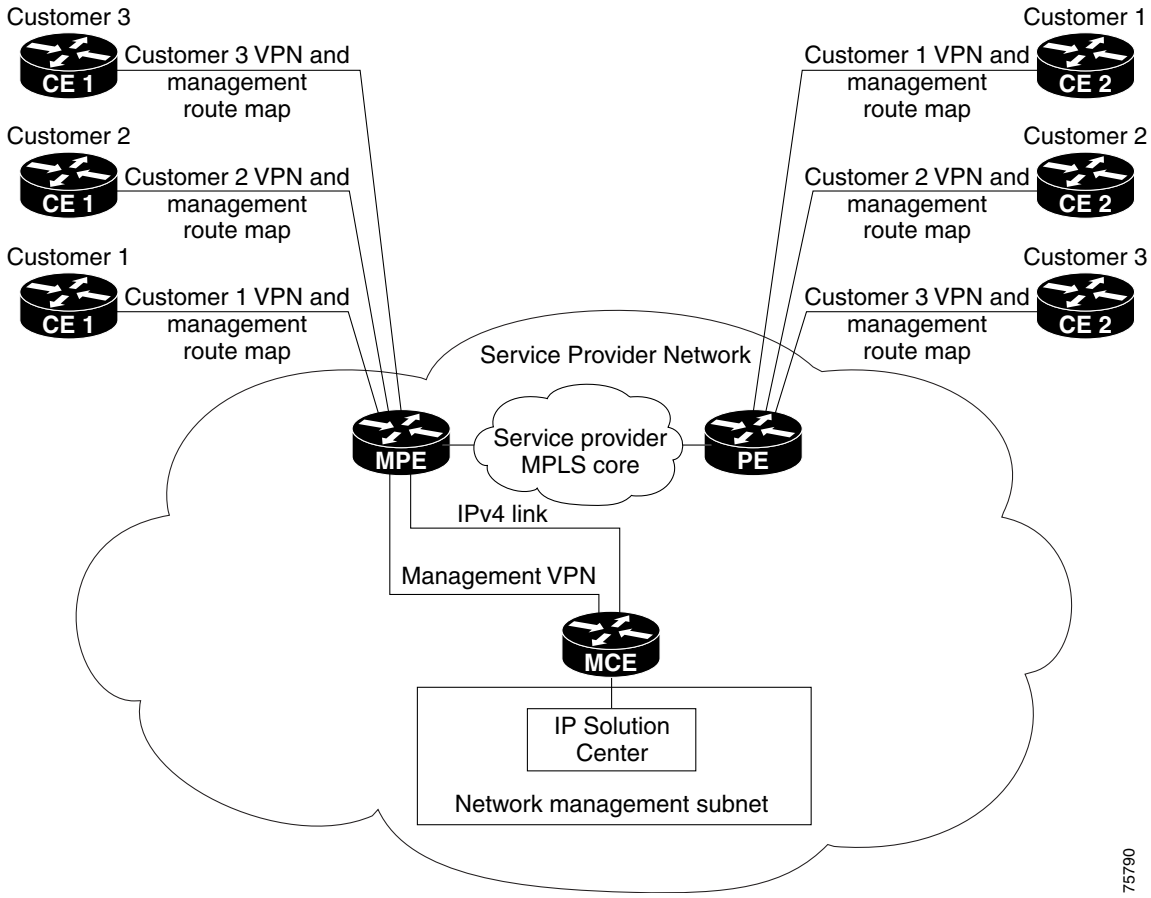
  In the Out-of-Band technique, the MCE has IPv4 connectivity (that is, not MPLS VPN connectivity) to all the CEs and PEs in the network (see Out-of-Band Technique, page 5-126). In this context, *out-of-band* signifies a separate link between PEs that carries the provider's management traffic.

The network management subnet technique the provider chooses to implement depends on many factors, which are discussed later in this section.

Management VPN

The Management VPN technique is the default method provisioned by Prime Fulfillment. A key concept for this implementation technique is that all the CEs in the network are a member of the management VPN. To connect to the PEs, the MPE-MCE link uses a parallel IPv4 link. Figure 5-19 shows a typical topology for the Management VPN technique.

Figure 5-19 Typical Topology for a Management VPN Network



When employing the Management VPN technique, the MPE-MCE link uses a management VPN to connect to managed CEs. To connect to the PEs, the MPE-MCE link employs a parallel IPv4 link.

Each CE in a customer VPN is also added to the management VPN by selecting the Join the management VPN option in the service request user interface.

The function of the management route map is to allow only the routes to the specific CE into the management VPN. The Cisco IOS supports only one export route map and one import route map per VRF.

As shown in Figure 5-19, a second parallel non-MPLS VPN link is required between the MPE and MCE to reach the PEs.

Note Implementation of the Management VPN technique requires Cisco IOS 12.07 or higher.

The advantages involved in implementing the Management VPN technique are as follows:

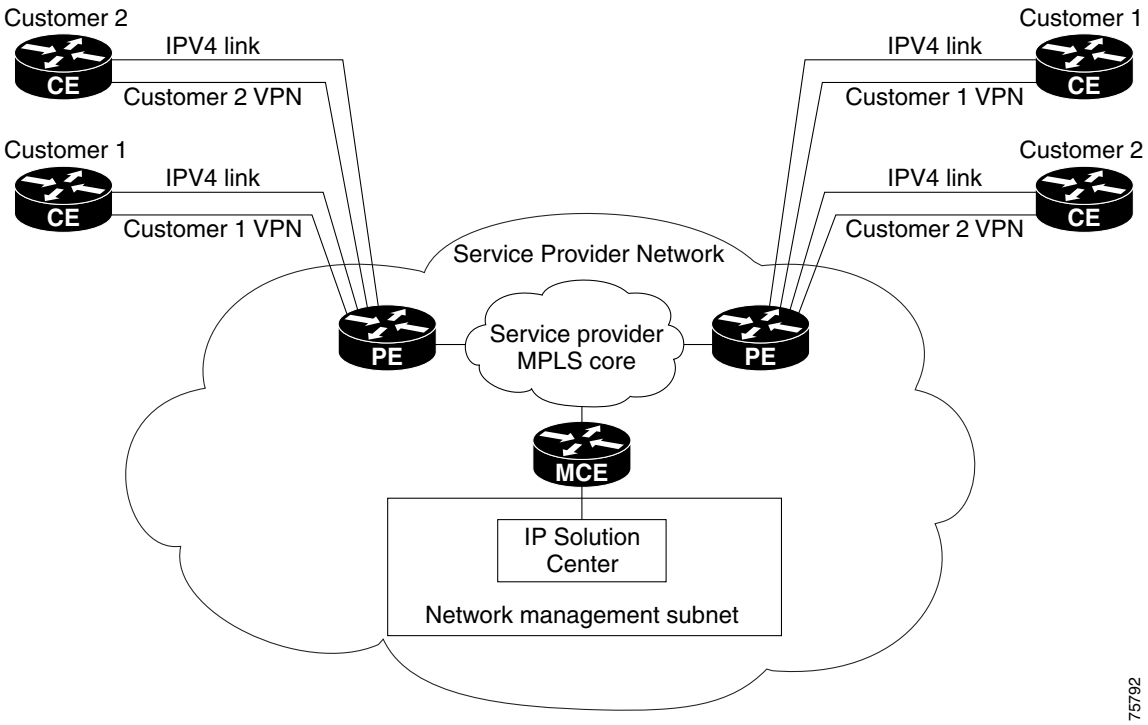
- Provisioning with this method requires only one service request.
- The only routes given to the network management subnet are the routes to the CEs—that is, either the address of the CE link to the PE or the CE loopback address. General VPN routes are *not* given to the network management subnet.

- A CE in the Management VPN method is a spoke to the Management VPN regardless of which role the CE has within its own VPN. Therefore, CEs cannot be accidentally exposed to inappropriate routes. The only management routes the CEs can learn must come from a hub of the Management VPN.

Out-of-Band Technique

The Out-of-Band technique does not employ a management VPN to manage the CEs. Out-of-band connectivity is provided by IPv4 links. *Out-of-band* signifies a separate link between PEs that carries the provider's management traffic. As shown in Figure 5-20, the MCE provides separation between the provider's routes and the customer's routes.

Figure 5-20 Out-of-Band Technique



The Out-of-Band technique has the advantage of being relatively simple to set up, and no management VPN is required. However, its disadvantages are that it is expensive since it requires an IPv4 connection to each CE. Also, due to the delicate staging requirements for this technique, the Out-of-Band implementation does have a high degree of complexity.

Provisioning a Management CE in Prime Fulfillment

The Prime Fulfillment network management subnet is connected to the Management CE (MCE). The MCE emulates the role of a customer edge router (CE), but the MCE is in provider space and serves as a network operations center gateway router. The MCE is part of a management site as defined in Prime Fulfillment.

Defining CE as MCE

You configure the MCE by identifying the CE as part of the management LAN in Prime Fulfillment software. To do this, perform the following steps:

Step 1 Choose **Inventory > Resources > Customer Devices**.

The list of CPE devices for all currently defined customers is displayed.

Step 2 Choose the CE that will function as the MCE in the management VPN, then click **Edit**.

The Edit CPE Device dialog box appears, displaying the pertinent information for the selected CPE.

Step 3 **Management Type:** From the drop-down list, set the management type to **Managed—Management LAN**.

Step 4 Click **Save**.

You return to the list of CPE devices, where the new management type for the selected CE (in our example, 3. mlce8.cisco.com) is now displayed.

Creating MCE Service Requests

To create an MCE service request, perform the following steps:

Step 1 Choose **Operate > Service Requests**.

The Select MPLS Policy window appears.

This window displays the list of all the MPLS service policies that have been defined in Prime Fulfillment.

Step 2 Choose the policy of choice, then click **OK**.

The MPLS Service Request Editor appears.

Step 3 Click **Add Link**.

The MPLS Service Request Editor now displays a set of fields. Notice that the Select CE field is enabled. Specifying the CE for the link is the first task required to define the link for this service.

Step 4 **CE:** Click **Select CE**.

The Select CPE Device dialog box appears.

a. From the "Show CPEs with" drop-down list, you can display CEs by Customer Name, by Site, or by Device Name.

b. You can use the **Find** button to either search for a specific CE, or to refresh the display.

c. You can set the "Rows per page" to **5**, **10**, **20**, **30**, **40**, or **All**.

d. This dialog box displays the first page of the list of currently defined CE devices. The number of pages of information is displayed in the lower right corner of the dialog box.

To go to the another page of CE devices, click the number of the page you want to go to.

Step 5 In the Select column, choose the name of the MCE for the MPLS link, then click **Select**.

You return to the Service Request Editor window, where the name of the selected CE is now displayed in the CE column.

Step 6 **CE Interface:** Choose the CE interface from the drop-down list.

Note that in the PE column, the **Select PE** option is now enabled.

Step 7 **PE:** Click **Select PE**.

The Select PE Device dialog box appears.

Step 8 In the Select column, choose the name of the PE for the MPLS link, then click **Select**.

You return to the Service Request Editor window, where the name of the selected PE is now displayed in the PE column.

Step 9 **PE Interface:** Choose the PE interface from the drop-down list.

The Link Attribute **Add** option is now enabled.

Step 10 In the Link Attribute column, click **Add**.

The MPLS Link Attribute Editor window appears, showing the fields for the interface parameters.

The field values displayed in this window reflect the values specified in the service policy associated with this service. For details on each of the PE and CE interface fields, see Specifying PE and CE Interface Parameters, page 5-42.

Note The VLAN ID is shared between the PE and CE, so there is one VLAN ID for both. The Second VLAN ID is an optional attribute that provides a method to match the Q-in-Q second VLAN tag of incoming frames on the PE interface. For usage details about these attributes, see Notes on the VLAN ID and Second VLAN ID Attributes, page 5-83.

Step 11 Edit any interface values that need to be modified for this particular link, then click **Next**.

The MPLS Link Attribute Editor for the IP Address Scheme appears.

The field values displayed in this dialog box reflect the values specified in the service policy associated with this service. For details on the IP address scheme fields, see Specifying the IP Address Scheme, page 5-45.

Step 12 Edit any IP address scheme values that need to be modified for this particular link, then click **Next**.

The MPLS Link Attribute Editor for Routing Information appears.

The field values displayed in this dialog box reflect the values specified in the service policy associated with this service. For details on the routing information for the PE and CE, see Specifying the Routing Protocol for a Service, page 5-48.

Because the service policy used for this service specified the routing protocol as editable, you can change the routing protocol for this service request as needed.

Step 13 Edit any routing protocol values that need to be modified for this particular link, then click **Next**.

The MPLS Link Attribute Editor for the VRF and VPN attributes appears. The field values displayed in this dialog box reflect the values specified in the service policy associated with this service. For details on the VRF and VPN information, see Defining VRF and VPN Information, page 5-72.

Note For more information on setting the VRF and VPN attributes in MPLS VPN service requests, see Defining VRF and VPN Attributes in an MPLS Service Request, page 5-85.

Step 14 Edit any VRF values that need to be modified for this particular link.

Step 15 Click the **Next** button to associate templates or data files to the service request.

The MPLS Link Attribute Editor - Template Association window appears. In this window, you can associate templates and data files with a device by clicking the **Add** button in Template/Data File column for the device. When you click the **Add** button, the Add/Remove Templates window appears. For instructions about associating templates with service requests and how to use the features in this window, see Chapter 9, "Managing Templates and Data Files."

Note The above step assumes the policy on which the service request is based has template association enabled. If not, there will be no **Next** button visible in the GUI. In that case, click **Finish** and return to the MPLS Service Request Editor window and proceed with Step 34, below.

Step 16 When you have completed setting up templates and data files for any device(s), click **Finish** in the Template Association window to close it and return to the MPLS Service Request Editor window.

The MPLS Service Request Editor window reappears.

Step 17 You can add additional links to this service request by choosing **Add Link** and specifying the attributes of the next link in the service.

Step 18 To save your work in the MPLS Service Request Editor window, click **Save**.

You return to the Service Requests window, where the service request is in the Requested state and ready to deploy.

Adding PE-CE Links to Management VPNs

When you have created the Management VPN, then you can proceed to add service for the PE-CE links you want to participate in the Management VPN. To do this, perform the following steps:

Step 1 Navigate to the MPLS Link Attribute Editor - VRF and VPN window for the selected CE.

Step 2 Check the **Join the management VPN** option.

When you join the CE with the Management VPN in this step, Prime Fulfillment generates the appropriate route-map statements in the PE configlet. The function of the management route map is to allow only the routes to the specific CE into the management VPN. Cisco IOS supports only one export route map and one import route map per VRF (and therefore, per VPN).

Step 3 Complete the service request user interface.

Provisioning Cable Services

Using MPLS VPN technology, service providers can create scalable and efficient private networks using a shared Hybrid Fiber Coaxial (HFC) network and Internet Protocol (IP) infrastructure. The cable MPLS VPN network consists of the following two major elements:

- The Multiple Service Operator (MSO) or cable company that owns the physical infrastructure and builds VPNs for the Internet Service Providers (ISPs) to move traffic over the cable and IP backbone.
- ISPs that use the HFC network and IP infrastructure to supply Internet service to cable customers.

Benefits of Cable MPLS VPNs

Provisioning cable services with MPLS VPNs provides the following benefits:

- MPLS VPNs give cable MSOs and ISPs a manageable way of supporting multiple access to a cable plant.

  Service providers can create scalable and efficient VPNs across the core of their networks MPLS VPNs provide systems support scalability in cable transport infrastructure and management.
- Each ISP can support Internet access services from a subscriber's PC through an MSO's physical cable plant to their networks.
- MPLS VPNs allow MSOs to deliver value-added services through an ISP, and thus, deliver connectivity to a wider set of potential customers.

  MSOs can partner with ISPs to deliver multiple services from multiple ISPs and add value within the MSO's own network using VPN technology.
- Subscribers can choose combinations of services from various service providers.
- The Cisco IOS MPLS VPN cable feature sets build on Cable Modem Termination Server (CMTS) and DOCSIS 1.0 extensions to ensure services are reliably and optimally delivered over the cable plant.

  MPLS VPN provides systems support domain selection, authentication per subscriber, selection of QoS, policy-based routing, and ability to reach behind the cable modem to subscriber end-devices for QoS and billing, while preventing session-spoofing.
- MPLS VPN technology ensures both secure access across the shared cable infrastructure and service integrity.

The Cable MPLS VPN Network

As shown in Figure 5-21, each ISP moves traffic to and from a subscriber's PC, through the MSO's physical network infrastructure, to the ISP's network. MPLS VPNs, created in Layer 3, provide privacy and security by constraining the distribution of VPN routes only to the routers that belong to its network. Thus, each ISP's VPN is insulated from other ISPs that use the same MSO infrastructure.

In the MPLS-based cable scheme, a VPN is a private network built over a shared cable plant and MPLS-core backbone. The public network is the shared cable plant or backbone connection points. A cable plant can support Internet access services and carry traffic for an MSO and its subscribers, as well as for multiple Internet Service Providers (ISPs) and their subscribers.

An MPLS VPN assigns a unique VPN Routing/Forwarding (VRF) instance to each VPN. A VRF instance 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 the contents of the forwarding table.

Each PE router maintains one or more VRF tables. If a packet arrives directly through an interface associated with a particular VRF, the PE looks up a packet's IP destination address in the appropriate VRF table. MPLS VPNs use a combination of BGP and IP address resolution to ensure security.

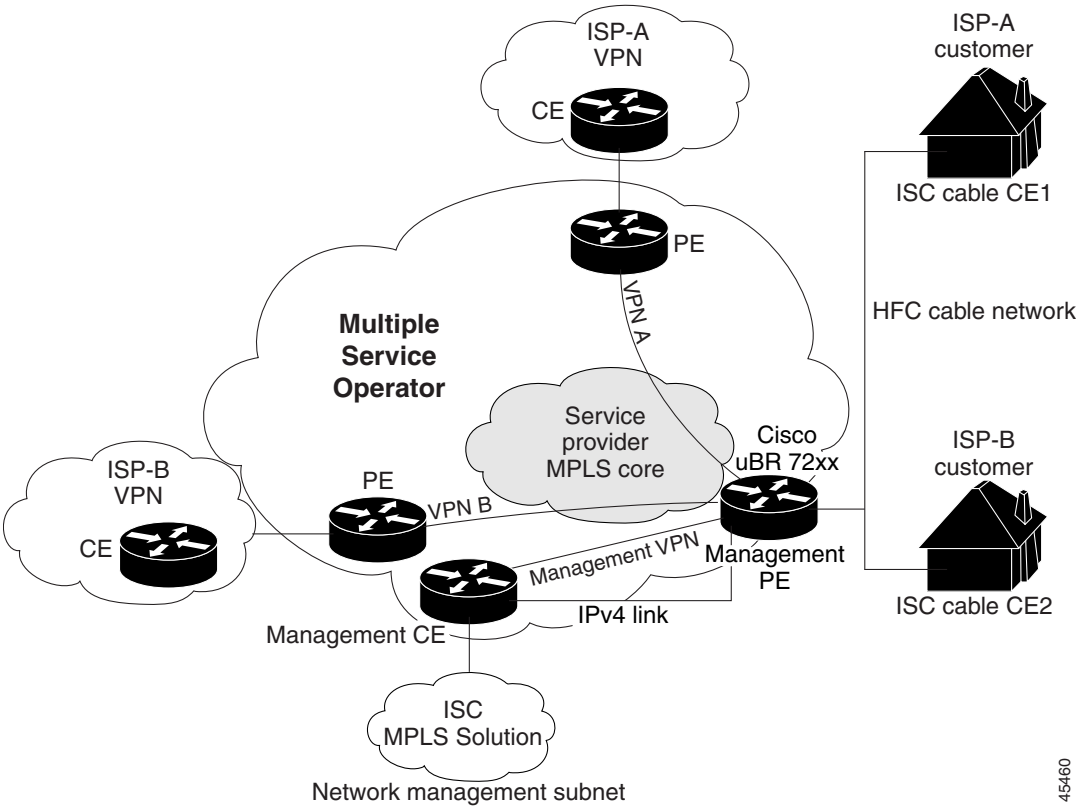
The routers in the cable network are as follows:

- Provider (P) router—Routers in the MPLS core of the service provider network. P routers run MPLS switching, and do not attach VPN labels (MPLS labels in each route assigned by the PE router) to routed packets. VPN labels direct data packets to the correct egress router.

- Provider Edge (PE) router—A router that attaches the VPN label to incoming packets based on the interface or subinterface on which they are received. A PE router attaches directly to a CE router. In the MPLS-VPN approach, each Cisco uBR72xx series router acts as a PE router.
- Customer (C) router—A router in the ISP or enterprise network.
- Customer Edge (CE) router—Edge router on the ISP's network that connects to the PE router on the MSO's network. A CE router must interface with a PE router.
- Management CE (MCE) router—The MCE emulates the role of a customer edge router (CE), but the MCE is in provider space and serves as a network operations center gateway router. The network management subnet is connected to the Management CE (MCE). The MCE is part of a management site as defined in the Prime Fulfillment.
- Management PE (MPE) router—The MPE emulates the role of a PE in the provider core network. The MPE connects the MCE to the provider core network. An MPE can have a dual role as both a PE and the MPE.

The shared cable plant supports Internet connectivity from ISP A to its subscribers and from ISP B to its subscribers.

Figure 5-21 Example of an MPLS VPN Cable Network



Management VPN in the Cable Network

The MPLS network has a unique VPN that exclusively manages the MSOs devices called the management VPN. It contains servers and devices that other VPNs can access. The management VPN connects the Management CE (MCE) router and the management subnet to the MSO PE router (a

uBr72xx router or equivalent). Prime Fulfillment and the management servers, such as Dynamic Host Configuration Protocol (DHCP), Cisco Network Registrar (CNR) Time of Day (ToD) are part of the management subnet and are within the management VPN for ISP connectivity. For an explanation of the management VPN, see Provisioning Management VPN, page 5-120

As shown in Figure 5-21, the management VPN is comprised of the network management subnet (where the Prime Fulfillment workstation resides), which is directly connected to the Management CE (MCE). The management VPN is a special VPN between the MCE and the cable VPN gateway. The cable VPN gateway is usually a Cisco uBR 72xx router that functions as both a regular PE and a Management PE. Notice that there is also a parallel IPv4 link between the MCE and the MPE.

Cable VPN Configuration Overview

Cable VPN configuration involves the following:

- An MSO domain that requires a direct peering link to each enterprise network (Prime Fulfillment), provisioning servers for residential and commercial subscribers, and dynamic DNS for commercial users. The MSO manages cable interface IP addressing, Data Over Cable Service Interface Specifications (DOCSIS) provisioning, cable modem host names, routing modifications, privilege levels, and user names and passwords.
- An ISP or enterprise domain that includes the DHCP server for subscriber or telecommuter host devices, enterprise gateway within the MSO address space, and static routes back to the telecommuter subnets.

Note Cisco recommends that the MSO assign all addresses to the end user devices and gateway interfaces. The MSO can also use split management to let the ISP configure tunnels and security.

To configure MPLS VPNs for cable services, the MSO must configure the following:

- Cable Modem Termination System (CMTS). The CMTS is usually a Cisco uBR72xx series router. The MSO must configure Cisco uBR72xx series routers that serve the ISP.
- PE routers. The MSO must configure PE routers that connect to the ISP as PEs in the VPN.

Tip When configuring MPLS VPNs for cable services, you must configure the cable maintenance subinterface on the PE. The cable maintenance interface is the means by which the cable device retrieves its own IP address. For this reason, the maintenance subinterface must be configured before cable services provisioning can take place.

- CE routers.
- P routers.
- One VPN per ISP.
- DOCSIS servers for all cable modem customers. The MSO must attach DOCSIS servers to the management VPN and make them visible to the network.

The MSO must determine the *primary IP address range*. The primary IP address range is the MSO's address range for all cable modems that belong to the ISP subscribers.

The ISP must determine the *secondary IP address range*. The secondary IP address is the ISP's address range for its subscriber PCs.

To reduce security breaches and differentiate DHCP requests from cable modems in VPNs or under specific ISP management, MSOs can use the **cable helper-address** command in Cisco IOS software. The MSO can specify the host IP address to be accessible only in the ISP's VPN. This lets the ISP use its DHCP server to allocate IP addresses. Cable modem IP address must be accessible from the management VPN.

In Prime Fulfillment, you specify the maintenance helper address and the host helper address and the secondary addresses for the cable subinterface.

Cable VPN Interfaces and Subinterfaces

In the cable subscriber environment, several thousand subscribers share a single physical interface. Configurations with multiple logical subinterfaces are a vital part of the MPLS VPN network over cable. You can configure multiple subinterfaces and associate a specific VRF with each subinterface. You can split a single physical interface (the cable plant) into multiple subinterfaces, where each subinterface is associated with a specific VRF. Each ISP requires access on a physical interface and is given its own subinterface. The MSO administrator can define subinterfaces on a cable physical interface and assign Layer 3 configurations to each subinterface.

The MPLS VPN approach of creating VPNs for individual ISPs or customers requires subinterfaces to be configured on the cable interface. One subinterface is required for each ISP. The subinterfaces are tied to the VPN Routing/Forwarding (VRF) tables for their respective ISPs.

You must create the maintenance subinterface on the cable interface and tie it to the management VPN. The maintenance interface is for the ISP's use, and it is used for VPN connectivity, as well as the management VPN using an extranet between the ISP and the management VPN.

Prime Fulfillment automatically selects the subinterface number based on the VRF. If a subinterface that is associated with the current VRF does not yet exist, Prime Fulfillment creates a subinterface and assigns it to the correct VRF. The subinterface number is incremented to 1 greater than the largest subinterface currently assigned for the selected cable interface.

The network management subnet (which includes the CNR, ToD, and Prime Fulfillment) can reply to the cable modem because the management VPN allows connectivity for one filtered route from the ISP's VPN to the Management CE (MCE). Similarly, in order to forward the management requests (such as DHCP renewal to CNR), the ISP VPN must import a route to the MCE in the management VPN.

Cisco uBR7200 series software supports the definition of logical network layer interfaces over a cable physical interface. The system supports subinterface creation on a physical cable interface.

Subinterfaces allow traffic to be differentiated on a single physical interface and associated with multiple VPNs. Each ISP requires access on a physical interface and is given its own subinterface. Using each subinterface associated with a specific VPN (and therefore, ISP) subscribers connect to a logical subinterface, which reflects the ISP that provides their subscribed services. Once properly configured, subscriber traffic enters the appropriate subinterface and VPN.

Provisioning Cable Services in Prime Fulfillment

The tasks you must complete to provision cable services in Prime Fulfillment are as follows:

- Add the PE that has cable interfaces to the appropriate Region.
- Generate a service request to provision the cable maintenance interface on the PE.
- Generate a second service request to provision the MPLS-based cable service. You must generate this cable service request for each VPN.

When using the Prime Fulfillment to provision cable services, there are no CEs in the same sense there are when provisioning a standard MPLS VPN. Thus, you must use a PE-only policy or create a cable policy with no CE.

Creating the Service Requests

This section contains the following subsections:

Creating a Cable Subinterface Service Request

The cable maintenance subinterface on the PE is the means by which the cable device retrieves its own IP address. For this reason, the maintenance subinterface must be configured before provisioning cable services. To create a cable subinterface service request, perform the following steps:

Step 1 Choose **Operate > Service Requests > MPLS**.

The MPLS Policy Selection dialog box appears. This dialog box displays the list of all the MPLS service policies that have been defined in Prime Fulfillment.

Step 2 Choose the PE-Only policy (**cable** in the example above) policy, and then click **OK**.

The MPLS Service Request Editor appears.

Step 3 Click **Add Link**.

The MPLS Service Request Editor now displays a set of fields. Notice that the Select PE field is enabled. Specifying the PE for the link is the first task required to define the link for this service.

Step 4 **PE:** Click **Select PE**.

The Select PE Device dialog box appears.

Step 5 In the Select column, choose the name of the PE for the MPLS link, then click **Select**.

You return to the Service Request Editor window, where the name of the selected PE is now displayed in the PE column.

Step 6 **PE Interface:** Choose the PE interface from the drop-down list.

Only the major interface names are available for you to select. Prime Fulfillment assigns the appropriate subinterface number for each VPN.

The Link Attribute **Add** option is now enabled.

Step 7 In the Link Attribute column, click **Add**.

The MPLS Link Attribute Editor is displayed, showing the fields for the interface parameters.

Step 8 Enter a subinterface name in the Interface Description field.

Step 9 Check the check box for the Cable Maintenance Interface, then click **Edit beside Cable Helper Addresses**.

The Cable Helper Addresses window appears.

Step 10 Click **Add**.

The Cable Helper Addresses window appears.

Step 11 Enter an **IP address** in the IP Address field and choose **Both** for IP Type.

Cable Modems and their attached CPE devices (hosts) will broadcast DHCP packets to the destination IP address, and this destination IP address is the configured cable helper address. So, from configured cable helper address, cable modems and their attached CPE (hosts) will receive their (CM and CPE) IP address.

IP Type can have the following values:

- Host—When selected, only UDP broadcasts from hosts (CPE devices) are forwarded to that particular destination IP address. (For example, only hosts will receive IP addresses from the mentioned helper address.)
- Modem—When selected, only UDP broadcasts from cable modems are forwarded to that particular destination IP address. (For example, only cable modems will receive IP addresses from the mentioned helper address.)
- Both—When selected, UDP broadcasts from hosts (CPE devices) and cable modems are forwarded to that particular destination IP address. (For example, both cable modems and hosts will receive IP addresses from the mentioned helper address.)

Step 12 Click **OK**.

The MPLS Link Attribute Editor reappears.

Step 13 Click **Next**.

The MPLS Link Attribute Editor - IP Address Scheme appears.

Step 14 Edit any IP address scheme values that must be modified for this particular link, then click **Next**. The MPLS Link Attribute Editor for Routing Information appears.

The following routing protocol options are supported:

- STATIC
- RIP
- OSPF
- EIGRP
- None

Because the service policy used for this service specified the routing protocol as editable, you can change the routing protocol for this service request as needed.

Step 15 Edit any routing protocol values that must be modified for this particular link, then click **Next**.

Note For information about protocol types, see Specifying the Routing Protocol for a Service, page 5-48.

The MPLS Link Attribute Editor for the VRF and VPN attributes appears. The field values displayed in this dialog box reflect the values specified in the service policy associated with this service.

Note If you want to set the VRF and VPN attributes via a previously defined VRF object, check the **Use VRF Object** check box. For more information on this feature, see Independent VRF Management, page 5-14. That section describes how to use independent VRF objects in MPLS VPN service policies and service requests.

Note For more information on setting the VRF and VPN attributes in MPLS VPN service requests, see Defining VRF and VPN Attributes in an MPLS Service Request, page 5-85.

Step 16 Check the check box for **Join the Management VPN**.

Step 17 Edit any VRF and VPN values that must be modified for this particular link.

Step 18 Click the **Next** button to associate templates or data files to the service request.

The MPLS Link Attribute Editor - Template Association window appears. In this window, you can associate templates and data files with a device by clicking the **Add** button in Template/Data File column for the device. When you click the **Add** button, the Add/Remove Templates window appears. For instructions about associating templates with service requests and how to use the features in this window, see Chapter 9, "Managing Templates and Data Files."

Note The above step assumes the policy on which the service request is based has template association enabled. If not, there will be no **Next** button visible in the GUI. In that case, click **Finish** and return to the MPLS Service Request Editor window and proceed with Step 34, below.

Step 19 When you have completed setting up templates and data files for any device(s), click **Finish** in the Template Association window to close it and return to the MPLS Service Request Editor window.

Note You can define multiple links in this service request.

Step 20 To save your work on this service request, click **Save**.

The MPLS Service Requests window reappears showing that the service request is in the Requested state and ready to deploy.

Creating Cable Link Service Requests

To create a cable link service request, perform the following steps:

Step 1 Choose **Operate > Service Requests > MPLS**.

The MPLS Policy Selection dialog box appears. This dialog box displays the list of all the MPLS service policies that have been defined in Prime Fulfillment.

Step 2 Choose the policy of choice, then click **OK**.

The MPLS Service Request Editor appears.

Step 3 Click **Add Link**.

The MPLS Service Request Editor now displays a set of fields. Note that in the PE column, the **Select PE** option is now enabled.

Step 4 **PE:** Click **Select PE**.

The Select PE Device dialog box appears.

Step 5 In the Select column, choose the name of the PE for the MPLS link, then click **Select**.

You return to the Service Request Editor window, where the name of the selected PE is now displayed in the PE column.

Step 6 **PE Interface:** Choose the PE interface from the drop-down list.

Note that the Link Attribute **Add** option is now enabled.

Step 7 In the Link Attribute column, click **Add**.

The MPLS Link Attribute Editor appears, showing the fields for the interface parameters.

Note Do not check the box for Cable Maintenance Interface.

Step 8 Edit any interface values that must be modified for this particular link, then click **Edit** beside Cable Helper Addresses.

The Cable Helper Addresses window appears.

Step 9 Click **Add**.

The Cable Helper Addresses window appears.

Step 10 Enter an **IP address** in the IP Address field and choose **Both**, **Modem,** or **Host** for IP Type.

Cable Modems and their attached CPE devices (hosts) will broadcast DHCP packets to the destination IP address, and this destination IP address is the configured cable helper address. So, from configured cable helper address, cable modems and their attached CPE (hosts) will receive their (CM and CPE) IP address.

IP Type can have the following values:

- Host—When selected, only UDP broadcasts from hosts (CPE devices) are forwarded to that particular destination IP address. (For example, only hosts will receive IP addresses from the mentioned helper address.)
- Modem—When selected, only UDP broadcasts from cable modems are forwarded to that particular destination IP address. (For example, only cable modems will receive IP addresses from the mentioned helper address.)
- Both—When selected, UDP broadcasts from hosts (CPE devices) and cable modems are forwarded to that particular destination IP address. (For example, both cable modems and hosts will receive IP addresses from the mentioned helper address.)

Step 11 Click **OK**.

The MPLS Link Attribute Editor reappears.

Step 12 Click **Edit** beside Secondary Addresses.

The Cable Secondary Addresses window appears. The secondary IP address enables CPE devices (hosts) attached to cable modem to talk to CMTS. (Usually this is a public IP address so that PCs can go to internet.)

Step 13 Enter an IP address in the IP address/Mask field and click **OK**.

The MPLS Link Attribute Editor reappears.

Step 14 Click **Next**.

Step 15 The MPLS Link Attribute Editor for the IP Address Scheme appears.

Step 16 Edit any IP address scheme values that must be modified for this particular link, then click **Next**.

The MPLS Link Attribute Editor for Routing Information appears.

Note For information about protocol types, see Specifying the Routing Protocol for a Service, page 5-48.

Step 17 Edit any routing protocol values that must be modified for this particular link, then click **Next**.

The MPLS Link Attribute Editor for the VRF and VPN attributes appears. The field values displayed in this dialog box reflect the values specified in the service policy associated with this service.

Note If you want to set the VRF and VPN attributes via a previously defined VRF object, check the **Use VRF Object** check box. For more information on this feature, see Independent VRF Management, page 5-14. That section describes how to use independent VRF objects in MPLS VPN service policies and service requests.

Note For more information on setting the VRF and VPN attributes in MPLS VPN service requests, see Defining VRF and VPN Attributes in an MPLS Service Request, page 5-85.

Step 18 Check the check box for Join the Management VPN.

Step 19 Edit any VRF and VPN values that must be modified for this particular link, then click **Add**.

The Select CERCs/VPN dialog box appears.

Step 20 Choose the customer name and VPN.

Step 21 Click **Join as Spoke**, then click **Done**.

The MPLS Link Attribute Editor for the VRF and VPN attributes appears.

Step 22 Edit any VRF and VPN values that must be modified for this particular link.

Step 23 Click the **Next** button to associate templates or data files to the service request.

The MPLS Link Attribute Editor - Template Association window appears. In this window, you can associate templates and data files with a device by clicking the **Add** button in Template/Data File column for the device. When you click the **Add** button, the Add/Remove Templates window appears. For instructions about associating templates with service requests and how to use the features in this window, see Chapter 9, "Managing Templates and Data Files."

Note The above step assumes the policy on which the service request is based has template association enabled. If not, there will be no **Next** button visible in the GUI. In that case, click **Finish** and return to the MPLS Service Request Editor window and proceed with Step 27, below.

Step 24 When you have completed setting up templates and data files for any device(s), click **Finish** in the Template Association window to close it and return to the MPLS Service Request Editor window.

Note You can define multiple links in this service request.

Step 25 To save your work on this service request, click **Save**.

The MPLS Service Requests window reappears showing that the service request is in the Requested state and ready to deploy.

Provisioning Carrier Supporting Carrier

This section describes how to configure the carrier supporting carrier (CSC) feature using the Prime Fulfillment provisioning process. It contains the following sections:

Carrier Supporting Carrier Overview

The Carrier Supporting Carrier (CSC) feature enables one MPLS VPN-based service provider to allow other service providers to use a segment of its backbone network. The service provider that provides the segment of the backbone network to the other provider is called the backbone carrier. The service provider that uses the segment of the backbone network is called the customer carrier.

This documentation focuses on a backbone carrier that offers Border Gateway Protocol and Multiprotocol Label Switching (BGP/MPLS) VPN services. There can be two types of customer carriers:

- An Internet service provider (ISP)
- A BGP/MPLS VPN service provider

This documentation describes both types of customer carrier.

It is transparent to the backbone provider when either scenario is in use, after the required functionality for basic MPLS VPN CSC is implemented in the backbone network.

In Prime Fulfillment, the customer carrier PE device is modeled as a CE device and the backbone carrier PE device is modeled as an N-PE device. An MPLS service request with the CSC option can be created with these PE and CE devices. You can configure the CSC feature on IOS and IOS XR PE devices.

The CSC service is applicable for the following PE-CE link configurations:

- IPv4 Unicast
- IPv4 Multicast

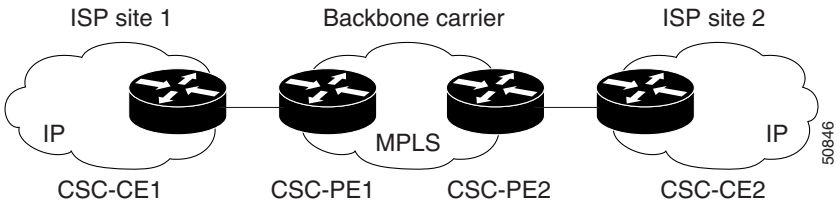
The CSC service is applicable for the BGP PE-CE routing protocol on IOS XR devices.

Backbone Network with ISP Customer Carrier

In this network configuration, the customer carrier has two sites, each of which is a point of presence (POP). The customer carrier connects these sites using a VPN service provided by a backbone carrier, who uses MPLS. The ISP sites use IP. To enable packet transfer between the ISP sites and the backbone carrier, the CSC-CE routers that connect the ISPs to the backbone carrier run MPLS.

Figure 5-22 shows a carrier supporting carrier network configuration where the customer carrier is an ISP. The customer carrier has two sites, each of which is a point of presence (POP). The customer carrier connects these sites using a VPN service provided by the backbone carrier. The backbone carrier uses MPLS. The ISP sites use IP. To enable packet transfer between the ISP sites and the backbone carrier, the CSC-CE routers that connect the ISPs to the backbone carrier run MPLS.

Figure 5-22 Carrier Supporting Carrier Network with an ISP Customer Carrier



In this example, only the backbone carrier uses MPLS. The customer carrier (ISP) uses only IP. As a result, the backbone carrier must carry all the Internet routes of the customer carrier, which could be as many as 100,000 routes. This poses a scalability problem for the backbone carrier. To solve the scalability problem, the backbone carrier is configured as follows:

- The backbone carrier allows only internal routes of the customer carrier (IGP routes) to be exchanged between the CSC-CE routers of the customer carrier and the CSC-PE routers of the backbone carrier.
- MPLS is enabled on the interface between the CSC-CE router of the customer carrier and the CSC-PE router of the backbone carrier.

Internal and external routes are differentiated this way:

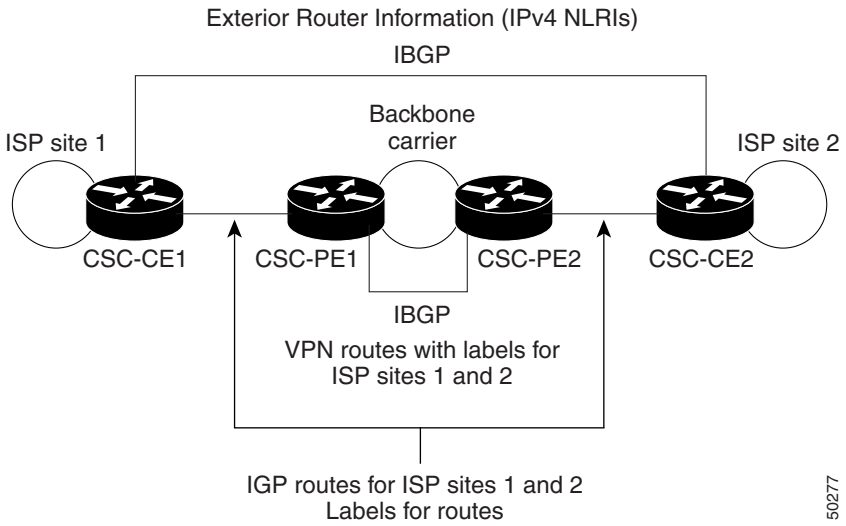
- Internal routes go to any of the routers within the ISP.
- External routes go to the Internet.

The number of internal routes is much smaller than the number of external routes. Restricting the routes between the CSC-CE routers of the customer carrier and the CSC-PE routers of the backbone carrier significantly reduces the number of routes that the CSC-PE router needs to maintain.

Since the CSC-PE routers do not have to carry external routes in the VRF routing table, they can use the incoming label in the packet to forward the customer carrier Internet traffic. Adding MPLS to the routers provides a consistent method of transporting packets from the customer carrier to the backbone carrier. MPLS allows the exchange of an MPLS label between the CSC-PE and the CSC-CE routers for every internal customer carrier route. The routers in the customer carrier have all the external routes either through iBGP or route redistribution to provide Internet connectivity.

Figure 5-23 shows how information is exchanged when the network is configured in this manner.

Figure 5-23 Backbone Carrier Exchanging Routing Information with a Customer Carrier Who Is an ISP



Backbone Network with BGP/MPLS VPN Service Provider Customer Carrier

When a backbone carrier and the customer carrier both provide BGP/MPLS VPN services, the method of transporting data is different from when a customer carrier provides only ISP services. The following list highlights those differences.

- When a customer carrier provides BGP/MPLS VPN services, its external routes are VPN-IPv4 routes. When a customer carrier is an ISP, its external routes are IP routes.
- When a customer carrier provides BGP/MPLS VPN services, every site within the customer carrier must use MPLS. When a customer carrier is an ISP, the sites do not need to use MPLS.

Figure 5-24 figure shows a carrier supporting carrier network configuration where the customer carrier is an MPLS VPN provider. The customer carrier has two sites. The backbone carrier and the customer carrier use MPLS. The iBGP sessions exchange the external routing information of the ISP.

Figure 5-24 Carrier Supporting Carrier Network with a Customer Carrier Who Is an MPLS VPN Provider

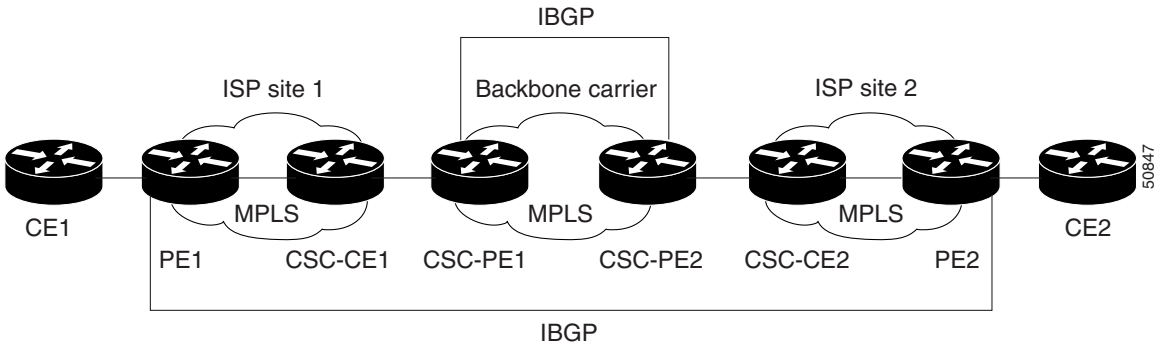
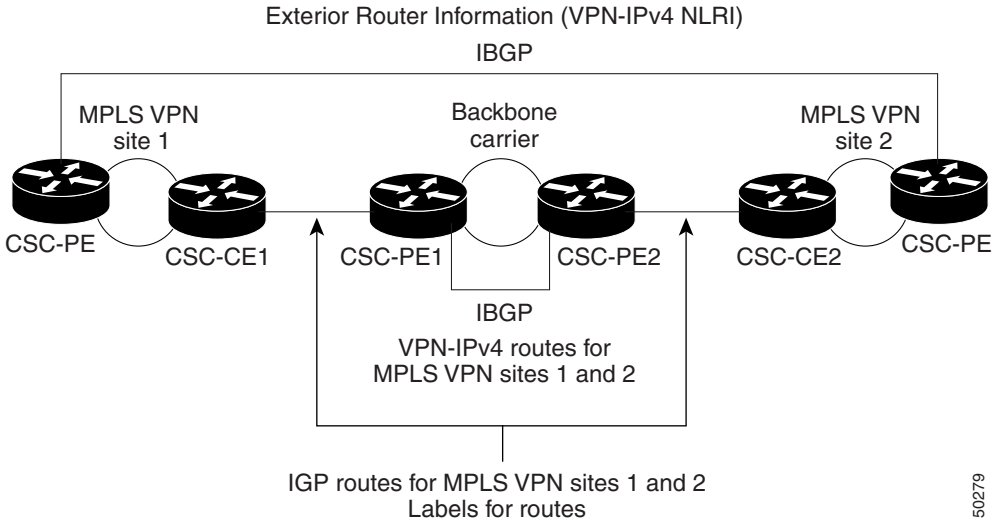


Figure 5-24 figure shows exchanging information with a customer carrier who is an MPLS VPN service provider.

Figure 5-25 Backbone Carrier Exchanging Information with a Customer Carrier Who Is an MPLS VPN Service Provider



Prime Fulfillment Configuration Options

To configure the CSC network to exchange routes and carry labels between the backbone carrier provider edge (CSC-PE) routers and the customer carrier customer edge (CSC-CE) routers, use Label Distribution Protocol (LDP) to carry the labels and an Interior Gateway Protocol (IGP) to carry the routes.

LDP/IGP

A routing protocol is required between the CSC-PE and CSC-CE routers that connect the backbone carrier to the customer carrier. The routing protocol enables the customer carrier to exchange IGP routing information with the backbone carrier. RIP, OSPF, or static routing as the routing protocol can be selected.

Label distribution protocol (LDP) is required between the CSC-PE and CSC-CE routers that connect the backbone carrier to the customer carrier. LDP is also required on the CSC-PE to CSC-CE interface for VPN routing/forwarding (VRF).

IPv4 BGP Label Distribution

BGP takes the place of an IGP and LDP in a VPN forwarding/routing instance (VRF) table. You can use BGP to distribute routes and MPLS labels. Using a single protocol instead of two simplifies the configuration and troubleshooting.

BGP is the preferred routing protocol for connecting two ISPs, mainly because of its routing policies and ability to scale. ISPs commonly use BGP between two providers. This feature enables those ISPs to use BGP.

When BGP (both eBGP and iBGP) distributes a route, it can also distribute an MPLS label that is mapped to that route. The MPLS label mapping information for the route is carried in the BGP update message that contains the information about the route. If the next hop is not changed, the label is preserved.

Defining CSC Service Policies

To define a Service Policy with CSC, choose the CSC Support check box from the MPLS Policy Editor - Routing Information.

When CSC Support is checked, the CSC functionality is enabled to the MPLS VPN service.

Provisioning CSC Service Requests

To provision a service request with CSC, choose the CSC Support check box from the MPLS Link Attribute Editor - Routing Information.

When CSC Support is checked, the CSC functionality is enabled for the MPLS VPN service.

Provisioning Multiple Devices

This section describes how to configure multiple devices, Layer 2 (L2) "switches" and Layer 3 (L3) "routers," using the Prime Fulfillment provisioning process. It contains the following sections:

- NPC Ring Topology, page 5-143
- Ethernet-To-The-Home (ETTH), page 5-147

NPC Ring Topology

This section describes how to create a Ring Topology, connect the CE starting and PE-POP ending points, and configure the Named Physical Circuits (NPC) from end to end, using the Prime Fulfillment provisioning process.

This section contains the following sections:

- Ring Topology Overview, page 5-143
- Creating Ring of Three PE-CLEs, page 5-144
- Configuring NPC Ring Topology, page 5-145

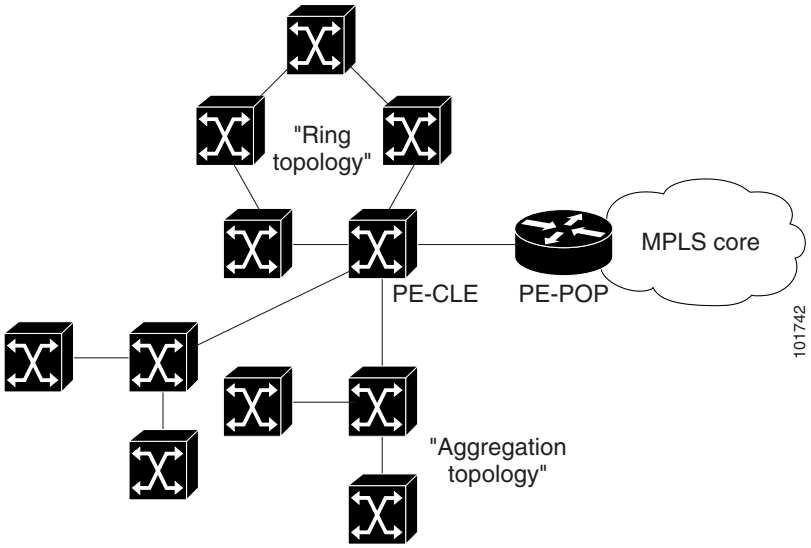
Ring Topology Overview

Service providers are now looking to offer L2 and L3 services that must integrate with a common MPLS infrastructure. Prime Fulfillment supports two basic L2 topologies to access L3 MPLS networks:

- Ring Topology
- Aggregation Topology ("Hub and Spoke")

Figure 5-26 shows an example of these two basic L2 access topologies.

Figure 5-26 L2 Access Topologies

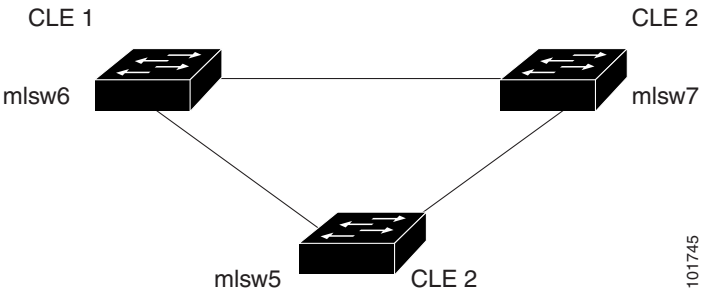


Creating Ring of Three PE-CLEs

In its simplest form, the Ring Topology is a tripartite structure that comprises at least three PE- CLE. A PE-POP and a Multi-VRF CE can also be part of a Ring.

Figure 5-27 shows an example ring of three Catalyst 3550 switches: mlsw5, mlsw6, and mlsw7.

Figure 5-27 A Ring of Three PE-CLE



To create a ring of three PE-CLEs, perform the following steps:

Step 1 Choose **Inventory > Logical Inventory > Physical Rings**.

The Physical Rings window appears.

Step 2 Click **Create** to continue.

The Create Ring window appears.

Step 3 Click **Select source device** in the first cell.

The Show Devices window appears.

Note The Show Devices drop-down window should show *CLE* rather than *PE*. This is a known application error. You cannot initiate this process with a PE-POP or a CE. You must begin with a PE-CLE.

Step 4 To search for a specific CLE, enter the *source device* in the **matching** dialog-box and click **Find**.

Step 5 Choose the CLE and click **Select**.

The Create Ring window appears.

Step 6 Continue from left to right and from top to bottom to fill the table with the appropriate Device and Interface information, which would be based on a network diagram from your own environment.

Note If you had used the network diagram in Figure 5-28 to populate the Create Ring table, it would contain the above information at the end of this process.

Step 7 Click **Save** to save your ring in the Repository.

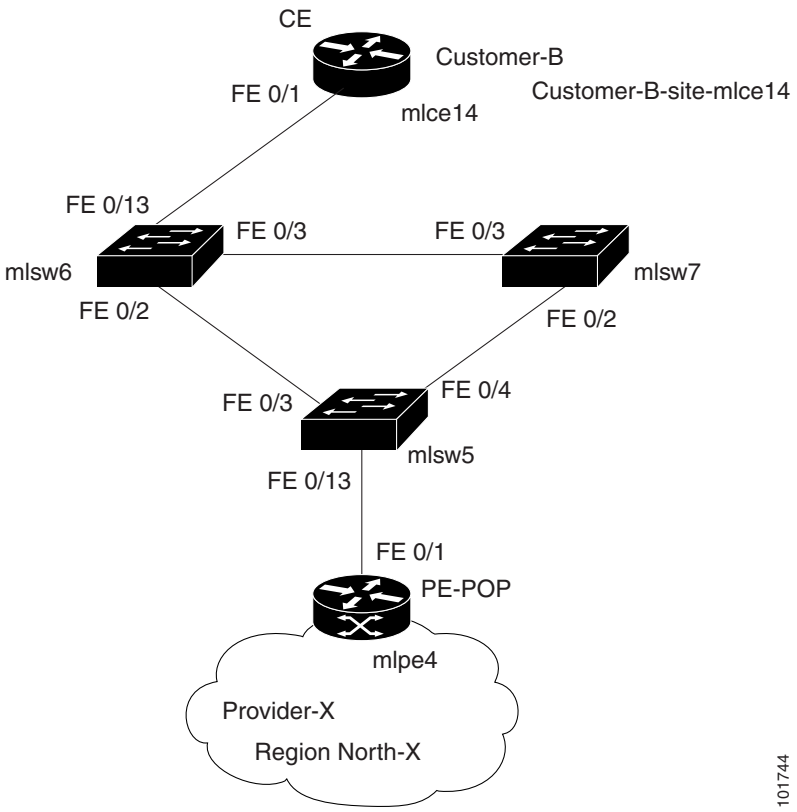
The NPC Rings window appears.

Proceed to Configuring NPC Ring Topology, page 5-145.

Configuring NPC Ring Topology

Figure 5-28 shows an example of the Ring Topology (three CLE) inserted between a CE (**mlce14**) and a PE-POP (**mlpe4**).

Figure 5-28 The Ring Topology



To configure end-to-end connectivity (CE > Ring (PE-CLE) > PE), perform the following steps:

Step 1 Choose **Inventory > Logical Inventory > Named Physical Circuits**.

The Named Physical Circuits window appears.

Step 2 Click **Create**.

The Create Named Physical Circuit window appears.

Step 3 Click **Add Device**.

The Select Devices window appears.

Step 4 Choose the CE and then click **Select**.

The Create Named Physical Circuit window appears.

Step 5 Click **Add Device**.

The Select Devices window appears.

Step 6 Choose the PE and then click **Select**.

The Create Named Physical Circuit window appears.

Step 7 Click **Insert Ring**.

The Show NPC Rings window appears.

Step 8 Choose an NPC Ring and click **Select**.

The Create a Named Physical Circuit window appears

Step 9 Choose a device with an available check box and click **Select device**.

The Select a device from ring window appears.

Step 10 Choose a PE-CLE and click **Select**.

The Create Named Physical Circuit window appears.

Step 11 Choose the incoming and outgoing interfaces for the CE, CLE, and PE until complete.

Step 12 Choose the remaining device with the darkened check box.

The Create a Named Physical Circuit window appears.

Step 13 Click **Save**.

The Named Physical Interfaces window appears.

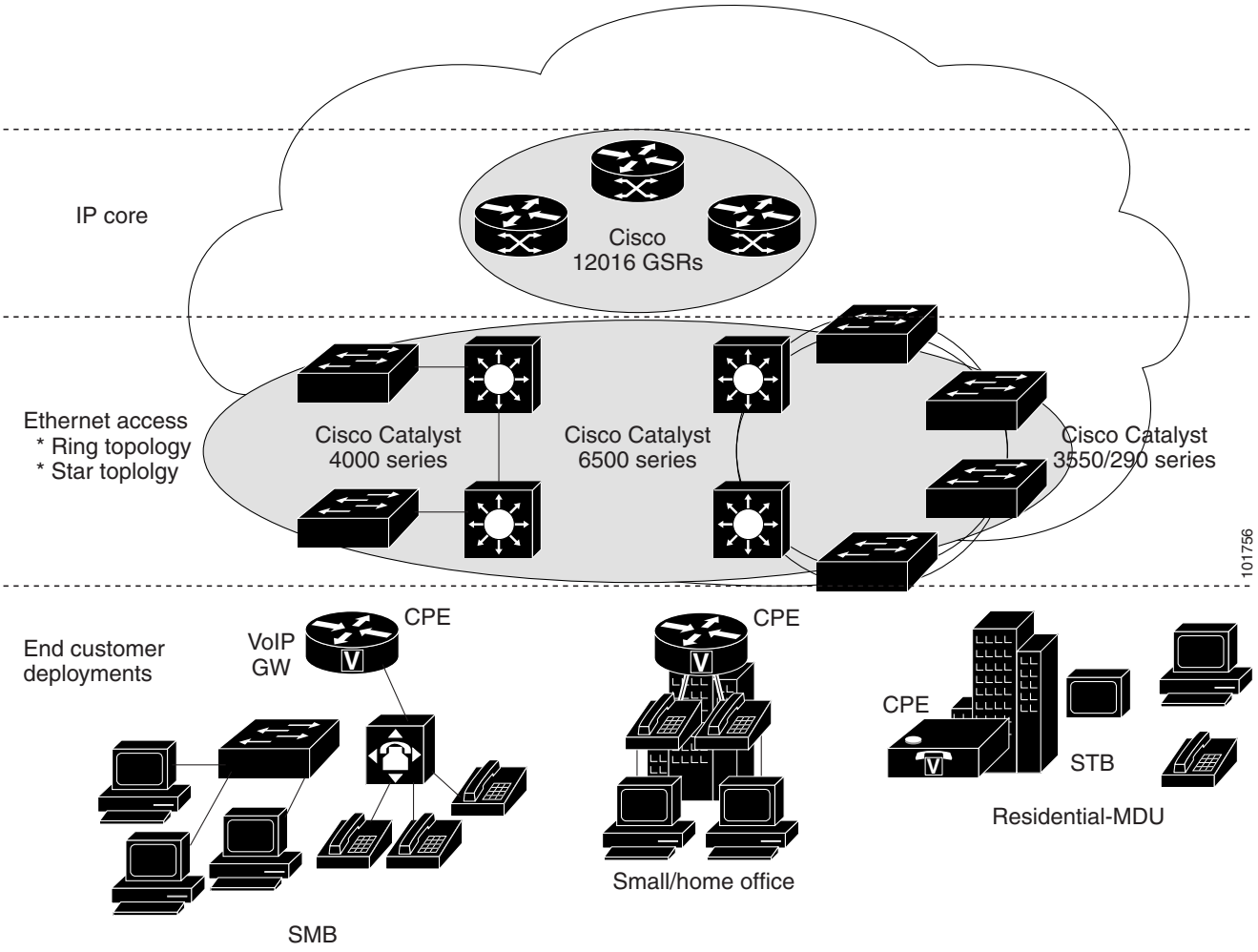
Ethernet-To-The-Home (ETTH)

This section describes how to configure Ethernet-To-The-Home (ETTH) using the Prime Fulfillment provisioning process.

ETTH is part of the Cisco ETTx solution, which contains both ETTH and Ethernet-to-the-Business (ETTB). ETTB is supported in Prime Fulfillment with the L2VPN Metro Ethernet service feature. Unlike ETTB, whose customers are mainly business customers, ETTH is targeted at residential customers.

Figure 5-29 shows an overview of the Cisco ETTx solution.

Figure 5-29 Cisco ETTx Solution



From a provisioning standpoint, the main difference between ETTB and ETTH is the consideration of resource scalability. For example, with ETTB, each business customer is allocated one or more VLAN(s).

With ETTH, it is not practical to assign a unique VLAN to each residential customer. The practical solution is to have all, or a group of residential customers, share the same VLAN and use common technology, such as a private VLAN (PVLAN) or a protected port, to guarantee traffic isolation.

Another difference between ETTB and ETTH is that most of the ETTB customers use an Ethernet trunk port while ETTH customers use an access port. In Prime Fulfillment, the access port is fully supported, with CE present or with no CE.

ETTH needs to support multicast based services, such as video, on a shared media such as a ring. Typically, Internet Group Management Protocol (IGMP) with Multicast VLAN Registration (MVR) would be the technology used to support these services.

Access Domain Management

To provide more flexibility in managing an access domain, you can define a management VLAN. Once defined, the management VLAN is used to construct the list of VLANs allowed on the trunk port for all non-UNI ports.

You can also specify how the VLAN allowed list is constructed in a trunk port for a domain, if the list is not on the device. This feature is implemented for L2VPN DCPL parameter. It is available for Layer 2 access to MPLS VPN as well.

As a part of Layer 2 access management, Prime Fulfillment provides the ability to create MAC access lists by specifying the MAC addresses to be allowed or blocked.

Prime Fulfillment ETTH Implementation

The Prime Fulfillment MPLS VPN implementation of ETTH consists of the following three subfeatures:

- PVLAN or Protected Port, page 5-149
- Access Port, page 5-149
- IGMP with MVR, page 5-149

PVLAN or Protected Port

This feature is used to isolate traffic within a PVLAN. It prevents traffic from flowing between two UNIs.

- PVLAN is only supported on the Catalyst 4500/6500 switches and Cisco 7600 router.
- Protected Port is only supported on the Catalyst 2950/3550 switches.

Access Port

In Prime Fulfillment, the untagged Ethernet default is supported in the CE present and no CE scenarios. You can choose between two encapsulations: DOT1Q and Default.

The Default encapsulation only indicates that the traffic coming in from the CE is untagged. The UNI, which is always a dot1q port, puts a tag on it before transmitting it. UNI has two options to handle this untagged traffic. It functions as an access port or a trunk port. For this reason, the GUI adds one more item for you to choose.

IGMP with MVR

This feature applies to a very specific user service and network topology. It is used for multicast video on a hub and spoke or ring network. However, it is not up to Prime Fulfillment to decide when it is used. Prime Fulfillment only makes it available and the network application running above Prime Fulfillment must invoke it when needed.

Creating an ETTH Policy

To configure a policy to support ETTH, perform the following steps:

Step 1 Choose **Service Design > Policies > Policy Manager**.

Step 2 From the Policy Manager window, choose a Service Policy and click **Edit**.

Step 3 From the Policy Type Information window, click **Next**.

The MPLS Policy Editor - Interface window appears.

Step 4 To enable ETTH, check the **ETTH Support** check box.

The ETTH UNI Information check boxes appear between the **ETTH Support** check box and the CE Information.

Step 5 To enable Private VLAN or Protected Port, check the **Private VLAN/Protected Port** check box.

Step 6 To enable IGMP Snooping with MVR, check the **IGMP Snooping with MVR** check box.

Three new UNI Information options appears.

Step 7 Choose UNI Information options:

- Mode
  - Compatible—Multicast addresses are statically configured on the device.
  - Dynamic—IGMP snooping is configured on the device.
- Query Time—Determines how often the device is queried for membership.
- Immediate—Removes the interface from the forwarding table immediately, when the session ends.

Step 8 Complete the standard steps and click **Save**.

Creating a Service Request for ETTH

To create a service request for ETTH, perform the following steps:

Step 1 Choose **Operate > Service Requests > Service Request Manager**.

Step 2 From the Service Requests Manager window, choose a Service Request and click **Edit**.

Step 3 From the MPLS Service Request Editor window, click **Edited** from the **Link Attribute** link.

The MPLS Link Attribute Editor - Interface window appears.

Step 4 Edit the following Link Attribute specific UNI Information:

- Secondary VLAN ID—Enter a VLAN ID for the Private VLAN, which is supported only on the Catalyst 4000 switch.
- Multicast IP Address—See Step 5.
- Multicast VLAN ID—Enter a *VLAN ID* for the Multicast VLAN.

Step 5 Click **Edit**.

The Multicast IP Addresses dialog box appears.

Step 6 Edit the following Link Attribute specific UNI Information:

- Multicast IP Address—Enter an IP Address for the join the multicast group, which allows users to have access to video on demand, for example.
- Counter—Enter a count to determine the number of contiguous IP addresses starting with the Multicast IP Address.

Step 7 Click **OK**.

Step 8 Complete the standard steps for creating a service request, and click **Save**.

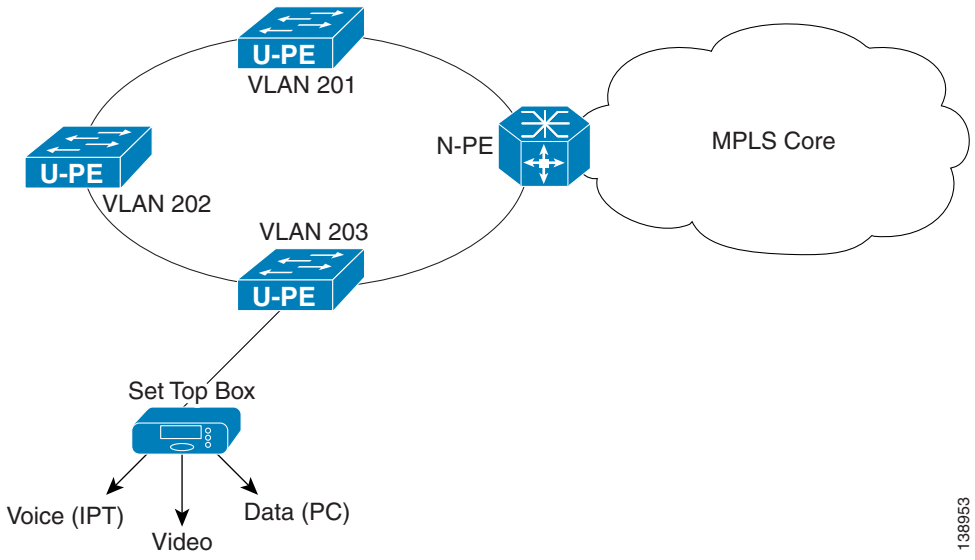
Note For more information on setting the VRF and VPN attributes in MPLS VPN service requests, see Defining VRF and VPN Attributes in an MPLS Service Request, page 5-85.

The MPLS Service Requests window reappears showing that the service request is in the Requested state and ready to deploy.

Residential Service

A group of residential customers can share the same VLAN on the same UNI switch with traffic isolation on different UNI interfaces. On an N-PE, a VRF SVI is defined for all the residential services from the same UNI switch, as shown in Figure 5-30.

Figure 5-30 Residential Services



Creating a Policy for Residential Services Over Shared VLAN

A special policy must be created by enabling Shared VLAN. To do this, perform the following steps:

Step 1 Choose **Operate > Service Requests > MPLS**.

The MPLS Policy Editor - Policy Type window appears.

Step 2 In the Policy Name field, enter a policy name.

Step 3 Under Policy Owner, click the **Global Policy** radio button.

Step 4 Under Policy Type accept **Regular: PE-CE**.

Step 5 Under CE Present, uncheck the check box, then click **Next**.

The MPLS Policy Editor - Interface window appears.

Step 6 Check the **Use SVI:** check box, then wait for the window to refresh.

Step 7 Check the **ETTH Support:** check box, then wait for the window to refresh.

Step 8 Check the **Standard UNI Port:** check box, then wait for the window to refresh.

Step 9 Check the **Shared VLAN:** check box, then wait for the window to refresh. Some fields are now grayed-out.

Note Because this policy enables ETTH Support and Shared VLAN, these attributes become unavailable at the link level.

Step 10 Check the **Private VLAN/Protected Port:** check box, wait for the window to refresh, then click **Next**.

Step 11 In the IP Address Scheme window, you can continue by clicking **Next**.

Step 12 In the Routing Information window, you can continue by clicking **Next**.

Note For information about protocol types, see Specifying the Routing Protocol for a Service, page 5-48.

Step 13 In the VRF and VPN Member window, you can continue by clicking **Next** to associate templates, or else finish creating this policy by clicking **Finish**.

Note For more information on setting the VRF and VPN attributes in MPLS VPN service requests, see Defining VRF and VPN Attributes in an MPLS Service Request, page 5-85.

Creating a Service Request for Residential Services Over Shared VLAN

To create the service request, perform the following steps:

Step 1 Choose **Service Design > Policies > MPLS Policy Editor - Policy Type**.

Step 2 Choose the policy you configured for Shared VLAN Residential Services, then click **OK**. The MPLS Service Request Editor window appears.

Step 3 In the MPLS Service Request Editor window, click **Add Link**, then wait for the window to refresh.

Step 4 Click the active field **Select U-PE**.

Step 5 Choose a PE device, then click **Select**.

Step 6 From the active drop-down list, choose an interface, then wait for the window to refresh.

Step 7 Under Link Attributes column, click the active **Add** field.

The Interface Attributes window appears.

Note Because the policy created for this feature enables ETTH Support and Shared VLAN, these attributes become unavailable at the link level.

Step 8 Enter a valid **VLAN ID** value, then click **Next**. The IP Address Scheme window appears.

Step 9 Enter valid values for each required field, then click **Next**.

Step 10 In the Routing Information window, check any applicable items, then click **Next**.

Note For information about protocol types, see Specifying the Routing Protocol for a Service, page 5-48.

Step 11 In the VRF and VPN window, for Maximum Route Threshold (required field), accept the default value, or enter a new value.

Note If you want to set the VRF and VPN attributes via a previously defined VRF object, check the **Use VRF Object** check box. For more information on this feature, see Independent VRF Management, page 5-14. That section describes how to use independent VRF objects in MPLS VPN service policies and service requests.

Note For more information on setting the VRF and VPN attributes in MPLS VPN service requests, see Defining VRF and VPN Attributes in an MPLS Service Request, page 5-85.

Step 12 Under VPN Selection (required), click **Add**.

Step 13 From the CERC window, choose the desired PE VPN Membership, then click **Done**.

Step 14 Back in the VRF and VPN window, click **Finish**.

Note If the policy on which the service request is based has template association enabled, a **Next** button is visible in the GUI. Click the **Next** button to add templates and data files to the devices defined in the service request. For instructions about associating templates with service requests, see Chapter 9, "Managing Templates and Data Files."

When you are finished setting the attributes for the service policy, the MPLS Service Request Editor window appears.

Step 15 Click **Save**.

The MPLS Service Requests window reappears showing that the service request is in the Requested state and ready to deploy.

Spanning Multiple Autonomous Systems

This section describes how to configure spanning multiple autonomous systems using the Prime Fulfillment provisioning process.

Overview

The inter-autonomous system for MPLS VPNs feature allows an MPLS VPN to span service providers and autonomous systems. An autonomous system is a single network or group of networks that is controlled by a common system administration group and that uses a single, clearly defined routing protocol.

As VPNs grow, their requirements expand. In some cases, VPNs need to reside on different autonomous systems in different geographic areas. Also, some VPNs need to extend across multiple service providers (overlapping VPNs). Regardless of the complexity and location of the VPNs, the connection between autonomous systems must be seamless to the customer.

The inter-autonomous systems for MPLS VPNs feature provides that seamless integration of autonomous systems and service providers. Separate autonomous systems from different service providers can communicate by exchanging IPv4 network layer reachability information (NLRI) in the form of VPN-IPv4 addresses. The autonomous systems' border edge routers use the Exterior Border Gateway Protocol (eBGP) to exchange that information. An Interior Gateway Protocol (IGP) then distributes the network layer information for VPN-IPv4 prefixes throughout each VPN and each autonomous system. Routing information uses the following protocols:

- Within an autonomous system, routing information is shared using an IGP.
- Between autonomous systems, routing information is shared using an eBGP. An eBGP allows a service provider to set up an inter-domain routing system that guarantees the loop-free exchange of routing information between separate autonomous systems.

An MPLS VPN with inter-autonomous system support allows a service provider to provide to customers scalable Layer 3 VPN services, such as web hosting, application hosting, interactive learning, electronic commerce, and telephony service. A VPN service provider supplies a secure, IP-based network that shares resources on one or more physical networks.

The primary function of eBGP is to exchange network reachability information between autonomous systems, including information about the list of autonomous system routes. The autonomous systems use EGBP border edge routers to distribute the routes, which include label switching information. Each border edge router rewrites the next-hop and MPLS labels. See Routing Between Autonomous Systems, page 5-155 for more information.

Inter-autonomous system configurations supported in an MPLS VPN can include:

- *Interprovider VPN*: MPLS VPNs that include two or more autonomous systems, connected by separate border edge routers. The autonomous systems exchange routes using eBGP. No Interior Gateway Protocol (IGP) or routing information is exchanged between the autonomous systems.
- *BGP Confederations*: MPLS VPNs that divide a single autonomous system into multiple subautonomous systems, and classify them as a single, designated confederation. The network recognizes the confederation as a single autonomous system. The peers in the different autonomous systems communicate over eBGP sessions; however, they can exchange route information as if they were iBGP peers.

Benefits

The inter-autonomous system MPLS VPN feature provides the following benefits:

- Allows a VPN to cross more than one service provider backbone

  The inter-autonomous systems for MPLS VPNs feature allows service providers, running separate autonomous systems, to jointly offer MPLS VPN services to the same end customer. A VPN can begin at one customer site and traverse different VPN service provider backbones before arriving at

another site of the same customer. Previously, MPLS VPNs could only traverse a single BGP autonomous system service provider backbone. The inter-autonomous system feature allows multiple autonomous systems to form a continuous (and seamless) network between a service provider's customer sites.

- Allows a VPN to exist in different areas

  The inter-autonomous systems for MPLS VPNs feature allows a service provider to create a VPN in different geographic areas. Having all VPN traffic flow through one point (between the areas) allows for better rate control of network traffic between the areas.

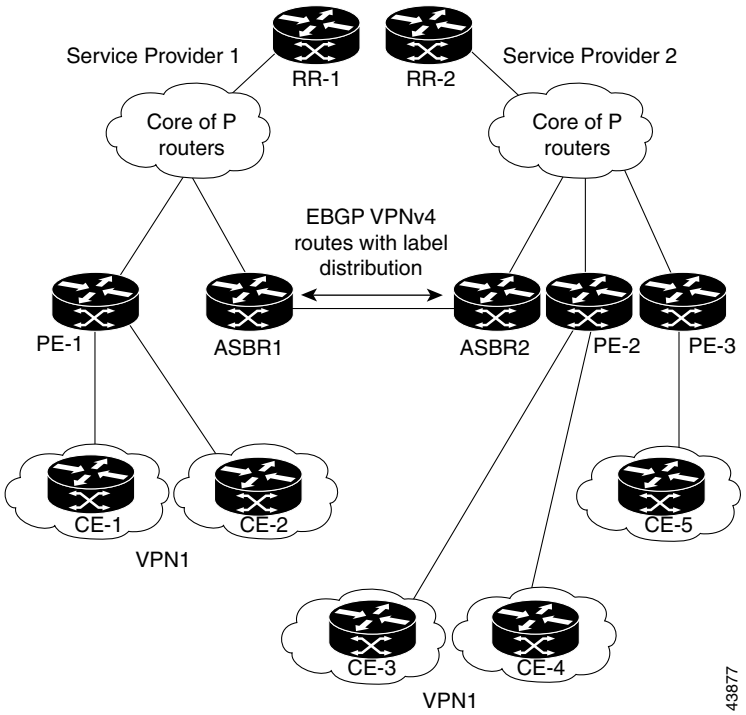
- Allows confederations to optimize iBGP meshing

  The inter-autonomous systems feature can make iBGP meshing in an autonomous system more organized and manageable. You can divide an autonomous system into multiple, separate subautonomous systems and then classify them into a single confederation (even though the entire VPN backbone appears as a single autonomous system). This capability allows a service provider to offer MPLS VPNs across the confederation because it supports the exchange of labeled VPN-IPv4 network layer reachability information between the subautonomous systems that form the confederation.

Routing Between Autonomous Systems

Figure 5-31 illustrates one MPLS VPN consisting of two separate autonomous systems. Each autonomous system operates under different administrative control and runs a different IGP. Service providers exchange routing information through eBGP border edge routers (ASBR1 and ASBR2).

Figure 5-31 eBGP Connection Between Two Autonomous Systems



This configuration uses the following process to transmit information:

1. The provider edge router (PE-1) assigns a label for a route before distributing that route. The PE router uses the multiprotocol extensions of a Border Gateway Protocol (BGP) to transmit label mapping information. The PE router distributes the route as a VPN-IPv4 address. The address label and the VPN identifier are encoded as part of the NLRI.
2. The two route reflectors (RR-1 and RR-2) reflect VPN-IPv4 internal routes within the autonomous system. The autonomous systems' border edge routers (ASBR1 and ASBR2) advertise the VPN-IPv4 external routes.
3. The eBGP border edge router (ASBR1) redistributes the route to the next autonomous system, (ASBR2). ASBR1 specifies its own address as the value of the eBGP next hop attribute and assigns a new label. The ASBR1 address ensures the following:
   - The next hop router is always reachable in the service provider (P) backbone network.
   - The label assigned by the distributing router is properly interpreted. The label associated with a route must be assigned by the corresponding next hop router.
4. The eBGP border edge router (ASBR2) redistributes the route in one of the following ways, depending on its configuration:
   - If the iBGP neighbors are configured with the **neighbor next-hop-self** command, ASBR2 changes the next hop address of updates received from the eBGP peer, then forwards it on.
   - If the iBGP neighbors are not configured with the **neighbor next-hop-self** command, the next hop address does not get changed. ASBR2 must propagate a host route for the eBGP peer through the IGP.

     To propagate the eBGP VPN-IPv4 neighbor host route, use the **redistribute connected subnets** command. The eBGP VPN-IPv4 neighbor host route is automatically installed in the routing table when the neighbor comes up. This is essential to establish the label-switched path between PE routers in different autonomous systems.

Exchanging VPN Routing Information

Autonomous systems exchange VPN routing information (routes and labels) to establish connections. To control connections between autonomous systems, the PE routers and eBGP border edge routers maintain a Label Forwarding Information Base (LFIB). The LFIB manages the labels and routes that the PE routers and eBGP border edge routers receive during the exchange of VPN information.

Figure 5-32 illustrates the exchange of VPN route and label information between autonomous systems. The autonomous systems use the following guidelines to exchange VPN routing information:

Routing information includes:

- The destination network (N)
- The next hop field associated with the distributing router
- A local MPLS label (L)

An *RD1: route distinguisher* is part of a destination network address to make the VPN-IPv4 route globally unique in the VPN service provider environment.

The *ASBRs* are configured to change the next hop (next-hop-self) when sending VPN-IPv4 NLRIs to the iBGP neighbors. Therefore, the ASBRs must allocate a new label when they forward the NLRI to the iBGP neighbors.

Figure 5-32 Exchanging Routes and Labels Between Two Autonomous Systems

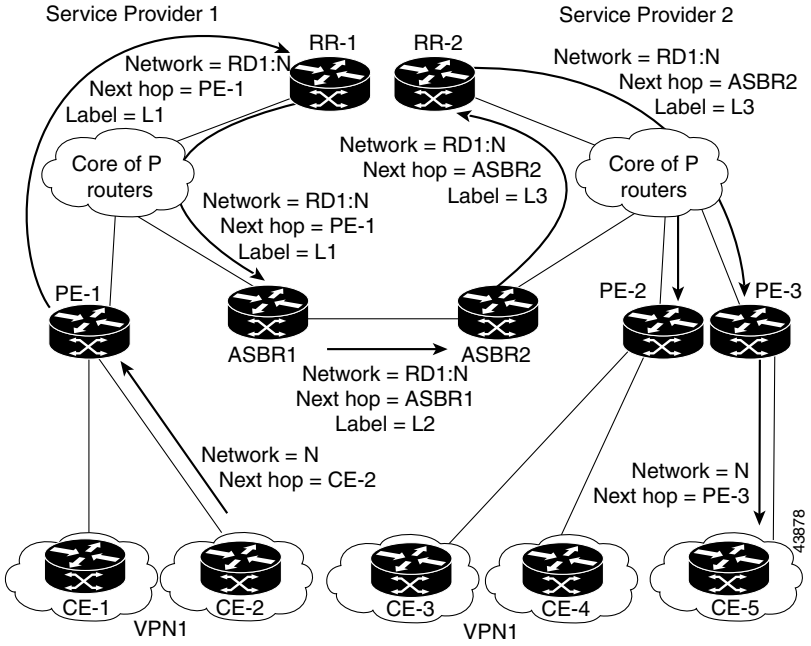


Figure 5-33 illustrates the exchange of VPN route and label information between autonomous systems. The only difference is that ASBR2 is configured with the **redistribute connected** command, which propagates the host routes to all PEs. The **redistribute connected** command is necessary because ASBR2 is not the configured to change the next hop address.

Figure 5-33 Host Routes Propagated to All PEs Between Two Autonomous Systems

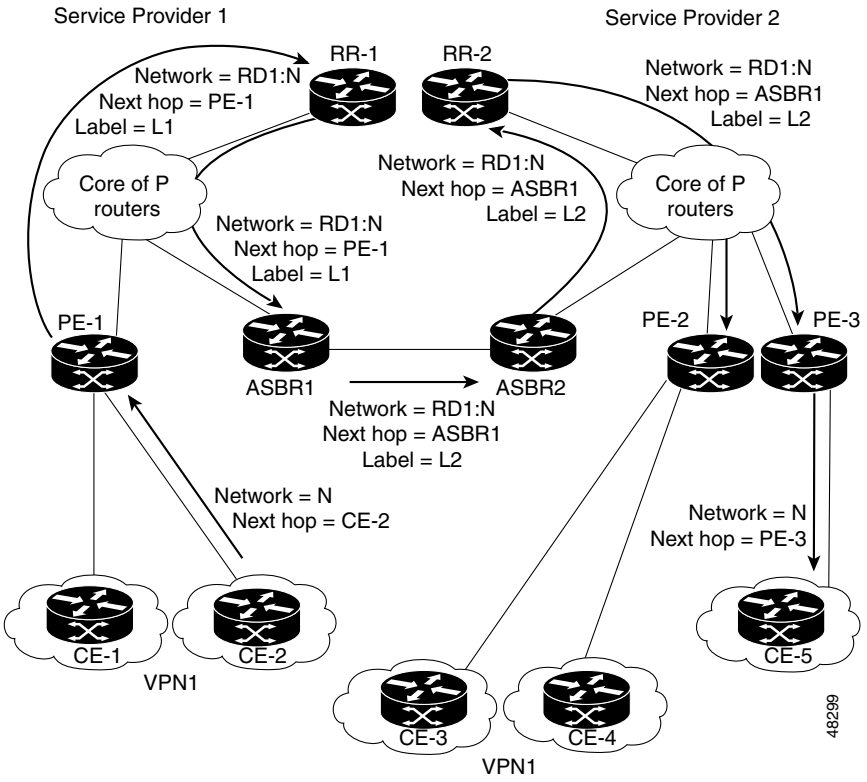


Figure 5-34 illustrates how packets are forwarded between autonomous systems in an interprovider network using the following packet forwarding method:

Packets are forwarded to their destination via MPLS. Packets use the routing information stored in the LFIB of each PE router and eBGP border edge router. The service provider VPN backbone uses dynamic label switching to forward labels.

Each autonomous system uses standard multi-level labeling to forward packets between the edges of the autonomous system routers (for example, from CE-5 to PE-3). Between autonomous systems, only a single level of labeling is used, corresponding to the advertised route.

A data packet carries two levels of labels when traversing the VPN backbone:

- The first label (*IGP route label*) directs the packet to the correct PE router or eBGP border edge router. (For example, the IGP label of ASBR2 points to the ASBR2 border edge router.)
- The second label (*VPN route label*) directs the packet to the appropriate PE router or eBGP border edge router.

Figure 5-34 Forwarding Packets Between Two Autonomous Systems

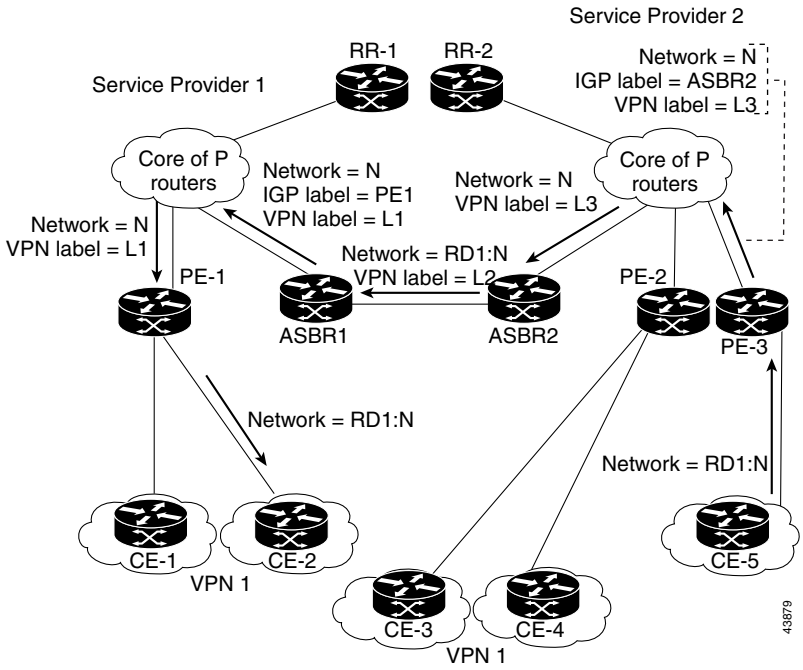
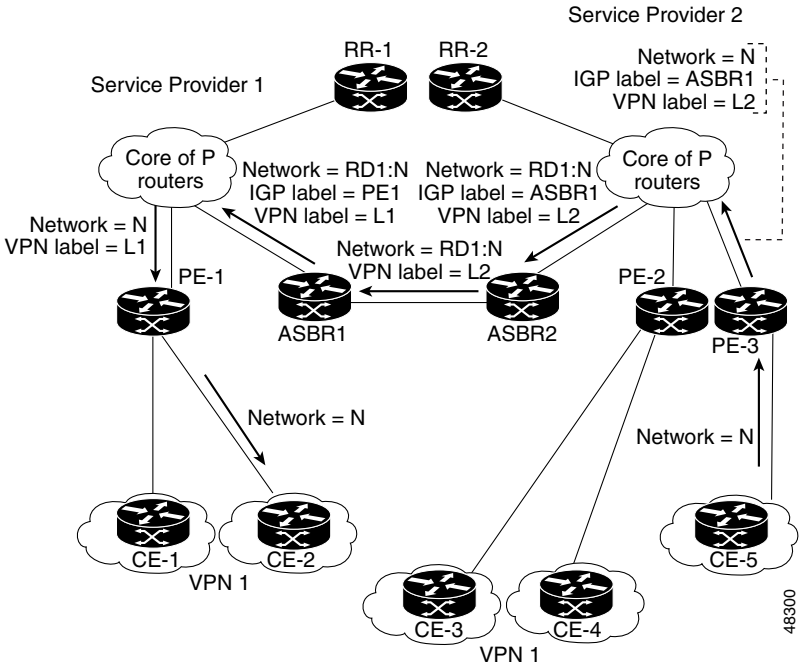


Figure 5-35 illustrates shows the same packet forwarding method, except the eBGP router (ASBR1) forwards the packet without reassigning it a new label.

Figure 5-35 Forwarding Packets Without Reassigning a New Label



Routing Between Subautonomous Systems in a Confederation

A VPN can span service providers running in separate autonomous systems or between multiple subautonomous systems that have been grouped together to form a confederation.

A confederation reduces the total number of peer devices in an autonomous system. A confederation divides an autonomous system into subautonomous systems and assigns a confederation identifier to the autonomous systems.

In a confederation, each subautonomous system is fully meshed with other subautonomous systems. The subautonomous systems communicate using an IGP, such as Open Shortest Path First (OSPF) or Intermediate System-to-Intermediate System (IS-IS). Each subautonomous system also has an eBGP connection to the other subautonomous systems. The confederation eBGP (CeBGP) border edge routers forward next-hop-self addresses between the specified subautonomous systems. The next-hop-self address forces the BGP to use a specified address as the next hop rather than letting the protocol choose the next hop.

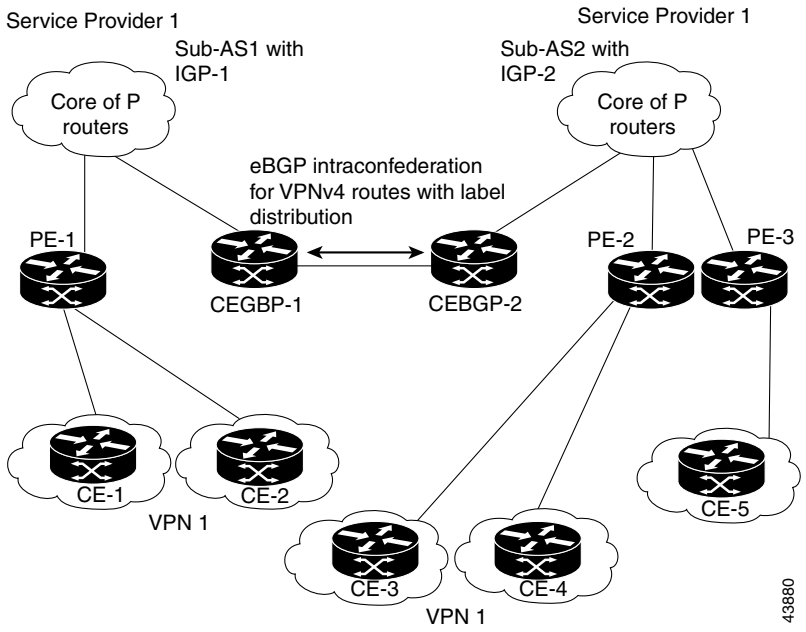
You can configure a confederation with separate subautonomous systems in two ways:

- You can configure a router to forward next-hop-self addresses between only the CeGRP border edge routers (both directions). The subautonomous systems (iBGP peers) at the subautonomous system border do not forward the next-hop-self address. Each subautonomous system runs as a single IGP domain. However, the CeGRP border edge router addresses are known in the IGP domains.
- You can configure a router to forward next-hop-self addresses between the CeGRP border edge routers (both directions) and within the iBGP peers at the subautonomous system border. Each subautonomous system runs as a single IGP domain but also forwards next-hop-self addresses between the PE routers in the domain. The CeGRP border edge router addresses are known in the IGP domains.

Figure 5-36 illustrates a typical MPLS VPN confederation configuration. In this confederation configuration:

- The two CeGRP border edge routers exchange VPN-IPv4 addresses with labels between the two subautonomous systems.
- The distributing router changes the next-hop addresses and labels and uses a next-hop-self address.
- IGP-1 and IGP-2 know the addresses of CEGRP-1 and CEBGP-2.

Figure 5-36 EGBP Connection Between Two AS's in a Confederation



In this confederation configuration:

- CeGRP border edge routers function as neighboring peers between the subautonomous systems. The subautonomous systems use eGRP to exchange route information.
- Each CeGRP border edge router (CEBGP-1, CEBGP-2) assigns a label for the route before distributing the route to the next subautonomous system. The CeGRP border edge router distributes the route as a VPN-IPv4 address by using the multiprotocol extensions of BGP. The label and the VPN identifier are encoded as part of the NLRI.
- Each PE and CeGRP border edge router assigns its own label to each VPN-IPv4 address prefix before redistributing the routes. The CeGRP border edge routers exchange VPN-IPv4 addresses with the labels.

  The next-hop-self address is included in the label (as the value of the eGRP next-hop attribute). Within the subautonomous systems, the CeGRP border edge router address is distributed throughout the iBGP neighbors and the two CeGRP border edge routers are known to both confederations.

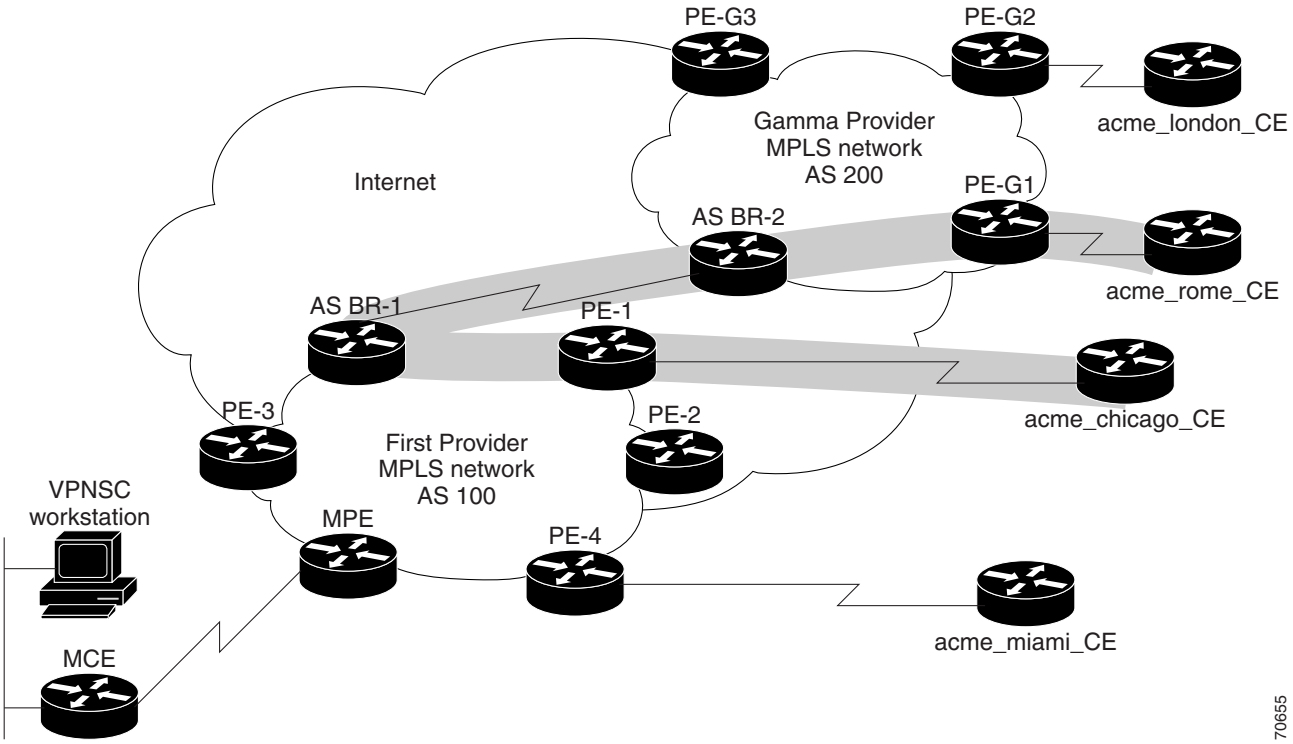
Using Prime Fulfillment to Span Multiple Autonomous Systems

As described in Exchanging VPN Routing Information, page 5-156, autonomous systems exchange VPN routing information (routes and labels) to establish connections. To control connections between autonomous systems, the PE routers and Exterior BGP ASBRs (Autonomous System Boundary Routers) maintain a Label Forwarding Information Base (LFIB). The LFIB manages the labels and routes that the PE routers and eGRP border edge routers receive during the exchange of VPN information.

The *ASBRs* are configured to change the next hop (next-hop-self) when sending VPN-IPv4 network layer reachability information to their iBGP neighbors. Therefore, the ASBRs must allocate a new label when they forward the NLRI to their iBGP neighbors.

Figure 5-37 shows the example Prime Fulfillment network used in this section.

Figure 5-37 Example VPN Network with Two Autonomous Systems



In order for traffic from Acme_Chicago in AS 100 to reach Acme_Rome in AS 200, Prime Fulfillment must provision two links only:

- The link between Acme_Chicago and PE-1
- The link between Acme_Rome and PE-G1

As shown in Figure 5-37, Prime Fulfillment routes the VPN traffic from PE-1 to ASBR-1, from ASBR-1 to ASBR-2, then from ASBR-2 to PE-G1; finally the traffic is routed to its destination, Acme-Rome.

ASBR-1 and ASBR-2 must run BGP (Border Gateway Protocol). Then iMP-BGP (interior Multiprotocol BGP) handles the routes between PE-1 to ASBR-1 in AS 100 and the routes between PE-2 to ASBR-2 in AS 200. eMP-BGP (exterior Multiprotocol BGP) handles the routes between ASBR-1 and ASBR-2.

Tip The service provider must configure a VPN-IPv4 eGRP session between directly connected Autonomous System Boundary Routers (ASBRs). This is a one-time setup procedure that the service provider must manage. Prime Fulfillment does not provision the link between the ASBR devices that span autonomous systems.

A VPN-IPv4 address (also referred to as a *VPNv4* address) is the combination of the IPv4 address and the 8-byte route distinguisher (RD). Combining the RD and the IPv4 address makes the IPv4 route globally unique across the MPLS VPN network. BGP considers an IPv4 address as different from another IPv4 address that has the same network and subnet mask when the route distinguishers are different.

Using Templates to Support Inter-Autonomous System Solutions

This section covers how Prime Fulfillment supports inter-autonomous system (inter-AS) and inter-provider VPNs through Prime Fulfillment templates.

Note Prime Fulfillment currently supports only the inter-AS 10B Hybrid model for L2TPV3 networks. This is the solution documented in the this section.

Inter-AS 10B Hybrid Model

The current release of Prime Fulfillment provides two pairs of template scripts for provisioning and decommissioning inter-AS 10B Hybrid VPNs:

- Provisioning and decommissioning VPN-independent inter-AS 10B Hybrid CLIs on an Autonomous System Border Router (ASBR)
- Provisioning and decommissioning VPN-specific inter-AS 10B Hybrid CLIs on an ASBR

Using the second pair of template scripts, the provider can create a new pair of data-files for provisioning and decommissioning a new inter-AS VPN on the ASBR, as and when added.The default inter-AS scripts can be modified to create or change scripts for modifying inter-AS configuration.

The following commands are supported in the VPN-independent inter-AS 10B Hybrid default templates:

- Provisioning resolve in VRF (RiV) VRF for L2TPV3 tunnel on an ASBR
- L2TPV3 tunnel configuration
- ASBR-facing interface provisioning
- BGP configuration:
  - BGP configuration with a **peer-group**
  - eBGP configuration
  - BGP **address-family ipv4** configuration
  - BGP **address-family ipv4 tunnel** configuration
  - BGP **address-family vpnv4** configuration
- Default route configuration through an L2TPV3 tunnel interface

The following commands are supported in the VPN-specific inter-AS 10B Hybrid default templates:

- Provisioning VRF for a customer VPN
- Recommended/standard route target (RT) support for full-mesh and hub-and-spoke VPN types. Spoke RTs are optional.
- RT-rewrite configuration:
  - Extended community (**extcommunity-list**) provisioning
  - Route maps provisioning

Inter-AS RT-Rewrite

Prime Fulfillment supports inter-AS RT-rewrite configuration on the ASBR. Velocity Template Language (VTL) template scripts for provisioning and decommissioning of RT-rewrite commands are provided as part of the inter-AS 10B hybrid templates, covered in the next section. You can edit these VTL scripts to create your own templates for the respective use-case.

Creating the Inter-AS Templates

Note For additional coverage of creating and using templates in Prime Fulfillment, see Chapter 9, "Managing Templates and Data Files."

The default inter-AS templates are provided in the Examples templates directory in Prime Fulfillment. The templates are created from the Service Design window, which you access by choosing:

Service Design > Templates > Examples

The templates for Inter-AS 10b hybrid are:

- Configure_PE_as_ASBR_non_VPN_Specific_Template_TMPL_
- Remove_PE_as_ASBR_non_VPN_Specific_Template_TMPL_
- Configure_PE_as_ASBR_VPN_Specific_Template_TMPL_
- Remove_PE_as_ASBR_VPN_Specific_Template_TMPL_

You can create and change templates, using the default provisioning and decommissioning scripts, based on the respective use-case. Because the inter-AS configurations are mostly a one time setup, the templates are downloaded from the device console only, but are not attached to a service request.

The Prime Fulfillment templates feature supports a basic deployment check to determine whether the template data file was successfully deployed or whether there was any command that failed to deploy. In addition, you can select the data-type for the variables, which facilitates entering the right values during data-file creation in the user interface.

After you successfully create the template data file that contains the inter-AS CLIs, you can download the template data file onto the ASBR or route reflector using the Prime Fulfillment Device Console window, which you access by choosing:

Service Inventory > Device Console

The templates you created under Service Design can be selected for deployment on a device or a device-group.

Note The Prime Fulfillment templates feature is not model-based, so no template deployment history or stack is saved, no template roll-back is supported, and no template CLI audit is supported when you download the templates using the Device Console. You can also select templates in a service request, and have them downloaded onto the PE routers, in case you need to download specific iBGP commands on the PE routers.

Sample Configlets

This section provides sample configlets for MPLS VPN provisioning in Prime Fulfillment. It contains the following sections:

Overview

The configlets provided in this section show the CLIs generated by Prime Fulfillment for particular services and features. Each configlet example provides the following information:

- Service.
- Feature.

- Devices configuration (network role, hardware platform, relationship of the devices and other relevant information).
- Sample configlets for each device in the configuration.
- Comments.

Note The configlets generated by Prime Fulfillment are only the delta between what needs to be provisioned and what currently exists on the device. This means that if a relevant CLI is already on the device, it does not show up in the associated configlet.

Note All examples in this appendix assume an MPLS core.

For information on how to view configlets, see Viewing Service Request Configlets, page 8-6.

L2 Access into L3 MPLS VPN

Configuration

- Service: L2VPN/Metro Ethernet.
- Feature: Access into L3 MPLS VPN.
- Device configuration:
  - The CE is a Cisco 3550 with IOS 12.1(22)EA1.

    Interface(s): F0/13 <-> F0/4.
  - The U-PE is a Cisco 3550 with IOS 12.1(22)EA1.

    Interface(s): F0/14.
  - The N-PE is a Cisco 7609 with IOS 12.2(18)SXF.

    Interface(s): F2/8.
  - VLAN = 3101.

Configlets

CE

```
!
vlan 3101
exit
!
interface FastEthernet0/13
no ip address
switchport
switchport trunk
encapsulation dot1q
switchport mode trunk
switchport trunk allowed
vlan 1,3101
!
interface Vlan3101
description By VPNSC: Job
Id# = 13
ip address 10.19.19.10
255.255.255.252
no shutdown
```

U-PE

```
!
vlan 3101
exit
!
interface FastEthernet0/14
no ip address
switchport
switchport trunk encapsulation
dot1q
switchport mode trunk
switchport trunk allowed vlan
1,3101
!
interface FastEthernet0/4
no keepalive
no ip address
switchport
switchport trunk encapsulation
dot1q
switchport mode trunk
switchport trunk allowed vlan
3101
switchport nonegotiate
cdp enable
no shutdown
mac access-group
ISC-FastEthernet0/4 in
!
mac access-list extended
ISC-FastEthernet0/4
deny any host 0100.0ccc.cccc
deny any host 0100.0ccc.cccd
deny any host 0100.0ccd.cdd0
deny any host 0180.c200.0000
permit any any
```

N-PE

```
!
ip vrf V5:VPN_sample
rd 100:1502
route-target import
100:1602
route-target import
100:1603
route-target export
100:1602
maximum routes 100 80
!
interface FastEthernet2/8
no shutdown
!
interface
FastEthernet2/8.3101
description
FastEthernet2/8.3101 dot1q
vlan id=3101. By VPNSC:
Job Id# = 13
encapsulation dot1Q 3101
ip vrf forwarding
V5:VPN_sample
ip address 10.19.19.9
255.255.255.252
no shutdown
!
router bgp 100
address-family ipv4 vrf
V5:VPN_sample
redistribute connected
redistribute static
exit-address-family
```

Comments

- IP Numbered scenario with Dot1q encapsulation for VPN Link.
- The VRF is created on the N-PE device (-s designates that the VRF is joining the VPN as a spoke in a hub-n-spoke topology.
- On the N-PE, the VRF is added to iBGP routing instance with user configured redistribution of connected and static options.
- The VRF is created on the NPE with forwarding associated with the U-PE facing interface.

CE-PE L3 MPLS VPN (BGP with full-mesh)

Configuration

- Service: L3 MPLS VPN.
- Feature: CE-PE BGP with full-mesh.
- Device configuration:
  - The PE is a Cisco 7609 with IOS 12.2(18)SXF.

    Interface(s): F2/5.
  - The CE is a Cisco 3550 with IOS 12.2(22)EA1.

    Interface(s): F0/13.
  - Routing protocol = BGP.

Configlets

CE	PE
!	!
vlan 62	ip vrf V9:mpls_vpn1
exit	rd 100:1506
!	route-target import 99:3204
interface FastEthernet0/13	route-target export 99:3204
no ip address	maximum routes 100 80
switchport	!
switchport trunk encapsulation dot1q	interface FastEthernet2/5.62
switchport mode trunk	description FastEthernet2/5.62 dot1q vlan
switchport trunk allowed vlan 62	id=62. By VPNSC: Job Id# = 29
!	encapsulation dot1Q 62
interface Vlan62	ip vrf forwarding V9:mpls_vpn1
description By VPNSC: Job Id# = 29	ip address 10.19.19.41 255.255.255.252
ip address 10.19.19.42 255.255.255.252	no shutdown
no shutdown	!
!	router bgp 100
router bgp 10	address-family ipv4 vrf V9:mpls_vpn1
neighbor 10.19.19.41 remote-as 100	neighbor 10.19.19.42 remote-as 10
	neighbor 10.19.19.42 activate
	neighbor 10.19.19.42 allowas-in 2
	redistribute connected
	redistribute static
	exit-address-family

Comments

- A full-mesh configuration is created by means of the CERC selected for the VPN policy. As a result, route-target import and route-target export are identical.
- BGP is the routing protocol on the CE-PE access link.
- IP Numbered scenario with dot1q encapsulation for the VPN link.
- The VRF is created on the PE device.
- The VRF is created on the PE with forwarding associated with the CE facing interface.

CE-PE L3 MPLS VPN (BGP with SOO)

Configuration

- Service: L3 MPLS VPN.
- Feature: CE-PE.
- Device configuration:
  - The PE is a Cisco 7609 with IOS 12.2(18)SXF.

    Interface(s): FE2/3.
  - The CE created in Prime Fulfillment.

    Interface(s): FE1/0/14.
  - Routing protocol = BGP.
  - VPN = hub.

Configlets

CE

```
!
vlan 3100
exit
!
interface FastEthernet1/0/14
no ip address
switchport
switchport trunk encapsulation dot1q
switchport mode trunk
switchport trunk allowed vlan 1,3100
no shutdown
!
interface Vlan3100
description By VPNSC: Job Id# = 12
ip address 10.19.19.6 255.255.255.252
no shutdown
!
router ospf 3500
network 10.19.19.4 0.0.0.3 area 12345
```

PE

```
!
ip vrf V4:VPN_sample-s
rd 100:1501
route-target import 100:1602
route-target export 100:1603
maximum routes 100 80
!
interface FastEthernet2/3.3100
description FastEthernet2/3.3100 dot1q vlan
id=3100. By VPNSC: Job Id# = 12
encapsulation dot1Q 3100
ip vrf forwarding V4:VPN_sample-s
ip address 10.19.19.5 255.255.255.252
no shutdown
!
router ospf 2500 vrf V4:VPN_sample-s
redistribute bgp 100 subnets
network 10.19.19.4 0.0.0.3 area 12345
!
router bgp 100
address-family ipv4 vrf V4:VPN_sample-s
redistribute connected
redistribute ospf 2500 vrf V4:VPN_sample-s
match internal external 1 external 2
redistribute static
exit-address-family
```

Comments

- IP Numbered scenario with dot1q encapsulation for the VPN link.
- The VRF is created on PE device (VPN is joining as a spoke).
- On PE, the VRF is added to iBGP routing instance with user configured redistribution of connected and static options.
- The VRF is created on the PE with forwarding associated with the CE-facing interface.

- This example is for an IOS device. Site-of-origin (SOO) is also supported for IOS XR devices. In the case of an IOS XR device, the resulting configlet is different. For an IOS XR device, the configlet generated for SOO would be of the form **site-of-origin 64512:500**.

CE-PE L3 MPLS VPN

Configuration

- Service: L3 MPLS VPN.
- Feature: CE-PE.
- Device configuration:
  - The PE is a Cisco 7603 with IOS 12.2(18)SXD7.

    Interface(s): FE2/25.
  - The CE is an Cisco 3750ME-I5-M with IOS 12.2(25)EY2.

    Interface(s): FE1/0/6.
  - VPN = spoke.

Configlets

CE	PE
! vlan 890 exit ! interface FastEthernet1/0/6 no ip address switchport trunk encapsulation dot1q switchport mode trunk switchport trunk allowed vlan 890 no shutdown ! interface Vlan890 description By VPNSC: Job Id# = 336 : SR Id# = 336 ip address 10.10.75.2 255.255.255.252 no shutdown ! router bgp 120 neighbor 10.10.75.1 remote-as 100 no auto-summary	! ip vrf V60:TestVPN-s rd 100:8069 route-target import 100:1891 route-target export 100:1892 ! interface FastEthernet2/25.890 description FastEthernet2/25.890 dot1q vlan id=890. By VPNSC: Job Id# = 336 : SR Id# = 336 encapsulation dot1Q 890 ip vrf forwarding V60:TestVPN-s ip address 10.10.75.1 255.255.255.252 no shutdown ! router bgp 100 no auto-summary address-family ipv4 vrf V60:TestVPN-s neighbor 10.10.75.2 remote-as 120 neighbor 10.10.75.2 activate neighbor 10.10.75.2 route-map SetSOO_V60:TestVPN-s_100:100 in exit-address-family ! route-map SetSOO_V60:TestVPN-s_100:100 permit 10 set extcommunity soo 100:100

Comments

- IP Numbered scenario with dot1q encapsulation for the VPN link.
- The VRF is created on the PE device.
- `neighbor 10.10.75.2 remote-as 120` is created as a result of the policy having the CE BGP AS ID set to 120.
- The VRF is created on the PE with forwarding associated with the CE-facing interface.
- On the PE, BGP defines a route-map for the CE neighbor.
- The associated route map sets the extended community attribute to SOO, which is the community value (SOO pool value defined in Prime Fulfillment).
- This example is for an IOS device. Site-of-origin (SOO) is also supported for IOS XR devices. In the case of an IOS XR device, the resulting configlet is different. For an IOS XR device, the configlet generated for SOO would be of the form **site-of-origin 64512:500**.

N-PE L3 MPLS VPN (IPv4, IOS XR, OSPF)

Configuration

- Service: L3 MPLS VPN.
- Feature: IPv4 with IOS XR.
- Device configuration:
  - The N-PE is a Cisco 12000 router with IOS XR.
  - Routing protocol = OSPF.

Configlets

N-PE

(See the extended code example below.)

```
<?xml version="1.0" encoding="UTF-8"?>
<Request MajorVersion="1" MinorVersion="0">
  <Delete>
    <Configuration Source="CurrentConfig">
      <InterfaceConfigurationTable>
        <InterfaceConfiguration>
          <Naming>
            <Name>GigabitEthernet0/1/1/1.856</Name>
            <Active>act</Active>
          </Naming>
          <Shutdown>true</Shutdown>
        </InterfaceConfiguration>
      </InterfaceConfigurationTable>
    </Configuration>
  </Delete>
  <Set>
    <Configuration Source="CurrentConfig">
      <VRFTable>
        <VRF>
          <Naming>
            <Name>ICICI_VPN_1</Name>
          </Naming>
          <AFI_SAFITable>
            <AFI_SAFI>
              <Naming>
                <AFI>IPv4</AFI>
                <SAFI>Unicast</SAFI>
              </Naming>
              <BGP>
                <ImportRouteTargets>
                  <RouteTargetTable>
                    <RouteTarget>
                      <Naming>
                        <Type>AS</Type>
                        <AS>100</AS>
                        <ASIndex>1</ASIndex>
                      </Naming>
                      <True>true</True>
                    </RouteTarget>
                  </RouteTargetTable>
                </ImportRouteTargets>
                <ExportRouteTargets>
```

```
                        <RouteTargetTable>
                          <RouteTarget>
                            <Naming>
                              <Type>AS</Type>
                              <AS>100</AS>
                              <ASIndex>1</ASIndex>
                            </Naming>
                            <True>true</True>
                          </RouteTarget>
                        </RouteTargetTable>
                      </ExportRouteTargets>
                    </BGP>
                  </AFI_SAFI>
                </AFI_SAFITable>
              </VRF>
            </VRFTable>
            <InterfaceConfigurationTable>
              <InterfaceConfiguration>
                <Naming>
                  <Name>GigabitEthernet0/1/1/1.856</Name>
                  <Active>act</Active>
                </Naming>
                <Description>GigabitEthernet0/1/1/1.856 dot1q vlan id=856. By VPNSC: Job Id# =
116</Description>
                <InterfaceModeNonPhysical>Default</InterfaceModeNonPhysical>
                <VLANSubConfiguration>
                  <VLANIdentifier>
                    <VlanType>VLANTypeDot1q</VlanType>
                    <FirstTag>856</FirstTag>
                  </VLANIdentifier>
                </VLANSubConfiguration>
                <VRF>ICICI_VPN_1</VRF>
                <IPV4Network>
                  <Addresses>
                    <Primary>
                      <IPAddress>10.10.56.1</IPAddress>
                      <Mask>255.255.255.252</Mask>
                    </Primary>
                  </Addresses>
                </IPV4Network>
              </InterfaceConfiguration>
            </InterfaceConfigurationTable>
            <BGP>
              <AS>
                <Naming>
                  <AS>0</AS>
                </Naming>
                <FourByteAS>
                  <Naming>
                    <AS>100</AS>
                  </Naming>
                  <VRFTable>
                    <VRF>
                      <Naming>
                        <Name>ICICI_VPN_1</Name>
                      </Naming>
                      <VRFGlobal>
                        <Exists>true</Exists>
                        <RouteDistinguisher>
                          <Type>AS</Type>
                          <AS>100</AS>
                          <ASIndex>8064</ASIndex>
                        </RouteDistinguisher>
                        <VRFGlobalAFTable>
```

```
                    <VRFGlobalAF>
                      <Naming>
                        <AF>IPv4Unicast</AF>
                      </Naming>
                      <Enabled>true</Enabled>
                      <Redistribution>
                        <ConnectedRoutes/>
                        <OSPFRouteTable>
                          <OSPFRoutes>
                            <Naming>
                              <OSPFInstanceName>100</OSPFInstanceName>
                            </Naming>
                            <RedistType>21</RedistType>
                            <DefaultMetric>20000</DefaultMetric>
                          </OSPFRoutes>
                        </OSPFRouteTable>
                        <StaticRoutes/>
                      </Redistribution>
                    </VRFGlobalAF>
                  </VRFGlobalAFTable>
                </VRFGlobal>
              </VRF>
            </VRFTable>
          </FourByteAS>
        </AS>
      </BGP>
      <OSPF>
        <ProcessTable>
          <Process>
            <Naming>
              <InstanceName>100</InstanceName>
            </Naming>
            <Start>true</Start>
            <VRFTable>
              <VRF>
                <Naming>
                  <VRFName>ICICI_VPN_1</VRFName>
                </Naming>
                <VRFStart>true</VRFStart>
                <Redistribution>
                  <RedistributeTable>
                    <Redistribute>
                      <Naming>
                        <ProtocolType>rip</ProtocolType>
                        <InstanceName>rip</InstanceName>
                      </Naming>
                      <Classful>false</Classful>
                    </Redistribute>
                    <Redistribute>
                      <Naming>
                        <ProtocolType>static</ProtocolType>
                        <InstanceName>static</InstanceName>
                      </Naming>
                      <Classful>false</Classful>
                    </Redistribute>
                  </RedistributeTable>
                </Redistribution>
                <AreaTable>
                  <Area>
                    <Naming>
                      <IntegerID>100</IntegerID>
                    </Naming>
                    <NameScopeTable>
                      <NameScope>
```

```
                        <Naming>
                          <Interface>GigabitEthernet0/1/1/1.856</Interface>
                        </Naming>
                        <Running>true</Running>
                      </NameScope>
                    </NameScopeTable>
                    <Running>true</Running>
                  </Area>
                </AreaTable>
                <DefaultInformation>
                  <AlwaysAdvertise>true</AlwaysAdvertise>
                </DefaultInformation>
              </VRF>
            </VRFTable>
          </Process>
        </ProcessTable>
      </OSPF>
    </Configuration>
  </Set>
  <Commit/>
</Request>
```

Comments

- In IOS XR, device configuration is specified in XML format.
- With respect to the XML schemas, different versions of IOS XR generates different XML configlets. However the configurations are almost identical, except for changes in the XML schema.
- There are different cases to consider. For example, when a service request is decommissioned or modified, the XML configurations are slightly different.

N-PE L3 MPLS VPN (IPv6, IOS XR, EIGRP)

Configuration

- Service: L3 MPLS VPN.
- Feature: N-PE running IOS XR 3.5.x.
- Device configuration:
  - The N-PE is a Cisco 12000 router with IOS XR 3.5.x.
  - Routing protocol = EIGRP.

Configlets

N-PE

(See the extended code example below.)

```
<?xml version="1.0" encoding="UTF-8"?>
<Request MajorVersion="1" MinorVersion="0">
 <CLI>
<Configuration>
interface GigabitEthernet0/1/1/1.840

ipv6 address fec0:140:9834::/64

exit

</Configuration>
</CLI>
<Delete>
    <Configuration Source="CurrentConfig">
      <EIGRP>
        <ProcessTable>
          <Process>
            <Naming>
              <ASNumber>100</ASNumber>
            </Naming>
            <VRFTable>
              <VRF>
                <Naming>
                  <VRFName>V10:ICICI_VPN</VRFName>
                </Naming>
                <VRF_AFTable>
                  <VRF_AF>
                    <Naming>
                      <VRF_AFType>IPv4</VRF_AFType>
                    </Naming>
                    <AutoSummary/>
                  </VRF_AF>
                </VRF_AFTable>
              </VRF>
            </VRFTable>
          </Process>
        </ProcessTable>
      </EIGRP>
      <InterfaceConfigurationTable>
        <InterfaceConfiguration>
          <Naming>
            <Name>GigabitEthernet0/1/1/1.840</Name>
```

```
            <Active>act</Active>
          </Naming>
          <Shutdown>true</Shutdown>
        </InterfaceConfiguration>
      </InterfaceConfigurationTable>
    </Configuration>
  </Delete>
  <Set>
    <Configuration Source="CurrentConfig">
      <InterfaceConfigurationTable>
        <InterfaceConfiguration>
          <Naming>
            <Name>GigabitEthernet0/1/1/1.840</Name>
            <Active>act</Active>
          </Naming>
          <Description>GigabitEthernet0/1/1/1.840 dot1q vlan id=840. By VPNSC: Job Id# =
50</Description>
          <InterfaceModeNonPhysical>Default</InterfaceModeNonPhysical>
          <VLANSubConfiguration>
            <VLANIdentifier>
              <VlanType>VLANTypeDot1q</VlanType>
              <FirstTag>840</FirstTag>
            </VLANIdentifier>
          </VLANSubConfiguration>
          <VRF>V10:ICICI_VPN</VRF>
        </InterfaceConfiguration>
      </InterfaceConfigurationTable>
      <BGP>
        <AS>
          <Naming>
            <AS>0</AS>
          </Naming>
          <FourByteAS>
            <Naming>
              <AS>100</AS>
            </Naming>
            <VRFTable>
              <VRF>
                <Naming>
                  <Name>V10:ICICI_VPN</Name>
                </Naming>
                <VRFGlobal>
                  <Exists>true</Exists>
                  <VRFGlobalAFTable>
                    <VRFGlobalAF>
                      <Naming>
                        <AF>IPv6Unicast</AF>
                      </Naming>
                      <Enabled>true</Enabled>
                      <Redistribution>
                        <EIGRPRouteTable>
                          <EIGRPRoutes>
                            <Naming>
                              <EIGRPInstanceName>120</EIGRPInstanceName>
                            </Naming>
                          </EIGRPRoutes>
                        </EIGRPRouteTable>
                      </Redistribution>
                    </VRFGlobalAF>
                  </VRFGlobalAFTable>
                </VRFGlobal>
              </VRF>
            </VRFTable>
          </FourByteAS>
```

```
      </AS>
    </BGP>
    <EIGRP>
      <ProcessTable>
        <Process>
          <Naming>
            <ASNumber>100</ASNumber>
          </Naming>
          <VRFTable>
            <VRF>
              <Naming>
                <VRFName>V10:ICICI_VPN</VRFName>
              </Naming>
              <Enabled>true</Enabled>
              <VRF_AFTable>
                <VRF_AF>
                  <Naming>
                    <VRF_AFType>IPv4</VRF_AFType>
                  </Naming>
                  <Enabled>true</Enabled>
                  <RedistributeTable>
                    <Redistribute>
                      <Naming>
                        <Protocol>BGP</Protocol>
                        <SecondASNumber>100</SecondASNumber>
                      </Naming>
                      <PolicySpecified>false</PolicySpecified>
                    </Redistribute>
                  </RedistributeTable>
                  <DefaultMetric>
                    <BW>2000</BW>
                    <Delay>2001</Delay>
                    <Reliability>200</Reliability>
                    <Load>201</Load>
                    <MTU>20000</MTU>
                  </DefaultMetric>
                  <InterfaceTable>
                    <Interface>
                      <Naming>
                        <InterfaceName>GigabitEthernet0/1/1/1.840</InterfaceName>
                      </Naming>
                      <Enabled>true</Enabled>
                    </Interface>
                  </InterfaceTable>
                  <AutonomousSystem>120</AutonomousSystem>
                </VRF_AF>
              </VRF_AFTable>
            </VRF>
          </VRFTable>
        </Process>
      </ProcessTable>
    </EIGRP>
  </Configuration>
 </Set>
 <Commit/>
</Request>Comments
```

- In IOS XR, device configuration is specified in XML format.
- With respect to the XML schemas, different versions of IOS XR generates different XML configlets. However the configurations are almost identical, except for changes in the XML schema.
- There are different cases to consider. For example, when a service request is decommissioned or modified, the XML configurations are slightly different.

PE L3 MPLS VPN (Dual-stack, Static [IPv4], BGP [IPv6], IOS)

Configuration

- Service: L3 MPLS VPN.
- Feature: MPLS service request with VPN routing protocol as Static and BGP (dual-stack) on an IOS device.
- Device configuration:
  - The PE is running IOS version 12.2(33) SRD2.

    Interface(s): GigabitEthernet2/3.345.
  - Routing protocol = STATIC (IPv4), BGP (IPv6).

Configlets

PE

(See the extended code sample below.)

```
!
vrf definition UP-Tony-1
rd 1:45
address-family ipv4
route-target import 64512:73647
route-target import 64512:73648
route-target export 64512:73647
mdt default 225.4.4.1
mdt data 225.4.4.2 0.0.0.0 threshold 2343
mdt mtu 2345
address-family ipv6
route-target import 64512:73647
route-target import 64512:73648
route-target export 64512:73647
!
interface GigabitEthernet2/3.345
description GigabitEthernet2/3.345 dot1q vlan id=345. By VPNSC: Job Id# = 42
encapsulation dot1Q 345
vrf forwarding UP-Tony-1
ip address 44.5.5.5 255.255.255.0
ipv6 address 53:33::3/60
ip pim sparse-dense-mode
mpls label protocol ldp
mpls ip
no shutdown
!
ip multicast vrf UP-Tony-1 route-limit 12343
!
ip multicast-routing vrf UP-Tony-1
!
ip pim vrf UP-Tony-1 autorp listener
!
ip pim vrf UP-Tony-1 rp-address 4.3.3.4 list132 override
!
router bgp 64512
address-family ipv4 vrf UP-Tony-1
default-information originate
redistribute connected
redistribute static
```

```
exit-address-family
address-family ipv6 vrf UP-Tony-1
neighbor 535::2 remote-as 35
neighbor 535::2 activate
neighbor 535::2 as-override
neighbor 535::2 allowas-in 1
neighbor 535::2 send-community both
neighbor 535::2 advertisement-interval 34
neighbor 535::2 maximum-prefix 455 23 restart 2345
redistribute connected
redistribute static
exit-address-family
!
ip route vrf UP-Tony-1 34.5.3.3 255.255.255.255 GigabitEthernet2/3.345 4.5.3.2 234
!
ip route vrf UP-Tony-1 44.3.4.4 255.255.255.255 GigabitEthernet2/3.345 4.5.3.2 23
```

Comments

- None

CE-PE L3 MPLS VPN (Q-in-Q/Second VLAN ID, IOS)

Configuration

- Service: L3 MPLS VPN.
- Feature: CE-PE. Q-in-Q (second VLAN ID) is configured on the PE.
- Device configuration:
  - The N-PE is a Cisco 7606-S with IOS 12.2(33)SRC, and with an ES20 line card.

    Interface(s): GE2/0/15.
  - The CE is a Cisco 2811.

    Interface(s): FE0/0.
  - VPN = spoke.

Configlets

CE	N-PE
!	!
interface FastEthernet0/0.158	ip vrf V15:MPLS-1
description FastEthernet0/0.158 dot1q vlan id=158. By VPNSC: Job Id# = 239	rd 100:6812
encapsulation dot1Q 158	route-target import 100:7000
ip address 10.1.1.98 255.255.255.252	route-target import 100:7001
no shutdown	route-target export 100:7000
!	!
ip route 0.0.0.0 0.0.0.0 FastEthernet0/0.158	interface GigabitEthernet2/0/15.158
	description GigabitEthernet2/0/15.158 dot1q vlan id=158. By VPNSC: Job Id# = 239
	encapsulation dot1Q 158 second-dot1q 1502
	ip vrf forwarding V15:MPLS-1
	ip address 10.1.1.97 255.255.255.252
	no shutdown
	!
	router bgp 100
	address-family ipv4 vrf V15:MPLS-1
	redistribute connected
	redistribute static
	exit-address-family

Comments

- Encapsulation must be dot1q; SVI disabled.
- The resulting CLI configuration command is:

  `encapsulation dot1Q <`*`VID-1`*`> second-dot1q <`*`VID-2`*`>`

  - *VID-1* can be assigned by Prime Fulfillment VLAN ID resource pools, or manually.
  - *VID-2* must be added manually. There is no support for autopick ID for the second VLAN ID.
- Platforms/IOS versions which support the command include, but are not limited to:
  - Cisco 7600/SRBx with ES-20, SIP400 + 2, and 5-port GE-V2 SPA.
  - Cisco 7600/SRCx ES-20, SIP400 + 2, 5-port GE-V2 SPA, and 10GE-V2 SPA.
  - Cisco 7200 NPE-G1 with IOS 12.4 mainline.
  - Cisco 7200 NPE-G2 with IOS 12.4(4)XD.

- Q-in-Q is also supported for IOS XR devices.
- There is a template variable for second VLAN ID: *Second_PE_Vlan_ID*.
- Network configurations supported include:
  - PE only.
  - PE-CE with managed and unmanaged CEs.

Note Q-in-Q/second VLAN ID is configured only on the PE, irrespective of whether the CE is managed or unmanaged.

For additional coverage of Q-in-Q support in Prime Fulfillment, see the coverage of the Second VLAN ID attribute in the section Creating an MPLS VPN PE-CE Service Request, page 5-81.

CE-PE L3 MPLS VPN (Q-in-Q/Second VLAN ID, IOS XR)

Configuration

- Service: L3 MPLS VPN.
- Feature: CE-PE. Q-in-Q (second VLAN ID) is configured on the PE.
- Device configuration:
  - The PE is a Cisco GSR 12008 with IOS XR versions 3.8.1 or 3.9.0.

    Interface(s): TenGigE0/0/0/0.

Configlets

PE

The code examples below show CLI and XML configlets. All configlets are deployed on the PE device.

Sample CLI Configlets

The following is a sample CLI configlet for an IOS XR device running IOS XR 3.8.1.

```
vrf V3:Vpn-Apr-30
 address-family ipv4 unicast
  import route-target
   64512:9688
   64512:9689
  !
  export route-target
   64512:9688
  !
 !
 address-family ipv6 unicast
  import route-target
   64512:9688
   64512:9689
  !
  export route-target
   64512:9688
  !
 !
!
interface TenGigE0/0/0/0.1825
 description TenGigE0/0/0/0.1825 dot1q vlan id=1825. By VPNSC: Job Id# = 29
 vrf V3:Vpn-Apr-30
 ipv4 address 6.8.14.15 255.255.255.0
 ipv6 address 18::219/64
 dot1q vlan 1825 869
!
router bgp 64512
 vrf V3:Vpn-Apr-30
  rd 64512:9864
  address-family ipv4 unicast
   redistribute static
  !
  address-family ipv6 unicast
   redistribute static
  !
 !
!
end
```

The following is a sample CLI configlet for an IOS XR device running IOS XR 3.9.0.

```
vrf V3:Vpn-Apr-30
 address-family ipv4 unicast
  import route-target
   64512:9688
   64512:9689
  !
  export route-target
   64512:9688
  !
 !
 address-family ipv6 unicast
  import route-target
   64512:9688
   64512:9689
  !
  export route-target
   64512:9688
  !
 !
!
interface GigabitEthernet0/3/0/1.488
 description GigabitEthernet0/3/0/1.488 dot1q vlan id=488. By VPNSC: Job Id# = 30
 vrf V3:Vpn-Apr-30
 ipv4 address 25.14.12.4 255.255.255.0
 ipv6 address 98::16/64
 dot1q vlan 488 758
!
router bgp 64512
 address-family vpnv4 unicast
 !
 address-family vpnv6 unicast
 !
 vrf V3:Vpn-Apr-30
  rd 64512:9864
  address-family ipv4 unicast
   redistribute static
  !
  address-family ipv6 unicast
   redistribute static
  !
 !
!
end
```

Sample XML Configlets

The following is a sample XML configlet for an IOS XR device running IOS XR 3.8.1.

```
<?xml version="1.0" encoding="UTF-8"?>
<Request MajorVersion="1" MinorVersion="0">
 <CLI>
<Configuration>
vrf V3:Vpn-Apr-30
address-family ipv6 unicast
import route-target 64512:9688
import route-target 64512:9689
export route-target 64512:9688
exit
interface TenGigE0/0/0/0.1825
ipv6 address 18::219/64
</Configuration>
```

```
</CLI>
<Delete>
    <Configuration Source="CurrentConfig">
      <InterfaceConfigurationTable>
        <InterfaceConfiguration>
          <Naming>
            <Name>TenGigE0/0/0/0.1825</Name>
            <Active>act</Active>
          </Naming>
          <Shutdown>true</Shutdown>
        </InterfaceConfiguration>
      </InterfaceConfigurationTable>
    </Configuration>
  </Delete>
  <Set>
    <Configuration Source="CurrentConfig">
      <VRFTable>
        <VRF>
          <Naming>
            <Name>V3:Vpn-Apr-30</Name>
          </Naming>
          <Create>true</Create>
          <AFI_SAFITable>
            <AFI_SAFI>
              <Naming>
                <AFI>IPv4</AFI>
                <SAFI>Unicast</SAFI>
                <Topology>default</Topology>
              </Naming>
              <Create>true</Create>
              <BGP>
                <ImportRouteTargets>
                  <RouteTargetTable>
                    <RouteTarget>
                      <Naming>
                        <Type>AS</Type>
                        <AS>64512</AS>
                        <ASIndex>9688</ASIndex>
                      </Naming>
                      <True>true</True>
                    </RouteTarget>
                    <RouteTarget>
                      <Naming>
                        <Type>AS</Type>
                        <AS>64512</AS>
                        <ASIndex>9689</ASIndex>
                      </Naming>
                      <True>true</True>
                    </RouteTarget>
                  </RouteTargetTable>
                </ImportRouteTargets>
                <ExportRouteTargets>
                  <RouteTargetTable>
                    <RouteTarget>
                      <Naming>
                        <Type>AS</Type>
                        <AS>64512</AS>
                        <ASIndex>9688</ASIndex>
                      </Naming>
                      <True>true</True>
                    </RouteTarget>
                  </RouteTargetTable>
                </ExportRouteTargets>
              </BGP>
```

```
            </AFI_SAFI>
          </AFI_SAFITable>
        </VRF>
      </VRFTable>
      <InterfaceConfigurationTable>
        <InterfaceConfiguration>
          <Naming>
            <Name>TenGigE0/0/0/0.1825</Name>
            <Active>act</Active>
          </Naming>
          <Description>TenGigE0/0/0/0.1825 dot1q vlan id=1825. By VPNSC: Job Id# =
29</Description>
          <InterfaceModeNonPhysical>Default</InterfaceModeNonPhysical>
          <VLANSubConfiguration>
            <VLANIdentifier>
              <VlanType>VLANTypeDot1q</VlanType>
              <FirstTag>1825</FirstTag>
              <SecondTag>869</SecondTag>
            </VLANIdentifier>
          </VLANSubConfiguration>
          <VRF>V3:Vpn-Apr-30</VRF>
          <IPV4Network>
            <Addresses>
              <Primary>
                <IPAddress>6.8.14.15</IPAddress>
                <Mask>255.255.255.0</Mask>
              </Primary>
            </Addresses>
          </IPV4Network>
        </InterfaceConfiguration>
      </InterfaceConfigurationTable>
      <BGP>
        <AS>
          <Naming>
            <AS>0</AS>
          </Naming>
          <FourByteAS>
            <Naming>
              <AS>64512</AS>
            </Naming>
            <VRFTable>
              <VRF>
                <Naming>
                  <Name>V3:Vpn-Apr-30</Name>
                </Naming>
                <VRFGlobal>
                  <Exists>true</Exists>
                  <RouteDistinguisher>
                    <Type>AS</Type>
                    <AS>64512</AS>
                    <ASIndex>9864</ASIndex>
                  </RouteDistinguisher>
                  <VRFGlobalAFTable>
                    <VRFGlobalAF>
                      <Naming>
                        <AF>IPv4Unicast</AF>
                      </Naming>
                      <Enabled>true</Enabled>
                      <StaticRoutes/>
                    </VRFGlobalAF>
                  </VRFGlobalAFTable>
                  <VRFGlobalAFTable>
                    <VRFGlobalAF>
                      <Naming>
```

```
                    <AF>IPv6Unicast</AF>
                  </Naming>
                  <Enabled>true</Enabled>
                  <StaticRoutes/>
                </VRFGlobalAF>
              </VRFGlobalAFTable>
            </VRFGlobal>
          </VRF>
        </VRFTable>
      </FourByteAS>
    </AS>
  </BGP>
 </Configuration>
</Set>
<Commit/>
</Request>
```

The following is a sample XML configlet for an IOS XR device running IOS XR 3.9.0.

```
<?xml version="1.0" encoding="UTF-8"?>
<Request MajorVersion="1" MinorVersion="0">
 <CLI>
<Configuration>
vrf V3:Vpn-Apr-30
address-family ipv6 unicast
import route-target 64512:9688
import route-target 64512:9689
export route-target 64512:9688
exit
interface GigabitEthernet0/3/0/1.488
ipv6 address 98::16/64
</Configuration>
</CLI>
<Delete>
    <Configuration Source="CurrentConfig">
      <InterfaceConfigurationTable>
        <InterfaceConfiguration>
          <Naming>
            <InterfaceName>GigabitEthernet0/3/0/1.488</InterfaceName>
            <Active>act</Active>
          </Naming>
          <Shutdown>true</Shutdown>
        </InterfaceConfiguration>
      </InterfaceConfigurationTable>
    </Configuration>
  </Delete>
  <Set>
    <Configuration Source="CurrentConfig">
      <VRFTable>
        <VRF>
          <Naming>
            <VRFName>V3:Vpn-Apr-30</VRFName>
          </Naming>
          <Create>true</Create>
          <AFTable>
            <AF>
              <Naming>
                <AFName>IPv4</AFName>
                <SAFName>Unicast</SAFName>
                <TopologyName>default</TopologyName>
              </Naming>
              <Create>true</Create>
              <BGP>
                <ImportRouteTargets>
```

```
                        <RouteTargetTable>
                          <RouteTarget>
                            <Naming>
                              <Type>AS</Type>
                              <AS_XX>0</AS_XX>
                              <AS>64512</AS>
                              <ASIndex>9688</ASIndex>
                            </Naming>
                            <Enable>true</Enable>
                          </RouteTarget>
                          <RouteTarget>
                            <Naming>
                              <Type>AS</Type>
                              <AS_XX>0</AS_XX>
                              <AS>64512</AS>
                              <ASIndex>9689</ASIndex>
                            </Naming>
                            <Enable>true</Enable>
                          </RouteTarget>
                        </RouteTargetTable>
                      </ImportRouteTargets>
                      <ExportRouteTargets>
                        <RouteTargetTable>
                          <RouteTarget>
                            <Naming>
                              <Type>AS</Type>
                              <AS_XX>0</AS_XX>
                              <AS>64512</AS>
                              <ASIndex>9688</ASIndex>
                            </Naming>
                            <Enable>true</Enable>
                          </RouteTarget>
                        </RouteTargetTable>
                      </ExportRouteTargets>
                    </BGP>
                  </AF>
                </AFTable>
              </VRF>
            </VRFTable>
            <InterfaceConfigurationTable>
              <InterfaceConfiguration>
                <Naming>
                  <InterfaceName>GigabitEthernet0/3/0/1.488</InterfaceName>
                  <Active>act</Active>
                </Naming>
                <Description>GigabitEthernet0/3/0/1.488 dot1q vlan id=488. By VPNSC: Job Id# =
30</Description>
                <InterfaceModeNonPhysical>Default</InterfaceModeNonPhysical>
                <VLANSubConfiguration>
                  <VLANIdentifier>
                    <VlanType>VLANTypeDot1q</VlanType>
                    <FirstTag>488</FirstTag>
                    <SecondTag>758</SecondTag>
                  </VLANIdentifier>
                </VLANSubConfiguration>
                <VRF>V3:Vpn-Apr-30</VRF>
                <IPV4Network>
                  <Addresses>
                    <Primary>
                      <Address>25.14.12.4</Address>
                      <Netmask>255.255.255.0</Netmask>
                    </Primary>
                  </Addresses>
                </IPV4Network>
```

```
  </InterfaceConfiguration>
</InterfaceConfigurationTable>
<BGP>
  <AS>
    <Naming>
      <AS>0</AS>
    </Naming>
    <FourByteAS>
      <Naming>
        <AS>64512</AS>
      </Naming>
      <VRFTable>
        <VRF>
          <Naming>
            <VRFName>V3:Vpn-Apr-30</VRFName>
          </Naming>
          <VRFGlobal>
            <Exists>true</Exists>
            <RouteDistinguisher>
              <Type>AS</Type>
              <AS_XX>0</AS_XX>
              <AS>64512</AS>
              <ASIndex>9864</ASIndex>
            </RouteDistinguisher>
            <VRFGlobalAFTable>
              <VRFGlobalAF>
                <Naming>
                  <AFName>IPv4Unicast</AFName>
                </Naming>
                <Enable>true</Enable>
                <StaticRoutes/>
              </VRFGlobalAF>
            </VRFGlobalAFTable>
            <VRFGlobalAFTable>
              <VRFGlobalAF>
                <Naming>
                  <AFName>IPv6Unicast</AFName>
                </Naming>
                <Enable>true</Enable>
                <StaticRoutes/>
              </VRFGlobalAF>
            </VRFGlobalAFTable>
          </VRFGlobal>
        </VRF>
      </VRFTable>
      <DefaultVRF>
        <Global>
          <GlobalAFTable>
            <GlobalAF>
              <Naming>
                <AFName>VPNv4Unicast</AFName>
              </Naming>
              <Enable>true</Enable>
            </GlobalAF>
            <GlobalAF>
              <Naming>
                <AFName>VPNv6Unicast</AFName>
              </Naming>
              <Enable>true</Enable>
            </GlobalAF>
          </GlobalAFTable>
        </Global>
      </DefaultVRF>
    </FourByteAS>
```

```
        </AS>
       </BGP>
     </Configuration>
   </Set>
   <Commit/>
</Request>
```

Comments

- None.

PE L3 MPLS VPN (with Multicast, IPv4 and IPv6 Enabled VPN, IOS XR)

Configuration

- Service: L3 MPLS VPN.
- Feature: MPLS service request with multicast IPv4 and IPv6 enabled on IOS XR.
- Device configuration:
  - The PE is an iscind-12010-1 (GSR) with IOS XR version 3.7.1[00].

    Interface(s): GigabitEthernet0/1/0/1.
  - Routing protocol = None.

Configlets

PE

The code examples below show CLI and XML configlets for the MPLS service request.

CLI Configlets

```
vrf V18:VPN_Verve1
 address-family ipv4 unicast
  import route-target
   100:19916
   100:19917
  !
  export route-target
   100:19916
  !
 !
 address-family ipv6 unicast
  import route-target
   100:19916
   100:19917
  !
  export route-target
   100:19916
  !
 !
!
interface GigabitEthernet0/1/0/1.2589
 description GigabitEthernet0/1/0/1.2589 dot1q vlan id=2589. By VPNSC: Job Id# = 54
 vrf V18:VPN_Verve1
 ipv4 address 115.106.116.122 255.255.255.0
 ipv6 address 1125::254/24
 dot1q vlan 2589
!
router bgp 100
 vrf V18:VPN_Verve1
  rd 100:19891
  address-family ipv4 unicast
  !
  address-family ipv6 unicast
  !
 !
!
```

```
multicast-routing
 vrf V18:VPN_Verve1 address-family ipv4
  interface GigabitEthernet0/1/0/1.2589
   enable
  !
  mdt mtu 8003
  mdt data 224.10.0.5/32 threshold 8002
  mdt default ipv4 224.10.0.4
 !
 vrf V18:VPN_Verve1 address-family ipv6
  interface GigabitEthernet0/1/0/1.2589
   enable
  !
  mdt mtu 8003
  mdt default ipv4 224.10.0.4
 !
!
router pim vrf V18:VPN_Verve1 address-family ipv4
 rp-address 115.101.110.122 list1
!
router pim vrf V18:VPN_Verve1 address-family ipv6
 rp-address 1114::122 list2
!
end
```

XML Configlets

```
<?xml version="1.0" encoding="UTF-8"?>
<Request MajorVersion="1" MinorVersion="0">
 <CLI>
<Configuration>
vrf V18:VPN_Verve1
address-family ipv6 unicast
import route-target 100:19916
import route-target 100:19917
export route-target 100:19916
exit
interface GigabitEthernet0/1/0/1.2589
ipv6 address 1125::254/24
multicast-routing
vrf V18:VPN_Verve1
mdt default 224.10.0.4
mdt data 224.10.0.5/32 threshold 8002
mdt mtu 8003
interface GigabitEthernet0/1/0/1.2589
enable
vrf V18:VPN_Verve1 address-family ipv6
mdt default 224.10.0.4
mdt mtu 8003
interface GigabitEthernet0/1/0/1.2589
enable
router pim vrf V18:VPN_Verve1 address-family ipv4 rp-address 115.101.110.122 list1
router pim vrf V18:VPN_Verve1 address-family ipv6 rp-address 1114::122 list2
</Configuration>
</CLI>
<Delete>
    <Configuration Source="CurrentConfig">
      <InterfaceConfigurationTable>
        <InterfaceConfiguration>
          <Naming>
            <Name>GigabitEthernet0/1/0/1.2589</Name>
            <Active>act</Active>
```

```
        </Naming>
        <Shutdown>true</Shutdown>
      </InterfaceConfiguration>
    </InterfaceConfigurationTable>
  </Configuration>
</Delete>
<Set>
  <Configuration Source="CurrentConfig">
    <VRFTable>
      <VRF>
        <Naming>
          <Name>V18:VPN_Verve1</Name>
        </Naming>
        <Create>true</Create>
        <AFI_SAFITable>
          <AFI_SAFI>
            <Naming>
              <AFI>IPv4</AFI>
              <SAFI>Unicast</SAFI>
              <Topology>default</Topology>
            </Naming>
            <Create>true</Create>
            <BGP>
              <ImportRouteTargets>
                <RouteTargetTable>
                  <RouteTarget>
                    <Naming>
                      <Type>AS</Type>
                      <AS>100</AS>
                      <ASIndex>19916</ASIndex>
                    </Naming>
                    <True>true</True>
                  </RouteTarget>
                  <RouteTarget>
                    <Naming>
                      <Type>AS</Type>
                      <AS>100</AS>
                      <ASIndex>19917</ASIndex>
                    </Naming>
                    <True>true</True>
                  </RouteTarget>
                </RouteTargetTable>
              </ImportRouteTargets>
              <ExportRouteTargets>
                <RouteTargetTable>
                  <RouteTarget>
                    <Naming>
                      <Type>AS</Type>
                      <AS>100</AS>
                      <ASIndex>19916</ASIndex>
                    </Naming>
                    <True>true</True>
                  </RouteTarget>
                </RouteTargetTable>
              </ExportRouteTargets>
            </BGP>
          </AFI_SAFI>
        </AFI_SAFITable>
      </VRF>
    </VRFTable>
    <InterfaceConfigurationTable>
      <InterfaceConfiguration>
        <Naming>
          <Name>GigabitEthernet0/1/0/1.2589</Name>
```

```
            <Active>act</Active>
          </Naming>
          <Description>GigabitEthernet0/1/0/1.2589 dot1q vlan id=2589. By VPNSC: Job Id# =
54</Description>
          <InterfaceModeNonPhysical>Default</InterfaceModeNonPhysical>
          <VLANSubConfiguration>
            <VLANIdentifier>
              <VlanType>VLANTypeDot1q</VlanType>
              <FirstTag>2589</FirstTag>
            </VLANIdentifier>
          </VLANSubConfiguration>
          <VRF>V18:VPN_Verve1</VRF>
          <IPV4Network>
            <Addresses>
              <Primary>
                <IPAddress>115.106.116.122</IPAddress>
                <Mask>255.255.255.0</Mask>
              </Primary>
            </Addresses>
          </IPV4Network>
        </InterfaceConfiguration>
      </InterfaceConfigurationTable>
      <BGP>
        <AS>
          <Naming>
            <AS>0</AS>
          </Naming>
          <FourByteAS>
            <Naming>
              <AS>100</AS>
            </Naming>
            <VRFTable>
              <VRF>
                <Naming>
                  <Name>V18:VPN_Verve1</Name>
                </Naming>
                <VRFGlobal>
                  <Exists>true</Exists>
                  <RouteDistinguisher>
                    <Type>AS</Type>
                    <AS>100</AS>
                    <ASIndex>19891</ASIndex>
                  </RouteDistinguisher>
                  <VRFGlobalAFTable>
                    <VRFGlobalAF>
                      <Naming>
                        <AF>IPv4Unicast</AF>
                      </Naming>
                      <Enabled>true</Enabled>
                    </VRFGlobalAF>
                  </VRFGlobalAFTable>
                  <VRFGlobalAFTable>
                    <VRFGlobalAF>
                      <Naming>
                        <AF>IPv6Unicast</AF>
                      </Naming>
                      <Enabled>true</Enabled>
                    </VRFGlobalAF>
                  </VRFGlobalAFTable>
                </VRFGlobal>
              </VRF>
            </VRFTable>
          </FourByteAS>
        </AS>
```

```
      </BGP>
    </Configuration>
  </Set>
  <Commit/>
</Request>
```

Comments

- This service request uses the MPLS VPN PE_NO_CE policy.
- This service request has multicast IPv4 and IPv6 enabled VPN and also static RPs, as shown in the in the configlets.

PE L3 MPLS VPN (Static, IOS, IPv6)

Configuration

- Service: L3 MPLS VPN.
- Feature: MPLS service request with VPN routing protocol as Static on an IOS device using IPv6 addressing.
- Device configuration:
  - The PE is running IOS 12.2(33) SRD2.

    Interface(s): GigabitEthernet2/3.455.
  - Routing protocol = STATIC.

Configlets

PE

```
vrf definition test-vpn-1
rd 123:4
address-family ipv6
route-target import 64512:73647
route-target import 64512:73648
route-target export 64512:73647
!
interface GigabitEthernet2/3.455
description GigabitEthernet2/3.455 dot1q vlan id=455. By VPNSC: Job Id# = 87
encapsulation dot1Q 455
vrf forwarding test-vpn-1
ipv6 address 455::2/60
no shutdown
!
router bgp 64512
address-family ipv6 vrf test-vpn-1
default-information originate
redistribute connected
redistribute static
exit-address-family
!
ipv6 route vrf test-vpn-1 54::4/128 GigabitEthernet2/3.455 24::5 45
```

Comments

- None.

PE L3 MPLS VPN (BGP, IOS)

Configuration

- Service: L3 MPLS VPN.
- Feature: MPLS service request with VPN routing protocol as BGP on IOS.
- Device configuration:
  - The PE is an iscind-7600-2 with IOS version 12.2(17r) S2.

    Interface(s): FastEthernet2/14.
  - Routing protocol = BGP.

Configlets

PE

```
!
ip vrf V21:VPN
rd 100:19894
route-target import 100:19906
route-target import 100:19907
route-target export 100:19906
!
interface FastEthernet2/14.2691
description FastEthernet2/14.2691 dot1q vlan id=2691. By VPNSC: Job Id# = 59
encapsulation dot1Q 2691
ip vrf forwarding V21:VPN
ip address 115.123.102.122 255.255.255.0
no shutdown
!
router bgp 100
address-family ipv4 vrf V21:VPN
neighbor 115.102.123.102 remote-as 100
neighbor 115.102.123.102 activate
neighbor 115.102.123.102 allowas-in 5
neighbor 115.102.123.102 send-community both
neighbor 115.102.123.102 advertisement-interval 122
neighbor 115.102.123.102 maximum-prefix 122 12 restart 122
neighbor 5.2.2.5 route-map TESTING_IN in
neighbor 5.2.2.5 route-map TESTING_OUT out
exit-address-family
```

Comments

- This service request uses the MPLS VPN PE_NO_CE policy.
- In this service request, the Neighbor Send Community attribute (which generates the **send-community** configuration command) is set to “Both”.

PE L3 MPLS VPN (BGP, IOS, IPv6)

Configuration

- Service: L3 MPLS VPN.
- Feature: MPLS service request with VPN routing protocol as BGP on an IOS device using IPv6 addressing.
- Device configuration:
  - The PE is running IOS version 12.2(33) SRD2.

    Interface(s): GigabitEthernet2/3.1234.
  - Routing protocol = BGP.

Configlets

PE

```
!
vrf definition VPN-test
rd 12:44
address-family ipv6
route-target import 64512:73647
route-target import 64512:73648
route-target export 64512:73647
!
interface GigabitEthernet2/3.1234
description GigabitEthernet2/3.1234 dot1q vlan id=1234. By VPNSC: Job Id# = 86
encapsulation dot1Q 1234
vrf forwarding VPN-test
ipv6 address 23::5/60
no shutdown
!
router bgp 64512
address-family ipv6 vrf VPN-test
neighbor 345::2 remote-as 44
neighbor 345::2 activate
neighbor 345::2 as-override
neighbor 345::2 allowas-in 4
neighbor 345::2 send-community both
neighbor 345::2 advertisement-interval 123
neighbor 345::2 maximum-prefix 4567 23 restart 234
redistribute connected
redistribute static
exit-address-family
```

Comments

- None

PE L3 MPLS VPN (BGP, IOS XR)

Configuration

- Service: L3 MPLS VPN.
- Feature: MPLS service request with VPN routing protocol as BGP on IOS XR.
- Device configuration:
  - The PE is a an iscind-12010-1 (GSR) with IOS XR version 3.7.1[00].

    Interface(s): GigabitEthernet0/1/0/1.
  - Routing protocol = BGP.

Configlets

PE

The code examples below show CLI and XML configlets for the MPLS service request.

CLI Configlets

```
vrf V25:Cisco3
 address-family ipv4 unicast
  import route-target
   100:19926
   100:19927
  !
  export route-target
   100:19926
  !
 !
!
interface GigabitEthernet0/1/0/1.2841
 description GigabitEthernet0/1/0/1.2841 dot1q vlan id=2841. By VPNSC: Job Id# = 86
 vrf V25:Cisco3
 ipv4 address 125.101.122.125 255.255.255.0
 dot1q vlan 2841
!
router bgp 100
 vrf V25:Cisco3
  rd 100:19898
  address-family ipv4 unicast
  !
  neighbor 112.120.102.112
   remote-as 100
   advertisement-interval 122
   address-family ipv4 unicast
    route-policy verve in
    allowas-in 3
    route-policy verve out
   site-of-origin 64512:700
   !
  !
 !
!
end
```

XML Configlets

```
<?xml version="1.0" encoding="UTF-8"?>
<Request MajorVersion="1" MinorVersion="0">
 <CLI>
<Configuration>
vrf V25:Cisco3
</Configuration>
</CLI>
<Delete>
    <Configuration Source="CurrentConfig">
      <InterfaceConfigurationTable>
        <InterfaceConfiguration>
          <Naming>
            <Name>GigabitEthernet0/1/0/1.2841</Name>
            <Active>act</Active>
          </Naming>
          <Shutdown>true</Shutdown>
        </InterfaceConfiguration>
      </InterfaceConfigurationTable>
    </Configuration>
  </Delete>
  <Set>
    <Configuration Source="CurrentConfig">
      <VRFTable>
        <VRF>
          <Naming>
            <Name>V25:Cisco3</Name>
          </Naming>
          <Create>true</Create>
          <AFI_SAFITable>
            <AFI_SAFI>
              <Naming>
                <AFI>IPv4</AFI>
                <SAFI>Unicast</SAFI>
                <Topology>default</Topology>
              </Naming>
              <Create>true</Create>
              <BGP>
                <ImportRouteTargets>
                  <RouteTargetTable>
                    <RouteTarget>
                      <Naming>
                        <Type>AS</Type>
                        <AS>100</AS>
                        <ASIndex>19926</ASIndex>
                      </Naming>
                      <True>true</True>
                    </RouteTarget>
                    <RouteTarget>
                      <Naming>
                        <Type>AS</Type>
                        <AS>100</AS>
                        <ASIndex>19927</ASIndex>
                      </Naming>
                      <True>true</True>
                    </RouteTarget>
                  </RouteTargetTable>
                </ImportRouteTargets>
                <ExportRouteTargets>
                  <RouteTargetTable>
                    <RouteTarget>
                      <Naming>
```

```
                        <Type>AS</Type>
                        <AS>100</AS>
                        <ASIndex>19926</ASIndex>
                      </Naming>
                      <True>true</True>
                    </RouteTarget>
                  </RouteTargetTable>
                </ExportRouteTargets>
              </BGP>
            </AFI_SAFI>
          </AFI_SAFITable>
        </VRF>
      </VRFTable>
      <InterfaceConfigurationTable>
        <InterfaceConfiguration>
          <Naming>
            <Name>GigabitEthernet0/1/0/1.2841</Name>
            <Active>act</Active>
          </Naming>
          <Description>GigabitEthernet0/1/0/1.2841 dot1q vlan id=2841. By VPNSC: Job Id# =
86</Description>
          <InterfaceModeNonPhysical>Default</InterfaceModeNonPhysical>
          <VLANSubConfiguration>
            <VLANIdentifier>
              <VlanType>VLANTypeDot1q</VlanType>
              <FirstTag>2841</FirstTag>
            </VLANIdentifier>
          </VLANSubConfiguration>
          <VRF>V25:Cisco3</VRF>
          <IPV4Network>
            <Addresses>
              <Primary>
                <IPAddress>125.101.122.125</IPAddress>
                <Mask>255.255.255.0</Mask>
              </Primary>
            </Addresses>
          </IPV4Network>
        </InterfaceConfiguration>
      </InterfaceConfigurationTable>
      <BGP>
        <AS>
          <Naming>
            <AS>0</AS>
          </Naming>
          <FourByteAS>
            <Naming>
              <AS>100</AS>
            </Naming>
            <VRFTable>
              <VRF>
                <Naming>
                  <Name>V25:Cisco3</Name>
                </Naming>
                <VRFGlobal>
                  <Exists>true</Exists>
                  <RouteDistinguisher>
                    <Type>AS</Type>
                    <AS>100</AS>
                    <ASIndex>19898</ASIndex>
                  </RouteDistinguisher>
                  <VRFGlobalAFTable>
                    <VRFGlobalAF>
                      <Naming>
                        <AF>IPv4Unicast</AF>
```

```
      </Naming>
      <Enabled>true</Enabled>
    </VRFGlobalAF>
  </VRFGlobalAFTable>
</VRFGlobal>
<VRFNeighborTable>
  <VRFNeighbor>
    <Naming>
      <IPAddress>
        <IPV4Address>112.120.102.112</IPV4Address>
      </IPAddress>
    </Naming>
    <VRFNeighborAFTable>
      <VRFNeighborAF>
        <Naming>
          <AF>IPv4Unicast</AF>
        </Naming>
        <Activate>true</Activate>
      </VRFNeighborAF>
    </VRFNeighborAFTable>
    <RemoteAS>
      <AS_XX>0</AS_XX>
      <AS_YY>100</AS_YY>
    </RemoteAS>
  </VRFNeighbor>
  <VRFNeighbor>
    <Naming>
      <IPAddress>
        <IPV4Address>112.120.102.112</IPV4Address>
      </IPAddress>
    </Naming>
    <VRFNeighborAFTable>
      <VRFNeighborAF>
        <Naming>
          <AF>IPv4Unicast</AF>
        </Naming>
        <AllowASIn>3</AllowASIn>
      </VRFNeighborAF>
    </VRFNeighborAFTable>
  </VRFNeighbor>
  <VRFNeighbor>
    <Naming>
      <IPAddress>
        <IPV4Address>112.120.102.112</IPV4Address>
      </IPAddress>
    </Naming>
    <VRFNeighborAFTable>
      <VRFNeighborAF>
        <Naming>
          <AF>IPv4Unicast</AF>
        </Naming>
        <RoutePolicyIn>verve</RoutePolicyIn>
        <RoutePolicyIn>verve</RoutePolicyIn>
      </VRFNeighborAF>
    </VRFNeighborAFTable>
  </VRFNeighbor>
  <VRFNeighbor>
    <Naming>
      <IPAddress>
        <IPV4Address>112.120.102.112</IPV4Address>
      </IPAddress>
    </Naming>
    <VRFNeighborAFTable>
      <VRFNeighborAF>
```

```
                        <Naming>
                          <AF>IPv4Unicast</AF>
                        </Naming>
                        <RoutePolicyOut>verve</RoutePolicyOut>
                        <RoutePolicyOut>verve</RoutePolicyOut>
                      </VRFNeighborAF>
                    </VRFNeighborAFTable>
                  </VRFNeighbor>
                  <VRFNeighbor>
                    <Naming>
                      <IPAddress>
                        <IPV4Address>112.120.102.112</IPV4Address>
                      </IPAddress>
                    </Naming>
                    <VRFNeighborAFTable>
                      <VRFNeighborAF>
                        <Naming>
                          <AF>IPv4Unicast</AF>
                        </Naming>
                        <Activate>true</Activate>
                      </VRFNeighborAF>
                    </VRFNeighborAFTable>
                    <AdvertisementInterval>122</AdvertisementInterval>
                  </VRFNeighbor>
                </VRFNeighborTable>
              </VRF>
            </VRFTable>
          </FourByteAS>
        </AS>
      </BGP>
    </Configuration>
  </Set>
  <Commit/>
</Request>
```

Comments

- This service request is using the MPLS VPN PE_NO_CE policy.
- In this service request, the Neighbor Send Community attribute (which generates the **send-community** configuration command) is set as “None”.
- In this service request, a route-policy name has been supplied using the Route Map/Policy In (Out) attribute(s).

  Note The route policy is already present on the device.

  The deployment used that name only, as shown in the configlets.
- If no route-map name had been supplied, then Prime Fulfillment would have added IscDefaultPassAll as the default. This default is only added in the case of IOS XR devices. No default is added for IOS devices.

PE L3 MPLS VPN (BGP, RD Format, IOS XR)

Configuration

- Service: L3 MPLS VPN
- Feature: MPLS service request with BGP protocol and RD IP address format on IOS XR.
- Device configuration:
  - The PE is a Cisco IOX device with IOS XR version 3.7.1.

    Interface(s): GigabitEthernet.
  - Routing protocol = BGP.

Configlets

PE

The code examples below show CLI and XML configlets for the MPLS service request.

MPLS Service Request CLI Configlet

```
vrf V29:vpn_techm_cisco
 address-family ipv6 unicast
  import route-target
   100:15038
   100:15039
  !
  export route-target
   100:15038
  !
 !
!

Router bgp 100
 vrf V29:vpn_techm_cisco
  rd 13.13.13.1:14540
    address-family ipv6 unicast
  !
 !
```

MPLS Service Request XML Configlets

```
<VRF>
   <Naming>
      <Name>V1:vpn1</Name>
   </Naming>
   <VRFGlobal>
      <Exists>true</Exists>
      <RouteDistinguisher>
         <Type> IPV4Address </Type>
         <Addr>13.13.13.1</Addr>
         <AddrIndex>14540</AddrIndex>
      </RouteDistinguisher>
      <VRFGlobalAFTable>
         <VRFGlobalAF>
            <Naming>
               <AF>IPv4Unicast</AF>
```

```
            </Naming>
            <Enabled>true</Enabled>
            <StaticRoutes/>
         </VRFGlobalAF>
      </VRFGlobalAFTable>
   </VRFGlobal>
</VRF>
```

Comments

- None.

PE L3 MPLS VPN (BGP, Maximum Prefix/Restart, IOS XR)

Configuration

- Service: L3 MPLS VPN.
- Feature: MPLS service request using the BGP routing protocol and specifying the number of maximum prefixes and restart value.
- Device configuration:
  - The PE is an IOS XR device running IOS XR version 3.8.1 or 3.9.0.

    Interface(s): Various.
  - Routing protocol = BGP.

Configlets

PE

The code examples below show CLI and XML configlets. All configlets are deployed on the PE device.

Sample CLI Configlets

The following is a sample CLI configlet for an IOS XR device running IOS XR 3.8.1.

```
router bgp 64512
 vrf V22:27Cerc1
  address-family ipv4 unicast
  !
  address-family ipv6 unicast
  !
  neighbor 1.2.5.4
   address-family ipv4 unicast
    maximum-prefix 101 91 restart 81
   !
  !
  neighbor 11::69
   address-family ipv6 unicast
    maximum-prefix 124 46 restart 6711
   !
  !
 !
!
end
```

The following is a sample CLI configlet for an IOS XR device running IOS XR 3.9.0. This is an example showing restart configlets.

```
router bgp 64512
 vrf V23:27Cerc2
  address-family ipv4 unicast
  !
  address-family ipv6 unicast
  !
  neighbor 8.5.2.33
   address-family ipv4 unicast
    maximum-prefix 160 80 restart 300
   !
  !
  neighbor 25::9
   address-family ipv6 unicast
```

```
    maximum-prefix 200 26 restart 214
   !
  !
 !
!
end
```

The following is a sample CLI configlet for an IOS XR device running IOS XR 3.9.0. This is an example showing warning-only configlets.

```
router bgp 64512
 vrf V23:27Cerc2
  address-family ipv4 unicast
  !
  address-family ipv6 unicast
  !
  neighbor 8.5.2.33
   address-family ipv4 unicast
    maximum-prefix 160 80 warning-only
   !
  !
  neighbor 25::9
   address-family ipv6 unicast
    maximum-prefix 200 26 warning-only
   !
  !
 !
!
end
```

Sample XML Configlets

The following is a sample XML configlet for an IOS XR device running IOS XR 3.8.1.

```
<?xml version="1.0" encoding="UTF-8"?>
<Request MajorVersion="1" MinorVersion="0">
  <Set>
    <Configuration Source="CurrentConfig">
      <BGP>
        <AS>
          <Naming>
            <AS>0</AS>
          </Naming>
          <FourByteAS>
            <Naming>
              <AS>64512</AS>
            </Naming>
            <VRFTable>
              <VRF>
                <Naming>
                  <Name>V22:27Cerc1</Name>
                </Naming>
                <VRFGlobal>
                  <Exists>true</Exists>
                  <VRFGlobalAFTable>
                    <VRFGlobalAF>
                      <Naming>
                        <AF>IPv4Unicast</AF>
                      </Naming>
                      <Enabled>true</Enabled>
                    </VRFGlobalAF>
                  </VRFGlobalAFTable>
                  <VRFGlobalAFTable>
```

```
                    <VRFGlobalAF>
                      <Naming>
                        <AF>IPv6Unicast</AF>
                      </Naming>
                      <Enabled>true</Enabled>
                    </VRFGlobalAF>
                  </VRFGlobalAFTable>
                </VRFGlobal>
                <VRFNeighborTable>
                  <VRFNeighbor>
                    <Naming>
                      <IPAddress>
                        <IPV4Address>1.2.5.4</IPV4Address>
                      </IPAddress>
                    </Naming>
                    <VRFNeighborAFTable>
                      <VRFNeighborAF>
                        <Naming>
                          <AF>IPv4Unicast</AF>
                        </Naming>
                        <MaximumPrefixes>
                           <Value>101</Value>
                           <WarningPercentage>91</WarningPercentage>
                           <RestartTime>81</RestartTime>
                           <WarningOnly>false</WarningOnly>
                         </MaximumPrefixes>
                      </VRFNeighborAF>
                    </VRFNeighborAFTable>
                  </VRFNeighbor>
                  <VRFNeighbor>
                    <Naming>
                      <IPAddress>
                        <IPV6Address>11::69</IPV6Address>
                      </IPAddress>
                    </Naming>
                    <VRFNeighborAFTable>
                      <VRFNeighborAF>
                        <Naming>
                          <AF>IPv6Unicast</AF>
                        </Naming>
                        <MaximumPrefixes>
                           <Value>124</Value>
                           <WarningPercentage>46</WarningPercentage>
                           <RestartTime>6711</RestartTime>
                           <WarningOnly>false</WarningOnly>
                         </MaximumPrefixes>
                      </VRFNeighborAF>
                    </VRFNeighborAFTable>
                  </VRFNeighbor>
                </VRFNeighborTable>
              </VRF>
            </VRFTable>
          </FourByteAS>
        </AS>
      </BGP>
    </Configuration>
  </Set>
  <Commit/>
</Request>
```

The following is a sample XML configlet for for an IOS XR device running IOS XR 3.9.0.

```
<?xml version="1.0" encoding="UTF-8"?>
<Request MajorVersion="1" MinorVersion="0">
```

```
<Set>
  <Configuration Source="CurrentConfig">
    <BGP>
      <AS>
        <Naming>
          <AS>0</AS>
        </Naming>
        <FourByteAS>
          <Naming>
            <AS>64512</AS>
          </Naming>
          <VRFTable>
            <VRF>
              <Naming>
                <VRFName>V23:27Cerc2</VRFName>
              </Naming>
              <VRFGlobal>
                <Exists>true</Exists>
                <VRFGlobalAFTable>
                  <VRFGlobalAF>
                    <Naming>
                      <AFName>IPv4Unicast</AFName>
                    </Naming>
                    <Enable>true</Enable>
                  </VRFGlobalAF>
                </VRFGlobalAFTable>
                <VRFGlobalAFTable>
                  <VRFGlobalAF>
                    <Naming>
                      <AFName>IPv6Unicast</AFName>
                    </Naming>
                    <Enable>true</Enable>
                  </VRFGlobalAF>
                </VRFGlobalAFTable>
              </VRFGlobal>
              <VRFNeighborTable>
                <VRFNeighbor>
                  <Naming>
                    <NeighborAddress>
                      <IPV4Address>8.5.2.33</IPV4Address>
                    </NeighborAddress>
                  </Naming>
                  <VRFNeighborAFTable>
                    <VRFNeighborAF>
                      <Naming>
                        <AFName>IPv4Unicast</AFName>
                      </Naming>
                      <MaximumPrefixes>
                         <PrefixLimit>160</PrefixLimit>
                         <WarningPercentage>80</WarningPercentage>
                         <RestartTime>300</RestartTime>
                         <WarningOnly>false</WarningOnly>
                       </MaximumPrefixes>
                    </VRFNeighborAF>
                  </VRFNeighborAFTable>
                </VRFNeighbor>
                <VRFNeighbor>
                  <Naming>
                    <NeighborAddress>
                      <IPV6Address>25::9</IPV6Address>
                    </NeighborAddress>
                  </Naming>
                  <VRFNeighborAFTable>
                    <VRFNeighborAF>
```

```
                        <Naming>
                          <AFName>IPv6Unicast</AFName>
                        </Naming>
                        <MaximumPrefixes>
                           <PrefixLimit>200</PrefixLimit>
                           <WarningPercentage>26</WarningPercentage>
                           <RestartTime>214</RestartTime>
                           <WarningOnly>false</WarningOnly>
                         </MaximumPrefixes>
                      </VRFNeighborAF>
                    </VRFNeighborAFTable>
                  </VRFNeighbor>
                </VRFNeighborTable>
              </VRF>
            </VRFTable>
          </FourByteAS>
        </AS>
      </BGP>
    </Configuration>
  </Set>
  <Commit/>
</Request>
```

Comments

- If the user gives both warning only and restart values, Prime Fulfillment validates and gives the higher priority to the restart value for all IOS and IOS XR versions.
- For individual values (like warning-only or restart, if either is given), Prime Fulfillment configures accordingly.

PE L3 MPLS VPN (BGP, Default Information Originate, IOS XR)

Configuration

- Service: L3 MPLS VPN.
- Feature: MPLS service request using the BGP routing protocol and specifying setting the Default Information Originate attribute to cause the BGP speaker (local router) to send a default route to a neighbor.
- Device configuration:
  - The PE is an IOS XR device running IOS XR version 3.8.1 or 3.9.0.

    Interface(s): Various.
  - Routing protocol = BGP.

Configlets

PE

The code examples below show CLI and XML configlets. All configlets are deployed on the PE device.

Sample CLI Configlets

The following is a sample CLI configlet for an IOS XR device running IOS XR 3.8.1.

```
vrf V1:mpls
  rd 100:345
  address-family ipv4 unicast
   redistribute static
  !
  address-family ipv6 unicast
  !
  neighbor 1.1.1.1
   remote-as 100
   address-family ipv4 unicast
    default-originate route-policy dinesh
   !
  !
  neighbor 1.1.1.2
   remote-as 100
   address-family ipv4 unicast
    default-originate
   !
  !

  neighbor 2002::23
   remote-as 100
   address-family ipv6 unicast
    default-originate disable
   !
  !
 !
```

The following is a sample CLI configlet for an IOS XR device running IOS XR 3.9.0.

```
vrf V1:mpls
  rd 100:345
  address-family ipv4 unicast
   redistribute static
  !
```

```
 address-family ipv6 unicast
 !
 neighbor 1.1.1.1
  remote-as 100
  address-family ipv4 unicast
   default-originate route-policy dinesh
  !
 !
 neighbor 1.1.1.2
  remote-as 100
  address-family ipv4 unicast
   default-originate
  !
 !

 neighbor 2002::23
  remote-as 100
  address-family ipv6 unicast
   default-originate inheritance-disable
  !
 !
!
```

Sample XML Configlets

The following is a sample XML configlet for for an IOS XR device running IOS XR 3.8.1.

```
<BGP MajorVersion="30" MinorVersion="2">
   <AS>
      <Naming>
         <AS>0</AS>
      </Naming>
      <FourByteAS>
         <Naming>
            <AS>100</AS>
         </Naming>
         <BGPRunning>true</BGPRunning>
         <VRFTable>
            <VRF>
               <Naming>
                  <Name>V1:mpls</Name>
               </Naming>
               <VRFGlobal>
                  <Exists>true</Exists>
                  <RouteDistinguisher>
                     <Type>AS</Type>
                     <AS>100</AS>
                     <ASIndex>345</ASIndex>
                  </RouteDistinguisher>
                  <VRFGlobalAFTable>
                     <VRFGlobalAF>
                        <Naming>
                           <AF>IPv4Unicast</AF>
                        </Naming>
                        <Enabled>true</Enabled>
                        <StaticRoutes/>
                     </VRFGlobalAF>
                     <VRFGlobalAF>
                        <Naming>
                           <AF>IPv6Unicast</AF>
                        </Naming>
                        <Enabled>true</Enabled>
                     </VRFGlobalAF>
                  </VRFGlobalAFTable>
```

```
</VRFGlobal>
<VRFNeighborTable>
   <VRFNeighbor>
      <Naming>
         <IPAddress>
            <IPV4Address>1.1.1.1</IPV4Address>
         </IPAddress>
      </Naming>
      <RemoteAS>
         <AS_XX>0</AS_XX>
         <AS_YY>100</AS_YY>
      </RemoteAS>
      <VRFNeighborAFTable>
         <VRFNeighborAF>
            <Naming>
               <AF>IPv4Unicast</AF>
            </Naming>
            <Activate>true</Activate>
            <DefaultOriginate>
                <Enable>true</Enable>
                <RoutePolicy>dinesh</RoutePolicy>
             </DefaultOriginate>
         </VRFNeighborAF>
      </VRFNeighborAFTable>
   </VRFNeighbor>
             <VRFNeighbor>
      <Naming>
         <IPAddress>
            <IPV6Address>2002::23</IPV6Address>
         </IPAddress>
      </Naming>
      <RemoteAS>
         <AS_XX>0</AS_XX>
         <AS_YY>100</AS_YY>
      </RemoteAS>
      <VRFNeighborAFTable>
         <VRFNeighborAF>
            <Naming>
               <AF>IPv6Unicast</AF>
            </Naming>
            <Activate>true</Activate>
            <DefaultOriginate>
                <Enable>true</Enable>
             </DefaultOriginate>
         </VRFNeighborAF>
      </VRFNeighborAFTable>
   </VRFNeighbor>

   <VRFNeighbor>
      <Naming>
         <IPAddress>
            <IPV6Address>2002::23</IPV6Address>
         </IPAddress>
      </Naming>
      <RemoteAS>
         <AS_XX>0</AS_XX>
         <AS_YY>100</AS_YY>
      </RemoteAS>
      <VRFNeighborAFTable>
         <VRFNeighborAF>
            <Naming>
               <AF>IPv6Unicast</AF>
            </Naming>
            <Activate>true</Activate>
```

```
                        <DefaultOriginate>
                            <Enable>false</Enable>
                        </DefaultOriginate>
                    </VRFNeighborAF>
                </VRFNeighborAFTable>
            </VRFNeighbor>
        </VRFNeighborTable>
    </VRF>
</VRFTable>
</FourByteAS>
</AS>
</BGP>
```

Comments

- None.

PE L3 MPLS VPN (OSPF, IOS)

Configuration

- Service: L3 MPLS VPN.
- Feature: MPLS service request with VPN routing protocol as OSPF on IOS.
- Device configuration:
  - The PE is an iscind-7600-2 with IOS version 12.2(17r) S2.
  - Routing protocol = OSPF.

Configlets

PE

```
!
no interface FastEthernet2/14.2685
!
interface FastEthernet2/14.2677
description FastEthernet2/14.2677 dot1q vlan id=2677. By VPNSC: Job Id# = 60
encapsulation dot1Q 2677
ip vrf forwarding Tester1
ip address 112.126.102.106 255.255.255.0
no shutdown
!
router ospf 1266 vrf Tester1
redistribute bgp 100 subnets
network 112.126.102.0 0.0.0.255 area 23693
!
router bgp 100
address-family ipv4 vrf Tester1
redistribute ospf 1266 vrf Tester1 metric 1263 route-map verve match internal external 1
external 2
```

Comments

- This service request is using the MPLS VPN PE_NO_CE policy.
- OSPF Match Criteria is set as “Both”. So **internal**, **external1**, and **external2** configuration commands are generated in the configlet.
- There is no support for **external type 1** or **external type 2** commands in the IOS XR variation of this command, but they are support in IOS.

PE L3 MPLS VPN (OSPF, IOS XR)

Configuration

- Service: L3 MPLS VPN
- Feature: MPLS service request with VPN routing protocol as OSPF on IOS XR.
- Device configuration:
  - The PE is an mlpe7 with IOS XR version 3.6.1[00].

    Interface(s): GigabitEthernet0/1/0/1.
  - Routing protocol = OSPF.

Configlets

PE

The code examples below show CLI and XML configlets for the MPLS service request.

MPLS Service Request CLI Configlet

```
vrf V28:Cisco5
 address-family ipv4 unicast
  import route-target
   100:19930
   100:19931
  !
  export route-target
   100:19930
  !
 !
!
interface GigabitEthernet0/1/1/4.2693
 description GigabitEthernet0/1/1/4.2693 dot1q vlan id=2693. By VPNSC: Job Id# = 90
 vrf V28:Cisco5
 ipv4 address 123.33.102.112 255.255.255.0
 dot1q vlan 2693
!
router ospf 1238
 vrf V28:Cisco5
  redistribute bgp 100
  area 29871
   interface GigabitEthernet0/1/1/4.2693
   !
  !
 !
!
router bgp 100
 vrf V28:Cisco5
  rd 100:19901
  address-family ipv4 unicast
   redistribute ospf 1238 match internal external metric 2581 route-policy verve
  !
 !
!
end
```

MPLS Service Request XML Configlets

```
<?xml version="1.0" encoding="UTF-8"?>
<Request MajorVersion="1" MinorVersion="0">
 <CLI>
<Configuration>
vrf V28:Cisco5
</Configuration>
</CLI>
<Delete>
    <Configuration Source="CurrentConfig">
      <InterfaceConfigurationTable>
        <InterfaceConfiguration>
          <Naming>
            <Name>GigabitEthernet0/1/1/4.2693</Name>
            <Active>act</Active>
          </Naming>
          <Shutdown>true</Shutdown>
        </InterfaceConfiguration>
      </InterfaceConfigurationTable>
    </Configuration>
  </Delete>
  <Set>
    <Configuration Source="CurrentConfig">
      <VRFTable>
        <VRF>
          <Naming>
            <Name>V28:Cisco5</Name>
          </Naming>
          <Create>true</Create>
          <AFI_SAFITable>
            <AFI_SAFI>
              <Naming>
                <AFI>IPv4</AFI>
                <SAFI>Unicast</SAFI>
              </Naming>
              <Create>true</Create>
              <BGP>
                <ImportRouteTargets>
                  <RouteTargetTable>
                    <RouteTarget>
                      <Naming>
                        <Type>AS</Type>
                        <AS>100</AS>
                        <ASIndex>19930</ASIndex>
                      </Naming>
                      <True>true</True>
                    </RouteTarget>
                    <RouteTarget>
                      <Naming>
                        <Type>AS</Type>
                        <AS>100</AS>
                        <ASIndex>19931</ASIndex>
                      </Naming>
                      <True>true</True>
                    </RouteTarget>
                  </RouteTargetTable>
                </ImportRouteTargets>
                <ExportRouteTargets>
                  <RouteTargetTable>
                    <RouteTarget>
                      <Naming>
                        <Type>AS</Type>
```

```
                        <AS>100</AS>
                        <ASIndex>19930</ASIndex>
                      </Naming>
                      <True>true</True>
                    </RouteTarget>
                  </RouteTargetTable>
                </ExportRouteTargets>
              </BGP>
            </AFI_SAFI>
          </AFI_SAFITable>
        </VRF>
      </VRFTable>
      <InterfaceConfigurationTable>
        <InterfaceConfiguration>
          <Naming>
            <Name>GigabitEthernet0/1/1/4.2693</Name>
            <Active>act</Active>
          </Naming>
           <Description>GigabitEthernet0/1/1/4.2693 dot1q vlan id=2693. By VPNSC: Job Id# =
90</Description>
          <InterfaceModeNonPhysical>Default</InterfaceModeNonPhysical>
          <VLANSubConfiguration>
            <VLANIdentifier>
              <VlanType>VLANTypeDot1q</VlanType>
              <FirstTag>2693</FirstTag>
            </VLANIdentifier>
          </VLANSubConfiguration>
          <VRF>V28:Cisco5</VRF>
          <IPV4Network>
            <Addresses>
              <Primary>
                <IPAddress>123.33.102.112</IPAddress>
                <Mask>255.255.255.0</Mask>
              </Primary>
            </Addresses>
          </IPV4Network>
        </InterfaceConfiguration>
      </InterfaceConfigurationTable>
      <BGP>
        <AS>
          <Naming>
            <AS>0</AS>
          </Naming>
          <FourByteAS>
            <Naming>
              <AS>100</AS>
            </Naming>
            <VRFTable>
              <VRF>
                <Naming>
                  <Name>V28:Cisco5</Name>
                </Naming>
                <VRFGlobal>
                  <Exists>true</Exists>
                  <RouteDistinguisher>
                    <Type>AS</Type>
                    <AS>100</AS>
                    <ASIndex>19901</ASIndex>
                  </RouteDistinguisher>
                  <VRFGlobalAFTable>
                    <VRFGlobalAF>
                      <Naming>
                        <AF>IPv4Unicast</AF>
                      </Naming>
```

```
                    <Enabled>true</Enabled>
                    <OSPFRouteTable>
                      <OSPFRoutes>
                        <Naming>
                          <OSPFInstanceName>1238</OSPFInstanceName>
                        </Naming>
                        <RoutePolicy/>
                        <RedistType>21</RedistType>
                        <DefaultMetric>2581</DefaultMetric>
                      </OSPFRoutes>
                    </OSPFRouteTable>
                  </VRFGlobalAF>
                </VRFGlobalAFTable>
              </VRFGlobal>
            </VRF>
          </VRFTable>
        </FourByteAS>
      </AS>
    </BGP>
    <OSPF>
      <ProcessTable>
        <Process>
          <Naming>
            <InstanceName>1238</InstanceName>
          </Naming>
          <Start>true</Start>
          <VRFTable>
            <VRF>
              <Naming>
                <VRFName>V28:Cisco5</VRFName>
              </Naming>
              <VRFStart>true</VRFStart>
              <Redistribution>
                <RedistributeTable>
                  <Redistribute>
                    <Naming>
                      <ProtocolType>bgp</ProtocolType>
                      <InstanceName>bgp</InstanceName>
                      <BGP_AS_XX>0</BGP_AS_XX>
                      <BGP_AS_YY>100</BGP_AS_YY>
                    </Naming>
                    <Classful>false</Classful>
                  </Redistribute>
                </RedistributeTable>
              </Redistribution>
              <AreaTable>
                <Area>
                  <Naming>
                    <IntegerID>29871</IntegerID>
                  </Naming>
                  <NameScopeTable>
                    <NameScope>
                      <Naming>
                        <Interface>GigabitEthernet0/1/1/4.2693</Interface>
                      </Naming>
                      <Running>true</Running>
                    </NameScope>
                  </NameScopeTable>
                  <Running>true</Running>
                </Area>
              </AreaTable>
            </VRF>
          </VRFTable>
        </Process>
```

```
        </ProcessTable>
       </OSPF>
     </Configuration>
   </Set>
   <Commit/>
</Request>
```

Comments

- This service request uses the MPLS VPN PE_NO_CE policy.
- OSPF Match Criteria is set as “Both”. So **internal** and **external** configuration commands are generated in the configlets.
- There is no support for e**xternal type 1** or **external type 2** in the IOS XR variation of this command, but the support exists in IOS.

L3 MPLS VPN (OSPF, Default Information Originate, IOS XR)

Configuration

- Service: L3 MPLS VPN.
- Feature: MPLS service request using the OSPF routing protocol and setting the Default Information Originate to generate a default external route into an OSPF routing domain.
- Device configuration:
  - The PE is an IOS XR device running IOS XR version 3.9.0.

    Interface(s): Various.
  - Routing protocol = OSPF.

Configlets

PE

The code examples below show CLI and XML configlets. All configlets are deployed on the PE device.

Sample CLI Configlets

The following is a sample CLI configlet for an IOS XR device running IOS XR 3.9.0.

```
vrf V35:apr26-vpn9
 address-family ipv4 unicast
  import route-target
   64512:2776
   64512:2777
  !
  export route-target
   64512:2776
  !
 !
 address-family ipv6 unicast
  import route-target
   64512:2776
   64512:2777
  !
  export route-target
   64512:2776
  !
 !
!
interface GigabitEthernet0/15/1/1.947
 description GigabitEthernet0/15/1/1.947 dot1q vlan id=947. By VPNSC: Job Id# = 191
 vrf V35:apr26-vpn9
 ipv4 address 26.27.28.21 255.255.255.0
 ipv6 address 2165::541/32
 dot1q vlan 947
!
router ospf 1611
 vrf V35:apr26-vpn9
   default-information originate always metric 652 metric-type 2 route-policy dinesh
  area 218
   interface GigabitEthernet0/15/1/1.947
   !
  !
 !
!
```

```
router bgp 64512
 vrf V35:apr26-vpn9
  rd 64512:2190
  address-family ipv4 unicast
   redistribute connected
   redistribute static
   redistribute ospf 1611 match internal metric 325
  !
  address-family ipv6 unicast
   redistribute static
  !
 !
!
end
```

Sample XML Configlets

The following is a sample XML configlet for an IOS XR device running IOS XR 3.9.0.

```
<?xml version="1.0" encoding="UTF-8"?>
<Request MajorVersion="1" MinorVersion="0">
 <CLI>
<Configuration>
vrf V35:apr26-vpn9
address-family ipv6 unicast
import route-target 64512:2776
import route-target 64512:2777
export route-target 64512:2776
exit
interface GigabitEthernet0/15/1/1.947
ipv6 address 2165::541/32
</Configuration>
</CLI>
<Delete>
    <Configuration Source="CurrentConfig">
      <InterfaceConfigurationTable>
        <InterfaceConfiguration>
          <Naming>
            <InterfaceName>GigabitEthernet0/15/1/1.947</InterfaceName>
            <Active>act</Active>
          </Naming>
          <Shutdown>true</Shutdown>
        </InterfaceConfiguration>
      </InterfaceConfigurationTable>
    </Configuration>
  </Delete>
  <Set>
    <Configuration Source="CurrentConfig">
      <VRFTable>
        <VRF>
          <Naming>
            <VRFName>V35:apr26-vpn9</VRFName>
          </Naming>
          <Create>true</Create>
          <AFTable>
            <AF>
              <Naming>
                <AFName>IPv4</AFName>
                <SAFName>Unicast</SAFName>
                <TopologyName>default</TopologyName>
              </Naming>
              <Create>true</Create>
              <BGP>
                <ImportRouteTargets>
```

```
                        <RouteTargetTable>
                          <RouteTarget>
                            <Naming>
                              <Type>AS</Type>
                              <AS_XX>0</AS_XX>
                              <AS>64512</AS>
                              <ASIndex>2776</ASIndex>
                            </Naming>
                            <Enable>true</Enable>
                          </RouteTarget>
                          <RouteTarget>
                            <Naming>
                              <Type>AS</Type>
                              <AS_XX>0</AS_XX>
                              <AS>64512</AS>
                              <ASIndex>2777</ASIndex>
                            </Naming>
                            <Enable>true</Enable>
                          </RouteTarget>
                        </RouteTargetTable>
                      </ImportRouteTargets>
                      <ExportRouteTargets>
                        <RouteTargetTable>
                          <RouteTarget>
                            <Naming>
                              <Type>AS</Type>
                              <AS_XX>0</AS_XX>
                              <AS>64512</AS>
                              <ASIndex>2776</ASIndex>
                            </Naming>
                            <Enable>true</Enable>
                          </RouteTarget>
                        </RouteTargetTable>
                      </ExportRouteTargets>
                    </BGP>
                  </AF>
                </AFTable>
              </VRF>
            </VRFTable>
            <InterfaceConfigurationTable>
              <InterfaceConfiguration>
                <Naming>
                  <InterfaceName>GigabitEthernet0/15/1/1.947</InterfaceName>
                  <Active>act</Active>
                </Naming>
                <Description>GigabitEthernet0/15/1/1.947 dot1q vlan id=947. By VPNSC: Job Id# =
191</Description>
                <InterfaceModeNonPhysical>Default</InterfaceModeNonPhysical>
                <VLANSubConfiguration>
                  <VLANIdentifier>
                    <VlanType>VLANTypeDot1q</VlanType>
                    <FirstTag>947</FirstTag>
                  </VLANIdentifier>
                </VLANSubConfiguration>
                <VRF>V35:apr26-vpn9</VRF>
                <IPV4Network>
                  <Addresses>
                    <Primary>
                      <Address>26.27.28.21</Address>
                      <Netmask>255.255.255.0</Netmask>
                    </Primary>
                  </Addresses>
                </IPV4Network>
              </InterfaceConfiguration>
```

```
</InterfaceConfigurationTable>
<BGP>
  <AS>
    <Naming>
      <AS>0</AS>
    </Naming>
    <FourByteAS>
      <Naming>
        <AS>64512</AS>
      </Naming>
      <VRFTable>
        <VRF>
          <Naming>
            <VRFName>V35:apr26-vpn9</VRFName>
          </Naming>
          <VRFGlobal>
            <Exists>true</Exists>
            <RouteDistinguisher>
              <Type>AS</Type>
              <AS_XX>0</AS_XX>
              <AS>64512</AS>
              <ASIndex>2190</ASIndex>
            </RouteDistinguisher>
            <VRFGlobalAFTable>
              <VRFGlobalAF>
                <Naming>
                  <AFName>IPv4Unicast</AFName>
                </Naming>
                <Enable>true</Enable>
                <ConnectedRoutes/>
                <OSPFRouteTable>
                  <OSPFRoute>
                    <Naming>
                      <InstanceName>1611</InstanceName>
                    </Naming>
                    <RoutePolicyName/>
                    <RedistType>01</RedistType>
                    <DefaultMetric>325</DefaultMetric>
                  </OSPFRoute>
                </OSPFRouteTable>
                <StaticRoutes/>
              </VRFGlobalAF>
            </VRFGlobalAFTable>
            <VRFGlobalAFTable>
              <VRFGlobalAF>
                <Naming>
                  <AFName>IPv6Unicast</AFName>
                </Naming>
                <Enable>true</Enable>
                <StaticRoutes/>
              </VRFGlobalAF>
            </VRFGlobalAFTable>
          </VRFGlobal>
        </VRF>
      </VRFTable>
    </FourByteAS>
  </AS>
</BGP>
<OSPF>
  <ProcessTable>
    <Process>
      <Naming>
        <ProcessName>1611</ProcessName>
      </Naming>
```

```
                    <Start>true</Start>
                    <VRFTable>
                      <VRF>
                        <Naming>
                          <VRFName>V35:apr26-vpn9</VRFName>
                        </Naming>
                        <VRFStart>true</VRFStart>
                        <DefaultInformation>
                           <AlwaysAdvertise>true</AlwaysAdvertise>
                           <Metric>652</Metric>
                           <MetricType>Type2</MetricType>
                           <Policy>dinesh</Policy>
                         </DefaultInformation>
                        <AreaTable>
                          <Area>
                            <Naming>
                              <AreaID>218</AreaID>
                            </Naming>
                            <NameScopeTable>
                              <NameScope>
                                <Naming>
                                  <InterfaceName>GigabitEthernet0/15/1/1.947</InterfaceName>
                                </Naming>
                                <Running>true</Running>
                              </NameScope>
                            </NameScopeTable>
                            <Running>true</Running>
                          </Area>
                        </AreaTable>
                      </VRF>
                    </VRFTable>
                  </Process>
                </ProcessTable>
              </OSPF>
            </Configuration>
          </Set>
          <Commit/>
        </Request>
```

Comments

- None.

PE L3 MPLS VPN (EIGRP, Authentication Keychain Name, IOS XR)

Configuration

- Service: L3 MPLS VPN.
- Feature: MPLS service request using the EIGRP routing protocol and specifying a keychain name to authentic EIGRP protocol traffic on an interface.
- Device configuration:
  - The PE is an IOS XR device running IOS XR version 3.8.1 or 3.9.0.

    Interface(s): Various.
  - Routing protocol = EIGRP.

Configlets

PE

The code examples below show CLI and XML configlets. All configlets are deployed on the PE device.

Sample CLI Configlets

The following is a sample CLI configlet for an IOS XR device.

```
vrf V67:apr26-vpn2
 address-family ipv4 unicast
  import route-target
   64512:2764
   64512:2765
  !
  export route-target
   64512:2764
  !
 !
 address-family ipv6 unicast
  import route-target
   64512:2764
   64512:2765
  !
  export route-target
   64512:2764
  !
 !
!
interface TenGigE0/0/0/3.841
 description TenGigE0/0/0/3.841 dot1q vlan id=841. By VPNSC: Job Id# = 188
 vrf V67:apr26-vpn2
 ipv4 address 31.32.33.23 255.255.255.0
 ipv6 address 500::200/32
 dot1q vlan 841
!
router bgp 64512
 vrf V67:apr26-vpn2
  rd 64512:2222
  address-family ipv4 unicast
   redistribute eigrp 1324
  !
  address-family ipv6 unicast
```

```
   redistribute eigrp 1321
  !
 !
!
router eigrp 100
 vrf V67:apr26-vpn2
  address-family ipv4
   default-metric 1509 1842 196 187 1657
   autonomous-system 1324
   interface TenGigE0/0/0/3.841
    authentication keychain keychain-ipv4
   !
  !
  address-family ipv6
   default-metric 1624 1428 186 127 1095
   autonomous-system 1321
   interface TenGigE0/0/0/3.841
    authentication keychain keychain-ipv6
   !
  !
 !
!
end
```

Sample XML Configlets

The following is a sample XML configlet for an IOS XR device.

```
<?xml version="1.0" encoding="UTF-8"?>
<Request MajorVersion="1" MinorVersion="0">
 <CLI>
<Configuration>
vrf V67:apr26-vpn2
address-family ipv6 unicast
import route-target 64512:2764
import route-target 64512:2765
export route-target 64512:2764
exit
interface TenGigE0/0/0/3.841
ipv6 address 500::200/32
</Configuration>
</CLI>
<Delete>
    <Configuration Source="CurrentConfig">
      <EIGRP>
        <ProcessTable>
          <Process>
            <Naming>
              <ASNumber>100</ASNumber>
            </Naming>
            <VRFTable>
              <VRF>
                <Naming>
                  <VRFName>V67:apr26-vpn2</VRFName>
                </Naming>
                <VRF_AFTable>
                  <VRF_AF>
                    <Naming>
                      <VRF_AFType>IPv4</VRF_AFType>
                    </Naming>
                    <AutoSummary/>
                  </VRF_AF>
                  <VRF_AF>
                    <Naming>
```

```
                    <VRF_AFType>IPv6</VRF_AFType>
                  </Naming>
                  <AutoSummary/>
                </VRF_AF>
              </VRF_AFTable>
            </VRF>
          </VRFTable>
        </Process>
      </ProcessTable>
    </EIGRP>
    <InterfaceConfigurationTable>
      <InterfaceConfiguration>
        <Naming>
          <Name>TenGigE0/0/0/3.841</Name>
          <Active>act</Active>
        </Naming>
        <Shutdown>true</Shutdown>
      </InterfaceConfiguration>
    </InterfaceConfigurationTable>
  </Configuration>
</Delete>
<Set>
  <Configuration Source="CurrentConfig">
    <VRFTable>
      <VRF>
        <Naming>
          <Name>V67:apr26-vpn2</Name>
        </Naming>
        <Create>true</Create>
        <AFI_SAFITable>
          <AFI_SAFI>
            <Naming>
              <AFI>IPv4</AFI>
              <SAFI>Unicast</SAFI>
              <Topology>default</Topology>
            </Naming>
            <Create>true</Create>
            <BGP>
              <ImportRouteTargets>
                <RouteTargetTable>
                  <RouteTarget>
                    <Naming>
                      <Type>AS</Type>
                      <AS>64512</AS>
                      <ASIndex>2764</ASIndex>
                    </Naming>
                    <True>true</True>
                  </RouteTarget>
                  <RouteTarget>
                    <Naming>
                      <Type>AS</Type>
                      <AS>64512</AS>
                      <ASIndex>2765</ASIndex>
                    </Naming>
                    <True>true</True>
                  </RouteTarget>
                </RouteTargetTable>
              </ImportRouteTargets>
              <ExportRouteTargets>
                <RouteTargetTable>
                  <RouteTarget>
                    <Naming>
                      <Type>AS</Type>
                      <AS>64512</AS>
```

```
                        <ASIndex>2764</ASIndex>
                      </Naming>
                      <True>true</True>
                    </RouteTarget>
                  </RouteTargetTable>
                </ExportRouteTargets>
              </BGP>
            </AFI_SAFI>
          </AFI_SAFITable>
        </VRF>
      </VRFTable>
      <InterfaceConfigurationTable>
        <InterfaceConfiguration>
          <Naming>
            <Name>TenGigE0/0/0/3.841</Name>
            <Active>act</Active>
          </Naming>
          <Description>TenGigE0/0/0/3.841 dot1q vlan id=841. By VPNSC: Job Id# =
188</Description>
          <InterfaceModeNonPhysical>Default</InterfaceModeNonPhysical>
          <VLANSubConfiguration>
            <VLANIdentifier>
              <VlanType>VLANTypeDot1q</VlanType>
              <FirstTag>841</FirstTag>
            </VLANIdentifier>
          </VLANSubConfiguration>
          <VRF>V67:apr26-vpn2</VRF>
          <IPV4Network>
            <Addresses>
              <Primary>
                <IPAddress>31.32.33.23</IPAddress>
                <Mask>255.255.255.0</Mask>
              </Primary>
            </Addresses>
          </IPV4Network>
        </InterfaceConfiguration>
      </InterfaceConfigurationTable>
      <BGP>
        <AS>
          <Naming>
            <AS>0</AS>
          </Naming>
          <FourByteAS>
            <Naming>
              <AS>64512</AS>
            </Naming>
            <VRFTable>
              <VRF>
                <Naming>
                  <Name>V67:apr26-vpn2</Name>
                </Naming>
                <VRFGlobal>
                  <Exists>true</Exists>
                  <RouteDistinguisher>
                    <Type>AS</Type>
                    <AS>64512</AS>
                    <ASIndex>2222</ASIndex>
                  </RouteDistinguisher>
                  <VRFGlobalAFTable>
                    <VRFGlobalAF>
                      <Naming>
                        <AF>IPv4Unicast</AF>
                      </Naming>
                      <Enabled>true</Enabled>
```

```
                    <EIGRPRouteTable>
                      <EIGRPRoutes>
                        <Naming>
                          <EIGRPInstanceName>1324</EIGRPInstanceName>
                        </Naming>
                      </EIGRPRoutes>
                    </EIGRPRouteTable>
                  </VRFGlobalAF>
                </VRFGlobalAFTable>
                <VRFGlobalAFTable>
                  <VRFGlobalAF>
                    <Naming>
                      <AF>IPv6Unicast</AF>
                    </Naming>
                    <Enabled>true</Enabled>
                    <EIGRPRouteTable>
                      <EIGRPRoutes>
                        <Naming>
                          <EIGRPInstanceName>1321</EIGRPInstanceName>
                        </Naming>
                      </EIGRPRoutes>
                    </EIGRPRouteTable>
                  </VRFGlobalAF>
                </VRFGlobalAFTable>
              </VRFGlobal>
            </VRF>
          </VRFTable>
        </FourByteAS>
      </AS>
    </BGP>
    <EIGRP>
      <ProcessTable>
        <Process>
          <Naming>
            <ASNumber>100</ASNumber>
          </Naming>
          <VRFTable>
            <VRF>
              <Naming>
                <VRFName>V67:apr26-vpn2</VRFName>
              </Naming>
              <Enabled>true</Enabled>
              <VRF_AFTable>
                <VRF_AF>
                  <Naming>
                    <VRF_AFType>IPv4</VRF_AFType>
                  </Naming>
                  <Enabled>true</Enabled>
                  <RedistributeTable>
                    <Redistribute>
                      <Naming>
                        <Protocol>BGP</Protocol>
                        <SecondASNumber>64512</SecondASNumber>
                      </Naming>
                      <PolicySpecified>false</PolicySpecified>
                    </Redistribute>
                  </RedistributeTable>
                  <DefaultMetric>
                    <BW>1509</BW>
                    <Delay>1842</Delay>
                    <Reliability>196</Reliability>
                    <Load>187</Load>
                    <MTU>1657</MTU>
                  </DefaultMetric>
```

```
                        <InterfaceTable>
                          <Interface>
                            <Naming>
                              <InterfaceName>TenGigE0/0/0/3.841</InterfaceName>
                            </Naming>
                            <Enabled>true</Enabled>
                            <Authentication>
                               <Keychain>keychain-ipv4</Keychain>
                             </Authentication>
                          </Interface>
                        </InterfaceTable>
                        <AutonomousSystem>1324</AutonomousSystem>
                      </VRF_AF>
                      <VRF_AF>
                        <Naming>
                          <VRF_AFType>IPv6</VRF_AFType>
                        </Naming>
                        <Enabled>true</Enabled>
                        <RedistributeTable>
                          <Redistribute>
                            <Naming>
                              <Protocol>BGP</Protocol>
                              <SecondASNumber>64512</SecondASNumber>
                            </Naming>
                            <PolicySpecified>false</PolicySpecified>
                          </Redistribute>
                        </RedistributeTable>
                        <DefaultMetric>
                          <BW>1624</BW>
                          <Delay>1428</Delay>
                          <Reliability>186</Reliability>
                          <Load>127</Load>
                          <MTU>1095</MTU>
                        </DefaultMetric>
                        <InterfaceTable>
                          <Interface>
                            <Naming>
                              <InterfaceName>TenGigE0/0/0/3.841</InterfaceName>
                            </Naming>
                            <Enabled>true</Enabled>
                            <Authentication>
                               <Keychain>keychain-ipv6</Keychain>
                             </Authentication>
                          </Interface>
                        </InterfaceTable>
                        <AutonomousSystem>1321</AutonomousSystem>
                      </VRF_AF>
                    </VRF_AFTable>
                  </VRF>
                </VRFTable>
              </Process>
            </ProcessTable>
          </EIGRP>
        </Configuration>
      </Set>
      <Commit/>
    </Request>
```

Comments

- None.

PE L3 MPLS VPN (Independent VRF, IOS XR)

Configuration

- Service: L3 MPLS VPN.
- Feature: MPLS service request using an independent VRF on IOS XR
- Device configuration:
  - The PE is an iscind-12010-1 (GSR) with IOS XR version 3.7.1[00].

    Interface(s): GigabitEthernet0/1/0/1.
  - Routing protocol = None.

Configlets

PE and VRF

The code examples below show CLI and XML configlets for both the MPLS service request and the VRF object.

MPLS Service Request CLI Configlets

```
interface GigabitEthernet0/1/0/0.3233
 description GigabitEthernet0/1/0/0.3233 dot1q vlan id=3233. By VPNSC: Job Id# = 64
 vrf VRF112
 ipv4 address 126.112.102.102 255.255.255.0
 ipv6 address 1365::126/28
 dot1q vlan 3233
!
router bgp 100
 vrf VRF112
  address-family ipv4 unicast
  !
  address-family ipv6 unicast
  !
 !
!
multicast-routing
 vrf VRF112 address-family ipv4
  interface GigabitEthernet0/1/0/0.3233
   enable
  !
 !
 vrf VRF112 address-family ipv6
  interface GigabitEthernet0/1/0/0.3233
   enable
  !
 !
!
end
```

MPLS Service Requesets XML Configlets

```
<?xml version="1.0" encoding="UTF-8"?>
<Request MajorVersion="1" MinorVersion="0">
 <CLI>
<Configuration>
interface GigabitEthernet0/1/0/0.3233
ipv6 address 1365::126/28
multicast-routing
vrf VRF112
interface GigabitEthernet0/1/0/0.3233
enable
vrf VRF112 address-family ipv6
interface GigabitEthernet0/1/0/0.3233
enable
</Configuration>
</CLI>
<Delete>
    <Configuration Source="CurrentConfig">
      <InterfaceConfigurationTable>
        <InterfaceConfiguration>
          <Naming>
            <Name>GigabitEthernet0/1/0/0.3233</Name>
            <Active>act</Active>
          </Naming>
          <Shutdown>true</Shutdown>
        </InterfaceConfiguration>
      </InterfaceConfigurationTable>
    </Configuration>
  </Delete>
  <Set>
    <Configuration Source="CurrentConfig">
      <InterfaceConfigurationTable>
        <InterfaceConfiguration>
          <Naming>
            <Name>GigabitEthernet0/1/0/0.3233</Name>
            <Active>act</Active>
          </Naming>
           <Description>GigabitEthernet0/1/0/0.3233 dot1q vlan id=3233. By VPNSC: Job Id# =
64</Description>
          <InterfaceModeNonPhysical>Default</InterfaceModeNonPhysical>
          <VLANSubConfiguration>
            <VLANIdentifier>
              <VlanType>VLANTypeDot1q</VlanType>
              <FirstTag>3233</FirstTag>
            </VLANIdentifier>
          </VLANSubConfiguration>
          <VRF>VRF112</VRF>
          <IPV4Network>
            <Addresses>
              <Primary>
                <IPAddress>126.112.102.102</IPAddress>
                <Mask>255.255.255.0</Mask>
              </Primary>
            </Addresses>
          </IPV4Network>
        </InterfaceConfiguration>
      </InterfaceConfigurationTable>
      <BGP>
        <AS>
          <Naming>
            <AS>0</AS>
          </Naming>
```

```
          <FourByteAS>
            <Naming>
              <AS>100</AS>
            </Naming>
            <VRFTable>
              <VRF>
                <Naming>
                  <Name>VRF112</Name>
                </Naming>
                <VRFGlobal>
                  <Exists>true</Exists>
                  <VRFGlobalAFTable>
                    <VRFGlobalAF>
                      <Naming>
                        <AF>IPv4Unicast</AF>
                      </Naming>
                      <Enabled>true</Enabled>
                    </VRFGlobalAF>
                  </VRFGlobalAFTable>
                  <VRFGlobalAFTable>
                    <VRFGlobalAF>
                      <Naming>
                        <AF>IPv6Unicast</AF>
                      </Naming>
                      <Enabled>true</Enabled>
                    </VRFGlobalAF>
                  </VRFGlobalAFTable>
                </VRFGlobal>
              </VRF>
            </VRFTable>
          </FourByteAS>
        </AS>
      </BGP>
    </Configuration>
  </Set>
  <Commit/>
</Request>
```

VRF Service Request CLI Configlets

```
vrf VRF112
 address-family ipv4 unicast
  import route-target
   100:19890
   100:19891
  !
  export route-target
   100:19890
  !
 !
 address-family ipv6 unicast
  import route-target
   100:19890
   100:19891
  !
  export route-target
   100:19890
  !
 !
!
router bgp 100
 vrf VRF112
  rd 112.101.112.101:1263
```

```
 !
!
multicast-routing
 vrf VRF112 address-family ipv4
  mdt mtu 8025
  mdt data 224.10.0.9/32 threshold 8024
  mdt default ipv4 224.10.0.8
 !
 vrf VRF112 address-family ipv6
  mdt mtu 8025
  mdt default ipv4 224.10.0.8
 !
!
router pim vrf VRF112 address-family ipv4
 rp-address 112.101.122.102 list1
!
router pim vrf VRF112 address-family ipv6
 rp-address 1253::214 list2
!
end
```

VRF Service Request XML Configlets

```
<?xml version="1.0" encoding="UTF-8"?>
<Request MajorVersion="1" MinorVersion="0">
 <CLI>
<Configuration>
vrf VRF112
address-family ipv6 unicast
import route-target 100:19890
import route-target 100:19891
export route-target 100:19890
exit
multicast-routing
vrf VRF112
mdt default 224.10.0.8
mdt data 224.10.0.9/32 threshold 8024
mdt mtu 8025
vrf VRF112 address-family ipv6
mdt default 224.10.0.8
mdt mtu 8025
router pim vrf VRF112 address-family ipv4 rp-address 112.101.122.102 list1
router pim vrf VRF112 address-family ipv6 rp-address 1253::214 list2
</Configuration>
</CLI>
<Set>
    <Configuration Source="CurrentConfig">
      <VRFTable>
        <VRF>
          <Naming>
            <Name>VRF112</Name>
          </Naming>
          <Create>true</Create>
          <AFI_SAFITable>
            <AFI_SAFI>
              <Naming>
                <AFI>IPv4</AFI>
                <SAFI>Unicast</SAFI>
                <Topology>default</Topology>
              </Naming>
              <Create>true</Create>
              <BGP>
```

```
            <ImportRouteTargets>
              <RouteTargetTable>
                <RouteTarget>
                  <Naming>
                    <Type>AS</Type>
                    <AS>100</AS>
                    <ASIndex>19890</ASIndex>
                  </Naming>
                  <True>true</True>
                </RouteTarget>
                <RouteTarget>
                  <Naming>
                    <Type>AS</Type>
                    <AS>100</AS>
                    <ASIndex>19891</ASIndex>
                  </Naming>
                  <True>true</True>
                </RouteTarget>
              </RouteTargetTable>
            </ImportRouteTargets>
            <ExportRouteTargets>
              <RouteTargetTable>
                <RouteTarget>
                  <Naming>
                    <Type>AS</Type>
                    <AS>100</AS>
                    <ASIndex>19890</ASIndex>
                  </Naming>
                  <True>true</True>
                </RouteTarget>
              </RouteTargetTable>
            </ExportRouteTargets>
          </BGP>
        </AFI_SAFI>
      </AFI_SAFITable>
    </VRF>
  </VRFTable>
  <BGP>
    <AS>
      <Naming>
        <AS>0</AS>
      </Naming>
      <FourByteAS>
        <Naming>
          <AS>100</AS>
        </Naming>
        <VRFTable>
          <VRF>
            <Naming>
              <Name>VRF112</Name>
            </Naming>
            <VRFGlobal>
              <Exists>true</Exists>
              <RouteDistinguisher>
                <Type>IPV4Address</Type>
                <Addr>112.101.112.101</Addr>
                <AddrIndex>1263</AddrIndex>
              </RouteDistinguisher>
            </VRFGlobal>
          </VRF>
        </VRFTable>
      </FourByteAS>
    </AS>
  </BGP>
```

```
    </Configuration>
  </Set>
  <Commit/>
</Request>
```

Comments

- This service request uses the MPLS VPN PE_NO_CE policy.
- This service request has multicast IPv4 and IPv6 enabled VPN and also static RPs, as shown in the in the configlets.

PE L3 MPLS VPN (Independent RTs for IPv4 and IPv6, IOS XR)

Configuration

- Service: L3 MPLS VPN.
- Feature: MPLS service request using independent RTs for IPv4 and IPv6.
- Device configuration:
  - The PE is an iscind-12010-1 (GSR) with IOS XR version 3.7.1[00].

    Interface(s): Various.
  - Routing protocol = None.

Configlets

PE

The code examples below show CLI and XML configlets for the specified independent RT configurations, as noted. All configlets are deployed on the PE device.

Sample CLI Configlets

The following examples show CLI configlets for the specified independent RT configurations.

Example 1: CE-PE with CERC Type set as IPv4.

```
address-family ipv4 unicast
  import route-target
   7777:12345
  export route-target
   7777:12345
 address-family ipv6 unicast
```

Note If the CERC were tagged as IPv6, the RTs would be configured under **ipv6 address-family**.

Example 2: PE-CE with CERC Type set as IPv4+IPv6.

```
address-family ipv4 unicast
  import route-target
   7777:12345
  export route-target
   7777:12345
address-family ipv6 unicast
  import route-target
   7777:123456
  export route-target
   7777:123456
```

Note If there were additional IPv4 or IPv6 CERCs selected and tagged, they would be incrementally added into the above format under the appropriate **address-family** CLIs.

Example 3: Adding More VPNs

When adding more VPNs to the configuration, then one VPN name shows up in the configlet with the the string **-etc** appended, as shown below.

```
vrf V872:vpn2-etc
address-family ipv4 unicast
```

```
import route-target
64512:1005
!
export route-target
64512:1005
!
!
```

Sample XML Configlets

The following is a sample XML configlet for PE-CE with CERC Type set as IPv4+IPv6. Key XML tags are shown in bold text.

```
<?xml version="1.0" encoding="UTF-8"?>
<Request MajorVersion="1" MinorVersion="0">
 <CLI>
<Configuration>
vrf V6:Verve_VPN32
address-family ipv6 unicast
import route-target <?xml version="1.0" encoding="UTF-8"?>
<Request MajorVersion="1" MinorVersion="0">
 <CLI>
<Configuration>
vrf V6:Verve_VPN32
address-family ipv6 unicast
import route-target 64512:25428
import route-target 64512:25429
export route-target 64512:25428
exit
interface GigabitEthernet0/3/0/2.3039
ipv6 address 10::12/24
ipv6 address 10::15/32
ipv6 address 15::20/28
</Configuration>
</CLI>
  <Set>
    <Configuration Source="CurrentConfig">
      <VRFTable>
        <VRF>
          <Naming>
            <Name>V6:Verve_VPN32</Name>
          </Naming>
          <Create>true</Create>
          <AFI_SAFITable>
            <AFI_SAFI>
              <Naming>
                <AFI>IPv4</AFI>
                <SAFI>Unicast</SAFI>
                <Topology>default</Topology>
              </Naming>
              <Create>true</Create>
              <BGP>
                <ImportRouteTargets>
                  <RouteTargetTable>
                    <RouteTarget>
                      <Naming>
                        <Type>AS</Type>
                        <AS>64512</AS>
                        <ASIndex>254288</ASIndex>
                      </Naming>
                      <True>true</True>
                    </RouteTarget>
                    <RouteTarget>
                      <Naming>
```

```
                        <Type>AS</Type>
                        <AS>64512</AS>
                        <ASIndex>254299</ASIndex>
                      </Naming>
                      <True>true</True>
                    </RouteTarget>
                  </RouteTargetTable>
                </ImportRouteTargets>
                <ExportRouteTargets>
                  <RouteTargetTable>
                    <RouteTarget>
                      <Naming>
                        <Type>AS</Type>
                        <AS>64512</AS>
                        <ASIndex>254288</ASIndex>
                      </Naming>
                      <True>true</True>
                    </RouteTarget>
                  </RouteTargetTable>
                </ExportRouteTargets>
              </BGP>
            </AFI_SAFI>
          </AFI_SAFITable>
        </VRF>
      </VRFTable>
    </Configuration>
  </Set>
</Request>
```

Comments

- None.

PE L3 MPLS VPN (Bundle-Ether Interface, IOS XR)

Configuration

- Service: L3 MPLS VPN.
- Feature: MPLS service request using a Bundle-Ethernet interface.
- Device configuration:
  - The PE is an iscind-12010-1 (GSR) with IOS XR version 3.7.1[00].

    Interface(s): Bundle-Ether147.
  - Routing protocol = None.

Configlets

PE

The code examples below show CLI and XML configlets for a Bundle-Ethernet interface, as noted. All configlets are deployed on the PE device.

Sample CLI Configlets

The following example is a CLI configlet for the bundle interface feature. The configlet is deployed on the PE device.

```
interface Bundle-Ether147
 description Bun
!
interface Bundle-Ether147.369
 description subbun
 vrf ISC521
 ipv4 address 66.174.25.3 255.255.255.254
 ipv6 address 2001:4888:10:100::3/64
 dot1q vlan 269
!
```

Sample XML Configlets

The following is a sample XML configlet for the bundle interface feature. The configlet is deployed on the PE device.

```
<InterfaceConfiguration>
     <Naming>
    <Active>act</Active>
      <Name>Bundle-Ether147</Name>
        </Naming>
        <InterfaceVirtual>true</InterfaceVirtual>
    <Description>Bun</Description>
 </InterfaceConfiguration>

<InterfaceConfiguration>
     <Naming>
    <Active>act</Active>
    <Name>Bundle-Ether147.369</Name>
     </Naming>
     <InterfaceModeNonPhysical>Default</InterfaceModeNonPhysical>
    <Description>subbun</Description>
    <VRF MajorVersion="3" MinorVersion="3">ISC521</VRF>
    <IPV4Network MajorVersion="5" MinorVersion="0">
       <Addresses>
```

```
        <Primary>
           <IPAddress>66.174.25.3</IPAddress>
            <Mask>255.255.255.254</Mask>
        </Primary>
       </Addresses>
       </IPV4Network>
    <VLANSubConfiguration MajorVersion="2" MinorVersion="1">
        <VLANIdentifier>
        <VlanType>VLANTypeDot1q</VlanType>
         <FirstTag>269</FirstTag>
        </VLANIdentifier>
   </VLANSubConfiguration>
</InterfaceConfiguration>
```

Comments

- None.

PE L3 MPLS VPN (Outgoing Interface + Next Hop IP Address, Static Route Configuration, IOS XR and IOS)

Configuration

- Service: L3 MPLS VPN.
- Feature: MPLS service request using the static routing protocol and specifying an outgoing interface and next hop IP address.
- Device configuration:
  - The PE is an iscind-12010-1 (GSR) with IOS XR version 3.7.1[00].

    Interface(s): Various.
  - Routing protocol = Static.

Configlets

PE

The code examples below show CLI and XML configlets. All configlets are deployed on the PE device.

Sample CLI Configlets

The following is a sample CLI configlet for an IOS device.

```
router bgp 64512
address-family ipv4 vrf V14:July7_VPN
redistribute static
exit-address-family
!
ip route vrf V14:July7_VPN 15.18.16.17 255.255.255.255  GigabitEthernet0/3/0/010.12.16.19
78
```

The following is a sample CLI configlet for an IOS XR device.

```
router static
 vrf V7:techm_vpn
  address-family ipv4 unicast
   12.23.34.34/32  GigabitEthernet0/3/0/210.14.54.18 45
 !
  address-family ipv6 unicast
   15:16:17:13:14:15:17:18/128 GigabitEthernet0/3/0/2 18:12:13:14:16:13:16:14
  !
```

Sample XML Configlets

The following is a sample XML configlet for IPv4 address family.

```
<VRF>
  <Naming>
   <VRFName>V1:VPN_June22</VRFName>
  </Naming>
  <AddressFamily>
   <VRFIPV4>
     <VRFUnicast>
      <VRFPrefixTable>
        <VRFPrefix>
         <Naming>
           <Prefix>
            <IPV4Address>10.77.66.58</IPV4Address>
```

```
          </Prefix>
          <Length>32</Length>
        </Naming>
        <VRFRouteTable>
          <VRFNexthopInfoTable>
           <VRFNexthopInfo>
             <Naming>
              <Interface>GigabitEthernet0/3/0/0</Interface>
                <Address>
                 <IPV4Address>10.12.16.19</IPV4Address>
                </Address>
             </Naming>
             <Metric>48</Metric>
           </VRFNexthopInfo>
          </VRFNexthopInfoTable>
        </VRFRouteTable>
      </VRFPrefix>
      </VRFPrefixTable>
     </VRFUnicast>
   </VRFIPV4>
  </AddressFamily>
</VRF>
```

The following is a sample XML configlet for IPv6 address family.

```
<VRF>
 <Naming>
  <VRFName>V39:techm_vpn</VRFName>
 </Naming>
 <AddressFamily>
  <VRFIPV6>
    <VRFUnicast>
     <VRFPrefixTable>
      <VRFPrefix>
       <Naming>
        <Prefix>
         <IPV6Address>10::19</IPV6Address>
        </Prefix>
         <Length>128</Length>
       </Naming>
       <VRFRouteTable>
        <VRFNexthopInfoTable>
         <VRFNexthopInfo>
          <Naming>
           <Interface>GigabitEthernet0/3/0/0</Interface>
           <Address>
            <IPV6Address>45::10</IPV6Address>
           </Address>
          </Naming>
          <Metric>75</Metric>
         </VRFNexthopInfo>
        </VRFNexthopInfoTable>
       </VRFRouteTable>
      </VRFPrefix>
     </VRFPrefixTable>
    </VRFUnicast>
   </VRFIPV6>
 </AddressFamily>
</VRF>
```

Comments

- None.

Troubleshooting MPLS VPNs

This section provides information about troubleshooting MPLS VPNs.

General Troubleshooting Guidelines

For general troubleshooting of failed provisioning, perform the following steps:

Step 1 Identify the failed service request and go into **Details**.

a. To do this, go to the Service Request Editor and click **Details**.

Of main concern is the status message—this tells you exactly what happened.

b. If the status message tells you it's a failed audit, click the **Audit** button to find out exactly what part of the audit failed.

Step 2 If the troubleshooting sequence in Step 1 does not give you a clear idea as to what happened, use the logs in the Task Manager to identify the problem.

a. To do this, choose **Monitoring > Task Manager > Logs > Task Name**.

b. There is a lot of information in this log. To isolate the problem, you can use the filter. If you filter by log level and/or component, you can usually reduce the amount of irrelevant information and focus on the information you must know to locate the problem.

Step 3 Also see the section Frequently Asked Questions, page 5-247 in this appendix for information on some common questions and issues.

Gathering Logs for Development Engineering

Go through the troubleshooting steps described in General Troubleshooting Guidelines, page 5-246. If you have failed to troubleshoot or identify the problem, this section provides information on how to gather logs for the development engineer to troubleshoot.

Tip The logs apply to both MPLS VPNs and Layer 2 VPNs.

There is a property in DCPL called **Provisioning.Service.mpls.saveDebugData**. If this property is set to **True**, whenever a service request is deployed, a temporary directory is created in PRIMEF_HOME/tmp/mpls.

The directory contains the job ID of the service request prefixed to it, along with a time stamp. This directory contains the uploaded configuration files, service parameters in XML format, and the provisioning and audit results.

The default is set to True.

To verify, perform the following steps:

Step 1 Locate the property by choosing **Administration > Control Center**.

The Control Center Hosts page appears.

Step 2 Check the check box for the host of interest.

The menu buttons for the Hosts page are enabled.

Step 3 Click **Config**.

The Host Configuration window appears.

Step 4 Navigate to **Provisioning > mpls**.

Step 5 Click **saveDebugData** to save the data to a temporary directory for debugging purposes.

Frequently Asked Questions

Below is a list of FAQs concerning MPLS VPN provisioning.

What is the MPLS provisioning workflow?

The tasks listed below depict the MPLS provisioning workflow. This section assumes an operator deploys a service request using a caller such as Task Manager.

1. The Provisioning driver (ProvDrv) gets the service request to be deployed.
2. From the service request, the Provisioning driver deduces which devices are involved.
3. The latest router configurations must be obtained, so the Provisioning driver tells the Generic Transport Library (GTL)/ Device Configuration Service (DCS) to upload the latest router configurations. The result is used by the service module.
4. The Provisioning driver determines what service modules are involved based on the service and device types.
5. The Provisioning driver queries the Repository for the service intention. The Provisioning driver sends the service intention to the service module, along with the uploaded configuration.
6. The service module generates configlets based on the configurations and service intention and returns the appropriate configlets to the Provisioning driver.
7. The Provisioning driver signals GTL/DCS to download the configlets to the target routers.
8. The Provisioning driver sends the updated result, including the download result, to the Repository, which then updates its state.

Definitions of terms mentioned in the above steps.

- **Device Configuration Service (DCS)**: Responsible for uploading and downloading configuration files.
- **Generic Transport Library (GTL)**: Provides APIs for downloading configlets to target devices, uploading configuration files from target devices, executing commands on target devices, and reloading the target device.

  This library provides a layer between the transport provider (DCS) and the client application (for example, the Provisioning Driver, Auditor, Collect Config operation, Exec command). The main role of the GTL is to collect the target specific information from the Repositories and the *properties* file and pass it on to the transport provider (DCS).
- **ProvDrv (the Provisioning driver)**: ProvDrv is the task responsible for deploying one or more services on multiple devices.

ProvDrv performs the tasks that are common to all services, such as the just-in-time upload of configuration files from the devices, invocation of the Data Driven Provisioning (DDP) engine, obtaining the generated configlets or the audit reports from the DDP engine, and downloading the configlets to the devices.

- **Repository**: The Repository houses various IP Solution Center data. The Prime Fulfillment Repository uses Sybase or Oracle.
- **Service module**: Generates configlets based on the service types.

What do I do if my task does not execute even if I schedule it for immediate deployment?

This problem is likely due to one of the Prime Fulfillment servers being stopped or disabled.

To check the status of all Prime Fulfillment servers, perform the following steps:

Step 1 Open the Host Configuration dialog by going to **Administration > Control Center**.

The Control Center Hosts page appears.

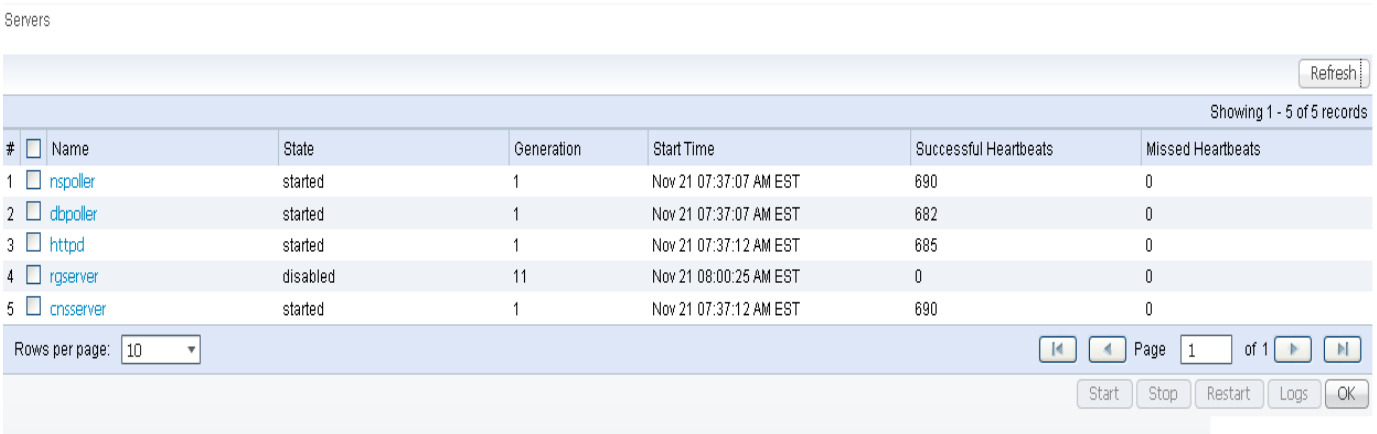
Step 2 Check the check box for the host of interest.

The menu buttons for the Hosts page are enabled.

Step 3 Choose **Servers**.

The Server Status page appears, as shown in Figure 5-38.

Figure 5-38 Prime Fulfillment Server Status



Step 4 On the Prime Fulfillment server, use the **wdclient status** command to find out the detailed status of the server.

What do I do when a service request is in the Wait Deployed state?

This concerns the devices that are configured to use Cisco Configuration Engine as the access method. If the devices are offline and a configlet was generated for it, the service request will move into the Wait Deployed state. As soon as the devices come online, the list of configlets will be downloaded and the status of the device will change.

What do I do when a service request is in the Failed Audit state?

At least one command is missing on the device. Perform the following steps:

Step 1 From the Prime Fulfillment user interface, go to **Service Request Editor > Audit > Audit Config**.

Step 2 Check the list of commands that are missing for each device.

Step 3 Look for any missing command that has an attribute with a default value.

What do I do if the service request is in the same state as it was before a deployment?

If after a deployment a service request state remains in its previously nondeployed state (Request, Invalid, or Pending), it's an indication that the provisioning task did not complete successfully. Use the steps described in General Troubleshooting Guidelines, page 5-246 to find out the reason for the service request failure.

What do I do if I receive the following out-of-memory error: OutOfMemoryError?

Perform the following steps:

Step 1 Open the Host Configuration dialog by choosing **Administration > Control Center**.

The Control Center Hosts page appears.

Step 2 Check the check box for the host of interest.

The menu buttons for the Hosts page are enabled.

Step 3 Click **Config**.

The Host Configuration window appears.

Step 4 Navigate to **watchdog > servers > worker > java > flags**.

Step 5 Change the following attribute:

Change the **Xmx256M** attribute to **Xmx384M** or **Xmx512M**.

What do I do if Prime Fulfillment will not remove a route target import/export for a VPN?

Scenario: When an MPLS service request is edited to be associated to a new VPN, the old VPN will only be removed if it is associated with only one interface. The relationship between the service request and the customer is via the VPN. The optional Customer field in a service request does not have any bearing on configuration. For example, if an MPLS service request for *custA* exists with *vpnB/cercB*, but needs to be modified to reflect *vpnA/cercA*, modifying the service request to use *vpnA/cercA* will not remove the route target for *vpnB* from the *vrfB* if there is more than one interface associated with the same VRF.

Recommended Action Running the same scenario with only one interface referring to *vrfB*, Prime Fulfillment will remove *vrfB* and correctly add *vrfA* with route target *A*.

Why does my service request go to Invalid when I choose provisioning of an extra CE Loopback interface?

It is possible that the auto pick option of the IP addresses was selected for the service request, but a /32 IP address pool was not defined. Check and make sure the IP address and the IP address pool defined for this service request are compatible.

When saving a service request, why does it say "CERC not initialized"?

It is necessary to pick a CERC for the link to join. Please check the service request to see if a CERC was selected.

Why does creation of a VLAN ID pool require an Access Domain?

VLAN ID pools are associated with an Access Domain. Access Domains model a bridged domain; VLAN IDs should be unique across a Bridged Domain.

PE-POPs must be associated with an Access Domain. An Access Domain can have more than one PE-POP associated with it.

In a Paging table, why are the Edit and Delete options disabled, even though only one check box is checked?

This is possible if one or more check boxes are selected in previous windows.

Why can I not edit an MPLS VPN or L2VPN policy?

If a service request is associated with a policy, that policy can no longer be edited.

I am unable to create a CERC—can you explain why?

You have to define a Route Target pool before you create a CERC, unless you specify the Route Targets manually.

How can I modify the configlet download order between the PE, CE, and PE-CLE devices?

There is a property called **Provisioning.Services.mpls.DownloadWeights.*** that allows you to specify the download order for the following device types: PE, CE, PE-CLE, and MVRF CE.

For example, to ensure that the configlet is downloaded to the PE before it is downloaded to the CE, configure the **Provisioning.Services.mpls.DownloadWeights.weightForPE** property with a weight value greater than that of the CE.

What does the property Provisioning.Service.mpls.reapplyIpAddress do?

If this property is set to True, during deployment of a decommissioned service request, this property will keep the IP address on the CE and PE intact on the router to maintain IPv4 connectivity to the CE.

When I create a multi-hop NPC between a CE and PE through at least one PE-CLE device, why do I see some extra NPCs created?

IP Solution Center creates the extra NPCs to prevent operators from having to enter the same information again. A CE can now be connected to the PE-CLE device, and a new NPC will be created that will connect the new CE to a PE over the PE-CLE-to-PE NPC link.

During service request provisioning, in the Interface selection list box, why don't I see the entire list of interfaces on the device?

This is probably due to a particular interface type being specified in the service policy. If that is the case, only interfaces of the specified interface type are displayed.

Why does my service request go to Invalid with the message "loopback address missing"?

This is a Layer 2 VPN question.

This is because the loopback address required to peer the pseudowire between PEs has not been defined in the PE-POP object in Prime Fulfillment.

What is the intent of the Allocate New Route Distinguisher check box in the MPLS policy?

There were some behavior changes implemented in Prime Fulfillment that differ from the legacy product "VPNSC". In VPNSC, VRFs were PE centric. Therefore, the behavior was for a new VRF to be configured for each VPN on a PE router. This behavior was modified in Prime Fulfillment to make VRFs VPN centric. For most of routing, the VRF/route distinguisher (RD) is only PE significant, except when doing iBGP load balancing. For this reason, it is possible to use the same values for a single VPN on all PE routers. This is more convenient for the user in context of troubleshooting, reporting, etc.

To increase flexibility for users where there is iBGP load balancing and also to address custom solutions and needs, there are two options available in Prime Fulfillment. One is VRF and RD Overwrite, and the other is Allocate New Route Distinguisher. VRF and RD Overwrite is exactly like it sounds. This gives the user the ability to force the VRF name and RD values for a link being provisioned. This is useful for joining a pre-existing VRF that was not provisioned by Prime Fulfillment.

Note Once you specify values to sub-attributes under the VRF and RD Overwrite attribute (that is, the VRF Name and RD Value attributes) and save an MPLS service request, both of these fields are disabled and are no longer editable. This behavior was introduced because changing the default values for the VRF Name and RD Value can alter or disable currently running service requests. Therefore, if these values need to be changed on a deployed service request, the workaround is that you must decommission and purge the service request and create a new service request. In the case of a new service request that has not yet been deployed, you must force purge the service request and then create a new service with new values.

The second option, Allocate New Route Distinguisher, is only valid for configuring a new VRF and RD on a PE router for the first time. This mimics the VPNSC behavior of individual VRFs per PE router. The following is the rule for new RD when a pre-existing VPNSC repository is not involved:

When Allocate New Route Distinguisher is enabled:

- Create a new VRF if there is no matching VRF configuration on that PE.
- If there is matching VRF configuration on that PE, then reuse it.

When Allocate New Route Distinguished is disabled:

- Find the first matching VRF configuration across the whole range of PEs, regardless of the PE, if this VRF is found on the PE being configured, reuse it. If it is not found on the PE create it.
- Note: The service request might get a VRF that has already been configured on another PE router.

An issue with pre-existing VRFs that were configured under VPNSC is that in VPNSC the Allocate New Route Distinguisher flag was always turned on. Thus, when you apply the flag again, Prime Fulfillment first looks for an existing VRF on the PE. It uses that VRF (in this case, the one provisioned by VPNSC). If no VRF is found, Prime Fulfillment creates a new VRF. When adding a new link to old VPNSC links, if the Allocate New Route Distinguisher flag is not turned on, Prime Fulfillment finds the first matching VRF configured across the network. If the PE does not have this VRF, Prime Fulfillment will create it on the router.

Use cases:

1. When adding a link to an existing PE with a legacy (VPNSC) VRF, you must select the Allocate New Route Distinguisher option.
2. When adding a link to a new PE, if you desire VRF/RD values that have not been configured before in this VPN, then you must select the Allocate New Route Distinguisher option.
3. When adding a new link to a new PE, if you want to reuse a VRF/RD value that has been used elsewhere in the network, then you must select the VRF and RD Overwrite option.
4. If you provisioned a link that has incorrect VRF/RD values (that is, not matching those previously provisioned by VPNSC), the link will need to be modified and redeployed. During the modification, you must select the VRF and RD Overwrite option and specify the same VRF/RD values used in VPNSC.
5. If you are planning to deploy iBGP load balancing across multiple PEs, the Allocate New Route Distinguisher option should be always enabled. This is to make sure the condition for unique RD is met, in order to satisfy load balancing requirements.

How can an MPLS service request using standard UNI ports allow CDP packets?

By default, an MPLS service request creates MAC ACLs for a standard UNI that restricts access of BPDU handling on the Layer2 control plane. The created ACLs are similar to the following:

```
interface FastEtherent0/15
mac access-group ISC-$name in
mac access-list extended ISC-$name

deny any host 0180.c200.0000 ===> PVST, MSTP, RSTP, and STP
deny any host 0100.0ccc.cccd ===> PVST+
deny any host 0100.0ccc.cccc ===> CDP, VTP, DTP, UDLD, PAgP
deny any host 0100.0ccd.cdd0 ===> CDP,VTP,STP
permit any any
```

Note The text appearing after "===>" is not part of the MAC ACL. It is a list of which protocols are blocked by each MAC address.

Alternatively, when the MPLS service request is created, you can edit the link attributes and perform the following steps:

Step 1 Enable **Use Existing ACL Name**.

This will enable the Port-Based ACL Name option

Step 2 Enter an empty or non-existing MAC ACL name.

When the MPLS service request is deployed, it will no longer issue the default BPDU filtering MAC ACLs. Instead, it will create an **access-group** command on the UNI interface that points to an empty ACL. Example:

```
interface FastEthernet0/15 mac access-group {$PACL_NAME} in
```

No MAC ACL is created.

Is it possible to use 2 or 3 address pools when creating an L3 VPN?

Imagine that you have IP pool 10.10.10.0/24 assigned to a region, and a PE is assigned to this region. What if one customer is using the same subnet in his LAN range? This forces you to use another subnet for the PE-CE link. How is this handled by Prime Fulfillment? The only way is to do it manually, without using auto pick. Prime Fulfillment does not support for the use of different address pools for different customers.

Another related issue is as follows. If a customer is using the same IP addresses inside his LAN segment as are used in the Prime Fulfillment pool of IP addresses, this causes a problem. For this reason, you must have multiple subnets for the PE-CE IP addresses, and use the suitable one (one that does not conflict with the IP addresses used by the customer). When you create an IP address pool, the repository knows the range, and will not allow you to use overlapping IP addresses as part of the pool.
Prime Fulfillment does not have any support for different pools to be used within the same PE.
Prime Fulfillment allows you to create multiple pools, but you can only use one based on the provider region. Prime Fulfillment picks up the next in line if the first pool runs out of IP addresses. There is no selection mechanism for you to select which pool will be used with auto pick. You can use manually added IP addresses, as long as the IP address do not overlap with the pool.

When will an IP address from the MPLS IP address pool be returned to the available pool after the service request is decommissioned?

When a service request is decommissioned, the IP address is returned back to the available pool after the service request goes to the DEPLOYED state. Prime Fulfillment prevents reuse of the returned IP addresses by a new service request for about twenty-four hours. The same behavior applies when the service request is decommissioned and then purged.

Why doesn't Prime Fulfillment remove some of the router BGP/EIGRP commands when a service request is decommissioned?

Prime Fulfillment removes the address family CLIs from router BGP or EIGRP configurations if and only if the VRF is removed. For router EIGRP, the process is not removed due to the potential presence of other CLIs that were not configured by Prime Fulfillment. This is particularly applicable when the network statement was added outside of Prime Fulfillment. Prime Fulfillment does not remove the redistribution from other routing protocols under EIGRP because the redistribute command might not be created specifically for the link.

Prime Fulfillment only removes the router OSPF process if the VRF is removed. This applies only for a PE. For a CE, router OSPF is removed if the network statement is removed. Prime Fulfillment does not remove router BGP nor router EIGRP.

What happens if the platform or IOS (or IOS XR) version does not support Q-in-Q (for example WS-X6724-SFP)?

The service request will result in a **Failed Deploy** state, and the log file will be similar to the following

For IOS:

```
SEVERE Provisioning.ProvDrvDownload failed for device NPE-1: 315 : Error downloading
cmd=[encapsulation dot1Q 158 second-dot1q 1510], response=[encapsulation dot1Q 158
second-dot1q 1510^
% Invalid input detected at '^' marker.NPE-1(config-subif)#]
```

For IOS XR:

```
SEVERE Provisioning.ProvDrvDownload failed for device NPE-1: 315 : Error downloading
cmd=[encapsulation dot1Q 158 1510], response=[encapsulation dot1Q 158 1510^
% Invalid input detected at '^' marker.NPE-1(config-subif)#]
```

Edit the service request, disable second VLAN ID, and then re-deploy.

Why doesn't Prime Fulfillment provision Q-in-Q , although the hardware/IOS does support Q-in-Q?

Possible errors:

- The port is in switchport mode. Solution: Check the port configuration, and if necessary, run **no switchport**.
- The SVI flag is enabled. Solution: Disable SVI.

Why does a port with existing subinterfaces (Q-in-Q) plus SVI on same interface result in INVALID?

If you modify a service request with only one sub interface to SVI enabled, then the service request goes to the Deployed state (in the case of an IOS device). If you create a new service request with the same interface (that is, an existing subinterface) with SVI enabled, the service request goes to the Invalid state.

Is it possible to deploy single dot1q and Q-in-Q service requests under the same interface/port?

Yes.

How can I remove the second VLAN ID from a service request that is Deployed with Q-in-Q?

You must edit/modify the service request, remove the second VLAN ID entry, and redeploy the service request. A configlet like the following will be created:

```
interface GigabitEthernet2/0/15.158
no encapsulation dot1Q
encapsulation dot1Q 158
ip address 10.1.1.105 255.255.255.252
```

VRFs

There are two VPN routing and forwarding (VRF) models.

In the traditional VRF model, the operator first creates a VPN object and then associates it to an MPLS VPN link. The necessary VRF information is generated and deployed at the time the MPLS VPN link is provisioned. The VRF information is removed only when the last link associated with the VRF is decommissioned.

The independent VRF management feature allows you to have the VRF information provisioned independent of the physical link. You can create, modify, and delete VRF objects independently of MPLS VPN links. This provides the following advantages:

- VRF information and templates can be directly deployed on a PE device without being associated with an interface.
- VRF information can exist without links pointing to it.
- A VRF object can be modified, even if it is associated with links.
- Route targets (RTs) can be added and removed without causing outages.

Managing VRFs independently of physical links involves the following tasks:

- Creating, modifying, and deleting VRF objects.
- Creating, modifying, deploying, decommissioning, and deleting a new type of service request, called a VRF service request.
- Using deployed VRF objects with MPLS VPN links via service policies and service requests.
- Migrating traditional MPLS VPN service requests to the independent VRF model.

This section describes how you can create and manage independent VRF objects. This section includes the following:

- Creating a VRF, page 5-255
- Editing VRFs, page 5-257

Creating a VRF

After you create a VRF object, you can provision it using a VRF service request, as explained in the *Cisco Prime Fulfillment User Guide 6.2*.

To create a VRF, follow these steps:

Step 1 Choose **Inventory > Logical Inventory > VRF**.

Step 2 Click **Create**.

The Create VRF window appears.

Step 3 Complete the fields as required for the VRF:

a. **Name** (required)—Enter the name of the VRF, any name of your choice. This name is directly deployed on the PE device.

b. **Provider** (required)—To select the provider associated with this VRF, choose **Select**.

c. From the list of providers, select the appropriate provider, and then click **Select**.

d. **Description** (optional)—Enter a description, if you choose.

e. **Route Targets** (required)—Click the **Select** button.

f. From the list of Route Targets, choose only one appropriate Route Target, and then click **Select**.

g. **Import RT List**—Enter one or more Route Targets (RTs) to be imported in the VRF. For multiple RTs, separate the RTs by commas. An example RT list is: 100:120,100:130,100:140.

h. **Export RT List**—Enter one or more Route Targets (RTs) to be exported from the VRF. For multiple RTs, separate the RTs by commas.

i. **Import Route Map**—Enter the name of a route map defined on the device. Prime Fulfillment validates this name while provisioning the VRF and generates an error if the route map is not defined.

j. **Export Route Map**—Enter the name of a route map defined on the device. Prime Fulfillment validates this name while provisioning the VRF and generates an error if the route map is not defined.

k. **Maximum Routes**—Specify an integer that indicates the maximum number of routes that can be imported into the VRF. The range for IOS devices is from 1 - 4294967295, and the range for IOS XR devices is from 32 - 2000000. Device type specific validations occur during service request creation.

l. **Threshold**—Specify the threshold value, which is a percentage, 1 to 100. If this percentage is exceeded, a warning message occurs. This is mandatory for IOS devices and optional for IOS XR devices. Device type specific validations occur during service request creation.

m. **RD Format**—From the drop-down list, you have two choices. Choose **RD_AS** for the Route Distinguisher (RD) to be in autonomous system (AS) format, for example: 100:202. Otherwise, choose **RD_IPADDR** for the RD to be in RD_IPADDRESS format, for example: 10.2.2.3:1021.

n. **RD** (required)—Specify a Route Distinguisher (RD) manually or check the **Autopick RD** check box to have Prime Fulfillment automatically choose an RD from the Route Distinguisher pool, if one has been set up.

o. **Enable IPv4 Multicast**—Multicast VRF deployments are supported only for IPv4 deployments. Route Target is mandatory if multicast is enabled. Check the check box to enable IPv4 multicast VRF deployments.

p. **Enable IPv6 Multicast**—Multicast VRF deployments are supported only for IPv6 deployments. Route Target is mandatory if multicast is enabled. Check the check box to enable IPv6 multicast VRF deployments.

q. **Enable Auto Pick MDT Addresses** (optional)—Check this check box to use **Default MDT Address** and **Default MDT Subnet** values from a multicast resource pool.

r. **Default MDT Address**—If **Enable Auto Pick MDT Addresses** is not checked (set on), you can provide the **Default MDT Address**.

s. **Data MDT Subnet** (optional)—If **Enable Auto Pick MDT Addresses** is not checked (set on), you can provide the **Default MDT Subnet**.

t. **Data MDT Size** (optional)—If **Enable Multicast** is set on, **Data MDT Size** is required. From the drop-down list, select the data MDT size.

   MDT refers to a *multicast distribution tree* (MDT). The MDT defined here carries multicast traffic from providers associated with the multicast domain.

u. **Data MDT Threshold** (optional)—If **Enable Multicast** is set on, **Data MDT Threshold** is required. Enter the bandwidth threshold for the data multicast distribution tree. The valid range is 1-4294967 and indicates kilobits/second.

   The *data MDT* contains a range of multicast group addresses and a bandwidth threshold. Thus, whenever a PE behind a multicast-VRF exceeds that bandwidth threshold while sending multicast traffic, the PE sets up a new data MDT for the multicast traffic from that source. The PE informs the other PEs about this data MDT and, if they have receivers for the corresponding group, the other PEs join this data MDT.

v. **Default PIM Mode** (optional)—For Default Protocol Independent Multicast (PIM) mode, click the drop-down list and choose **SPARSE_MODE** or **SPARSE_DENSE_MODE**. For IOS XR devices, no configlet is generated for either mode.

w. **MDT MTU** (optional)**—**For this MDT Maximum Transmission Unit (MTU), the range for IOS devices is 576 to 18010, and the range for IOS XR devices is 1401 to 65535. Device type specific validations occur during service request creation.

x. **Enable PIM SSM** (optional)—Check this check box for PIM Source Specific Multicast (SSM).

y. **SSM List Name** (optional)—Choose **DEFAULT** from the drop-down list and you create the following CLI: **ip pim vrf** <*vrfName*> **ssm default**. No configlet is generated for IOS XR devices, because they are using the standard SSM range 232.0.0.0/8. Choose **RANGE** from the drop-down list to associate an access-list number or a named access-list with the SSM configuration. This creates the following CLI: **ip pim vrf** <*vrfName*> **ssm range {ACL#!named-ACL-name}**.

z. **Multicast Route Limit** (optional)—Enter a valid value of 1 to 2147483647. For IOS XR devices, no configlet is generated.

aa. **Enable Auto RP Listener** (optional)—Check this check box to enable the Rendezvous Point (RP) listener function. By default, this feature is running on IOS XR devices and no configlet is generated for this attribute.

ab. **My PIM Static-RPs**—To configure static RPs, check this check box. An edit option then goes active. Click **Edit** and fill in the applicable fields in the window that appears. Then click **OK**.

Step 4 When you are satisfied with the settings for this VRF, click **Save**.

You have successfully created a VRF, as shown in the **Status** display in the lower left corner of the VRFs window.

Editing VRFs

From the VRFs window, you can edit one or more VRFs.

To edit VRF(s), follow these steps:

Step 1 Choose **Inventory > Logical Inventory > VRF**.

Step 2 Check the check box(es) for all the VRFs you want to edit and then click **Edit**.

Step 3 If you check only one check box for one VRF, you receive a window with the title of the window as **Edit VRF**, the **Name** field has the name of the VRF you selected, and the **Provider** field already has the name of the provider for the VRF you selected. After you make your changes, you proceed to Step 8.

Step 4 If you check multiple check boxes, you receive a window with the title as Edit Multiple VRFs.

Step 5 In the **VRFs Affecting** section, the names of the VRFs you chose are given. If you click on **Attributes**, you receive a window with the currently configured attributes of all the selected VRFs.

Step 6 In the **Route Attributes** section, specify the **Import Targets** and **Export Targets** you want to **Add** and **Remove.** These lists of Route Targets (RTs) should be separated by commas, as indicated in **Import RT List** and **Export RT List** in the "Creating a VRF" section on page 5-255. See the "Creating a VRF" section on page 5-255 for information about the remaining fields you want to edit.

Step 7 In the **Multicast Attributes** section, you can edit the fields. See the "Creating a VRF" section on page 5-255 for information about the fields you want to edit.

Step 8 Click **Save** and the VRFs will be updated.

Deleting VRFs

From the VRFs window, you can delete specific VRF(s).

Note Only VRFs not associated with VRF service requests can be deleted.

To delete VRF(s), follow these steps:

Step 1 Choose **Inventory > Logical Inventory > VRF**.

Step 2 Select VRF(s) to delete by checking the check box(es) to the left of the VRF name(s).

Step 3 Click the **Delete** button.

The Confirm Delete window appears.

Step 4 Click **OK** to confirm that you want to delete the VRF(s) listed.

The VRFs window reappears with the specified VRF(s) deleted.

CHAPTER 6

Managing MPLS Transport Profile Services

This chapter describes the tasks required to get started using Cisco Prime Fulfillment 6.2, Multiprotocol Label Switching (MPLS) Transport Profile (TP) services.

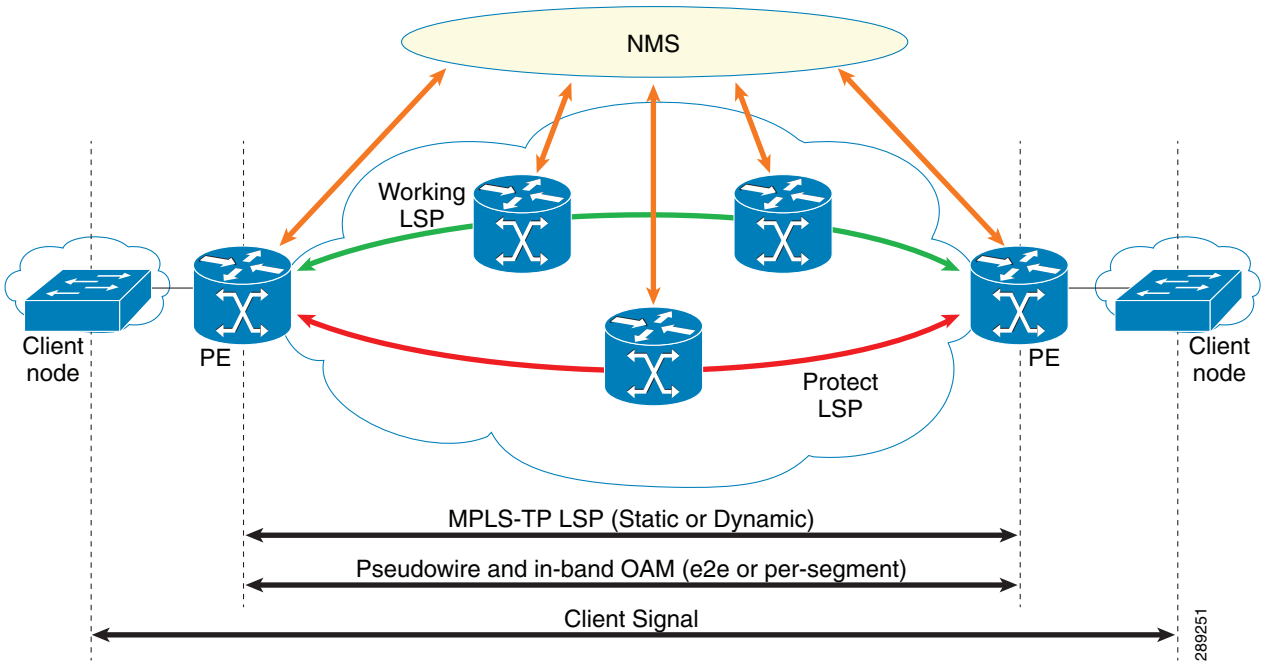
This section covers the following topics:

Introduction

MPLS-TP is a transport service (managed by Prime Fulfillment) for a dynamic MPLS core (not managed by Prime Fulfillment).

In the current implementation of MPLS-TP, an MPLS-TP tunnel can be provisioned between two arbitrary nodes in an MPLS-TP enabled network. The provisioned tunnel can have one or two paths, a working and an optional protect label-switched path (LSP). The normal use case is for Prime Fulfillment to automatically calculate the working and protect paths using a path selection algorithm that chooses MPLS-TP enabled links based on shortest path, and to provision the tunnel on the endpoints and all nodes traversed by the tunnel.

Figure 6-1 An MPLS-TP Enabled Network



Prerequisites and Limitations

The current release of Prime Fulfillment involves certain prerequisites and limitations, which are described in the *Cisco Prime Fulfillment Installation Guide 6.2*, including general system recommendations.

Note that Internet Explorer 8 (IE8) will not show the calculated path graphically (as described in Creating an MPLS-TP Service Request, page 6-8) as IE8 offers no support for SVG display. Until IE9 is supported, a textual summary of the path can be used to review the path in IE8.

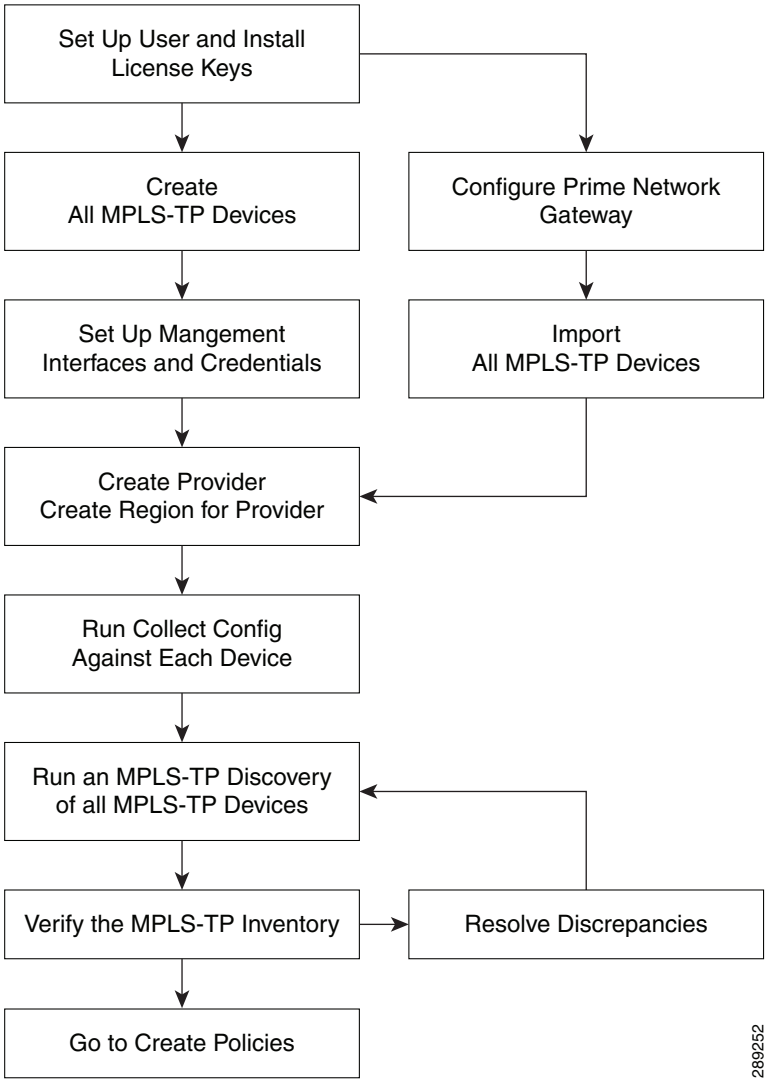
For supported device and OS information , refer to *Cisco Prime Fulfillment Supported Devices 6.2*.

Preconfiguration Process

The preconfiguration process sets up key parameters that enable the system to collect MPLS-TP network information and subsequently deploy MPLS-TP configurations on the chosen network.

The different steps in the preconfiguration process are provided in Figure 6-2.

Figure 6-2 Preconfiguration Process



Before commencing the preconfiguration process, MPLS-TP needs to be enabled on the network devices by making sure that the IP addresses used as devices' MPLS-TP IDs are accessible from the management station (this step is not supported by MPLS-TP). This is described in Other MPLS-TP Preconfiguration Requirements, page 6-4.

Setting up new user and installing license keys is described in Chapter 14, "Administration Tasks" and the other steps are covered in Setting Up Devices and Device Groups, page 2-1 and the Inventory - Discovery appendix (Collect Config step).

See below for a description of specific MPLS-TP user roles.

The MPLS-TP-specific steps are as follows:

1. **Run an MPLS-TP Discovery Task**—Use Task Manager to discover the MPLS-TP network for a particular MPLS-TP provider to populate the repository with a view to creating primary and backup tunnels. (See Running MPLS-TP Discovery, page 6-5.)

2. **Verify the MPLS-TP Inventory**—Verify that the MPLS-TP Discovery task was successfully completed. This can be done in a couple of ways. (See Verifying the MPLS-TP Discovery Results, page 6-6.)

MPLS-TP Setup and Installation

Before setting up Prime Fulfillment, the Prime Fulfillment software must be installed. To do so, see the *Cisco Prime Fulfillment Installation Guide 6.2*.

To set up a new Prime Fulfillment user, one or more users with a MPLS-TP role must be created. MPLS-TP roles are decribed in MPLS-TP User Roles, page 6-4 and step by step instructions for creating user roles are documented in Users, page 14-9.

Licensing information, including the Prime Fulfillment licensing options and the procedure needed to install licenses is described in Licensing, page 14-6.

MPLS-TP User Roles

Prime Fulfillment currently supports two MPLS-TP roles, the MPLS-TPRole and MPLS-TPServiceOpRole. These 2 user roles behave similarly to the other roles in Prime Fulfillment, for example the MPLSRole and the MPLSServiceOpRole found in MPLS.

They have the following permissions:

- MPLS-TPRole—full permission to manage the inventory (create, read, update, delete, and deploy MPLS-TP policies and service requests)
- MPLS-TPServiceOpRole—permission to deploy MPLS-TP service requests

For an explanation of how to work with roles, see User Roles, page 14-16.

Other MPLS-TP Preconfiguration Requirements

Prior to performing MPLS-TP provisioning, perform the following additional configuration steps:

Step 1 Enable MPLS-TP on the device:

- Choose a global ID common to all devices (AS number, for example)
- Allocate a device ID to each device.
- Configure MPLS-TP-related timers.

Step 2 Configure a range of statically defined MPLS labels to be used by MPLS-TP tunnels and static pseudowires.

Step 3 Enable MPLS-TP links to select which interfaces will form the links in the MPLS-TP topology:

- Give each interface an ID.
- Optionally configure a bandwidth pool on each interface.
- Enable CDP on the MPLS-TP enabled links. This is not needed for MPLS-TP but is required for MPLS-TP discovery to work in Prime Fulfillment.

Step 4 Create a BFD class to be used to monitor your MPLS-TP tunnels.

Running MPLS-TP Discovery

As a prerequisite for running MPLS-TP discovery, all devices must be present and a Collect Config task must be run (see the Inventory - Discovery appendix, Collect Config step).

The MPLS-TP network is discovered using the **MPLS-TP Discovery** task. This populates the repository with the network topology in an automated way. The necessary steps are described in this section.

Note MPLS-TP discovery will update only the functional MPLS-TP links in the MPLS-TP routing diagram (Service Request Editor, Review Routing accordion).

The MPLS-TP discovery process discovers the following from the live network:

- TP enabled links
- MPLS Static label pools
- MPLS Static label pool usage
- BFD templates
- TP Router ID
- TP Global ID

Where possible, the discovery process will try to keep the repository consistent with the network, for example delete links which have been removed. In cases where this is not possible, for example if a link is in use, a log message will be recorded.

This section includes the following:

Creating an MPLS-TP Discovery Task

To create a MPLS-TP Discovery task on the MPLS-TP network, use the following steps:

Step 1 Choose **Operate > Task Manager**.

The Task Manager window appears.

Step 2 Create a new task by selecting **Create > MPLS-TP Discovery**.

The Create Task window appears.

Step 3 Make any desired changes to the auto-generated name and description text and click Next.

The **MPLS-TP Discovery** window appears.

Step 4 Select the devices through which the MPLS-TP network should be discovered.

Step 5 Click Submit.

The discovery process begins.

Step 6 Once the MPLS-TP discovery task is complete, the outcome will be documented in a log under Operate > Task Logs.

Links and resource pools should now be visible in the MPLS-TP Details window, which is accessible from the Inventory > Devices > MPLS-TP Details page.

Verifying the MPLS-TP Discovery Results

After running MPLS-TP Discovery, you can see the result in various ways.

Viewing Logs

Once the **MPLS-TP Discovery** task is completed, you can view the log that is generated. This summary log will list any changes that have occured in the MPLS-TP network.

To view the log, select the relevant task in Task Manager and click **Logs**.

Verifying Links, Pools, and MPLS-TP Global and Router IDs

To verify the status of links and pools, go to the MPLS-TP Details page at Inventory > Devices > MPLS-TP Details.

The MPLS-TP global and router IDs for a particular device can be verified by going to Inventory > Devices > Edit.

Creating an MPLS-TP Policy

An MPLS-TP policy is needed to successfully create and deploy a service request. It serves as a template for the settings that are needed on the device.

To create an MPLS-TP policy, use the following steps:

Step 1 Choose one of the following:

a. **Service Design > Policy Manager**.

In the Policy Manager window, click **Create**.

b. **Service Design > Create Policy**.

In either case, a Policy Type drop-down appears.

Step 2 Click the down-arrow to open the **Policy Types** drop-down list and select **MPLS-TP Tunnel**.

The Policy Information accordion opens.

Step 3 Complete accordion 1 – Policy Information.

Enter **Policy Name** and optionally a **Description**. Policy Name is the only field that is mandatory in the Policy Editor.

Step 4 Click **Next.**

The Policy Information accordion closes and the next accordion opens.

Step 5 Complete accordion 2 – Tunnel Characteristics.

Set how each of the attributes will be displayed within the Service Request Editor window using the drop-down next to each field:

- **Editable** will display the attribute and permit modification.
- **Visible** will display the attribute but prevent editing.
- **Hidden** will not display the attribute.

Make sure to select **Editable** for any fields that you want to be able to edit in the Service Request Editor.

Use the **State** field to indicate whether the tunnel should be provisioned with the **shutdown** command or not.

For path protection, keep the **Protection** box selected to have Prime Fulfillment autogenerate an alternate protective path for the new tunnel.

For the **Diversity Options** drop-down menu, choose one of the following options:

- Node Diversity Required—Path calculation will fail if protection with unique nodes cannot be found.
- Node Diversity Desired—Allow a path with common nodes to be returned.
- Link Diversity Only—Do not allow working and protection path to pass through the same links.

Step 6 Complete accordion 3 – Tunnel End-points.

As in the previous accordion, remember to specify which fields should be Editable, Visible, and Hidden in the Service Request Editor.

Complete the fields as needed, using the drop-downs to select source and destination nodes and BFD templates.

Select the required BFD templates from a list of available BFD templates on the source and destination devices respectively. A valid BFD template name is max. 31 characters long.

For an explanation of global ID and router ID, see Global ID and Router ID, page 6-7.

Step 7 Click **Finish** to create the policy.

The new policy appears in the list of tunnels in the Policy Manager.

Global ID and Router ID

Global ID and router ID are used to identify devices within the MPLS-TP network so they can be discovered and managed.

If you as a user decide to specify the router ID and global ID, those values will be used for tunnel creation. If they are not specified, the router ID and global ID configured on the device itself are used.

Every MPLS-TP tunnel and LSP has a unique ID formed by the concatenation of the Global ID, Router ID, Tunnel ID, and LSP ID of both ends of the tunnel. This ID is configured at every endpoint and midpoint of the tunnel. The Global ID and router ID are normally configured globally on a router but it is possible to override these values for specific tunnels. Prime Fulfillment is aware of the globally configured IDs and uses them when configuring tunnels but also allows you to override these values as needed.

Global ID

Every MPLS-TP enabled node can have an MPLS-TP global ID configured within the global configuration. If the Global ID is set at the MPLE-TP global configuration level, it will be used as the default global ID for all endpoint and midpoint configuration. If not configured, a global ID of 0 is used for configured tunnels unless a different value is explicitly specified within the tunnel configuration itself.

The MPLS-TP global ID is retrieved from a device via MPLS-TP discovery.

Router ID

To be MPLS-TP emabled, a device must have a router ID.

If neither the MPLS-TP router ID nor the MPLS-TP global ID can be retrieved from the device, this is logged in the corresponding **MPLS-TP Discovery** task log file and all remaining MPLS-TP Discovery steps are halted for this device. The device in question is flagged as being MPLS-TP Disabled.

Creating an MPLS-TP Service Request

An MPLS-TP service request needs to be created to deploy a service request. It is assumed that at least one MPLS-TP policy is available. If not, see Creating an MPLS-TP Policy, page 6-6.

To create an MPLS-TP service request, use the following steps:

Step 1 This operation can be done in two ways:

a. From the Policy Manager, select the desired policy and click **Create Service Request**.

b. Choose **Operate > Create Service Request**.

The Service Request Editor window appears.

Next to the **Policy** field, click the down-arrow to open the **Policies** drop-down list.

Step 2 Select the desired MPLS-TP policy.

The Service Request Editor opens. In this editor,

Step 3 In the Service Request accordion, add a description in the **Service Description** field.

Step 4 In the Tunnel Characteristics accordion, use the pre-populated field values or make the desired modifications.

To set the **Diversity Options**, see Creating an MPLS-TP Policy, page 6-6 for an explanation.

Step 5 In the Tunnel End-Points accordion, complete the **Source Node** and **Destination Node** fields and optionally any other fields.

In this accordion, both source device, destination device, and BFD information is mandatory.

Step 6 In the Review Routing accordion, specify which links should be required or excluded.

Click **Calculate Path** to view the path diagram:

Working path—Green solid line

Protect Link—Red dotted line

For an example of an MPLS-TP routing diagram generated with **Calculate Path**, see Figure 6-3 .

Figure 6-3 MPLS-TP Routing Diagram

- **Working Path Summary**—Click this button to view hop and link information for the working path.
- **Protect Path Summary**—Click this button to view hop and link information for the protect path.
- Add (or remove) path constraints by clicking the plus (or minus) icons to the right:
  - **Required NE/Link**—Specify network elements or links that traffic must pass through for either the working or the protect path.
  - **Excluded NE/Link**—Specify network elements or links that traffic must **not** pass through for either the working or the protect path.

For more information about path constraints, see Working with Path Constraints, page 6-10.

Step 7 Go back over the various accordions to check and edit as necessary.

Step 8 Click **Finish** on the last accordion to complete the create service request operation.

The Service Request Manager window opens.

For information about the Service Request Manager elements and operations, see Chapter 8, "Managing Service Requests."

Guidelines for working with path constraints are provided in Working with Path Constraints, page 6-10.

An MPLS-TP service request that is in the **DRAFT** state can be modified. If a **DRAFT** MPLS-TP service request is modified, the new values will replace the previously saved values.

If the **DRAFT** service request is determined to be valid and complete so that all mandatory fields have been populated with valid values and the **Save** button is pressed, then the existing MPLS-TP service request will be moved to the **REQUESTED** state. The wizard is completed when you press the **Finish** button. Values to be auto-allocated from resource pools are allocated at that point.

If the **DRAFT** service request is determined to be incomplete so that not all mandatory fields have been populated or there are validation errors and the **Save** button is pressed, then the existing MPLS-TP service request will remain in the **DRAFT** state. The wizard is completed when you press the **Finish** button. Values to be auto-allocated from resource pools are NOT allocated at that point.

A service request in **DRAFT** state is marked by a white/orange work cone in the Service Request Manager.

[

Working with Path Constraints

Path constraints can be added to control the tunnel path when a service request is created or modified as shown in the procedure in Step 6 in the create procedure.

There are two ways to add path constraints:

- Clicking a node or link on the routing diagram and clicking the plus sign. This adds a new path constraint to the working path by default. Change to **Protect Path** using the drop down if needed. Similarly, clicking the minus sign will remove the constraint.
- If the node/link you want to exclude/include is not present in the diagram, you can use the selector next to **Required NE/Link**.

Note If you change anything after the first path calculation, for example adding/removing constraints, switching protection on/off, etc., you will need to re-run path calculation by clicking **Calculate Path**.

Running Config Audit

A config audit task can be run against an MPLS-TP service requests to check that the configuration rolled onto a device by a particular service request is still present as expected.

To create a MPLS-TP Config Audit task, use the following steps:

Step 1 Choose **Operate > Task Manager**.

Step 2 Click **Audit > Config Audit** to open the Create Task window.

Step 3 Modify the **Name** or **Description** fields as desired and click **Next**.

The service request selection window appears.

Step 4 Click **Select SRs** to add a service request and select schedule.

Step 5 Click **Submit**.

If successful, this adds the task to the list of created tasks in the Tasks window.

To view the task logs for the created tasks, in Task Manager select the created task and click **Logs**.

Running MPLS-TP Functional Audit

In an MPLS-TP Functional Audit, information is retrieved from source and destination endpoints to provide tunnel audit information.

This task only performs functional audit on service requests, which are not in one of the following states:

- **Draft**
- **Closed**
- **Requested**
- **Invalid**
- **Failed Deploy**

For more information on working with service requests, see Chapter 8, "Managing Service Requests."

To create a MPLS-TP Functional Audit task, use the following steps:

Step 1 Choose **Operate > Task Manager**.

Step 2 Click **Audit > MPLS-TP Tunnel Functional Audit** to open the Create Task window.

Step 3 Modify the **Name** or **Description** fields as desired and click **Next**.

The service request selection window appears.

Step 4 Click **Select SRs** to add a service request and select schedule.

Step 5 Click **Submit**.

If successful, this adds the task to the list of created tasks in the Tasks window.

To view the task logs for the created tasks, in Task Manager select the created task and click **Logs**.

Deploying an MPLS-TP Tunnel

The final step required to provision an MPLS-TP service request is the deploy the service request. This pushes the service request and the associated configuration updates to the network.

Note A service request in **DRAFT** state cannot be deployed.

The deploy functionality is the same as for other Prime Fulfillment services. For instructions on how to deploy an MPLS-TP service request, see Deploying Service Requests, page 8-10.

Decommissioning

MPLS-TP service request configurations can be removed from the network using the decommissioning functionality within the Service Request Manager. Decommissioning will cause the previously deployed configurations to be removed from all tunnel endpoint and mid-point devices within the MPLS-TP tunnel path.

To decommission one or more service requests, see Chapter 8, "Managing Service Requests."

Sample Configlets

The configlets included in this section show the CLIs generated by Prime Fulfillment for particular services and features. Each configlet example provides the following information:

- Service
- Feature
- Devices configuration (network role, hardware platform, relationship of the devices and other relevant information)
- Sample configlets for each device in the configuration
- Comments.

All examples in this section assume the presence of an MPLS-TP core.

Note The configlets generated by Prime Fulfillment are only the delta between what needs to be provisioned and what currently exists on the device. This means that if a relevant CLI is already on the device, it does not show up in the associated configlet.

This section provides sample configlets for MPLS-TP service provisioning in Cisco Prime Fulfillment.

It includes the following section:

- MPLS-TP Working Tunnel Configlet (IOS), page 6-13

MPLS-TP Working Tunnel Configlet (IOS)

Configuration

- Service: MPLS-TP Working Tunnel
- Feature: MPLS-TP configlet (IOS) for configuring MPLS-TP enabled nodes

Configlets

IOS Device Configuration	Comments

Endpoint Config
`interface Tunnel-tp200`
`  description PrimeF:JobID:2(testTunnel)`
`  tp tunnel-name test`
`  tp bandwidth 100`
`  tp source 3.3.3.3 global-id 2`
`  tp destination 1.1.1.1 tunnel-tp 200 global-id 3`
`  bfd BFDTemplate-SingleHopMicrosec-1`
`  working-lsp`
`  lsp-number 0`
`  in-label 8018`
`  out-label 5003 out-link 8`
`  protect-lsp`
`  lsp-number 1`
`  in-label 8019`
`  out-label 50012 out-link 12`
 	
Midpoint Config
`mpls tp lsp source 3.3.3.3 global-id 2 tunnel-tp 200 lsp working destination 1.1.1.1 global-id 3 tunnel-tp 200`
`  forward-lsp`
`  tp bandwidth 100`
`  in-label 5003 out-label 50011 out-link 10`
`  reverse-lsp`
`  tp bandwidth 100`
`  in-label 5004 out-label 8018 out-link 8`
 	
EndPoint Config
`interface Tunnel-tp200`
`  description PrimeF:JobID:2(testTunnel)`
`  tp tunnel-name test`
`  tp bandwidth 100`
`  tp source 1.1.1.1 global-id 3`
`  tp destination 3.3.3.3 tunnel-tp 200 global-id 2`
`  bfd BFDTemplate-SingleHopMicrosec-1`
`  working-lsp`
`  lsp-number 0`
`  in-label 50011`
`  out-label 5004 out-link 10`
`  protect-lsp`
`  lsp-number 1`
`  in-label 50012`
`  out-label 8019 out-link 12`
	`Create an MPLS-TP working tunnel with endpoint and midpoint nodes. This involves configuring the settings on each node in the tunnel.`
	
	`Create an MPLS-TP working tunnel with the following attributes:`
	
	`Endpoint 1:`
	`- tp tunnel name: test`
	`- Source: 3.3.3.3`
	`- Destination 1.1.1.1`
	`- Bandwidth 100 kbps`
	`- bfd BFDTemplate-SingleHopMicrosec-1`
	`- Working LSP configuration`
	`- Protect LSP configuration`
	
	`Midpoint:`
	`- Source: 3.3.3.3`
	`- Destination 1.1.1.1`
	`- Bandwidth 100 kbps`
	`- Forward LSP configuration`
	`- Reverse LSP configuration`
	
	`Endpoint 2:`
	`- tp tunnel name: test`
	`- Source: 1.1.1.1`
	`- Destination 3.3.3.3`
	`- Bandwidth 100 kbps`
	`- bfd BFDTemplate-SingleHopMicrosec-1`
	`- Working LSP configuration`
	`- Protect LSP configuration`

CHAPTER 7

Managing MPLS Traffic Engineering Services

This chapter contains a detailed description of the Cisco Prime Fulfillment Traffic Engineering Management (TEM) product, including the various features, the GUI, and the step-by-step processes needed to perform various traffic engineering management tasks.

To get an overview of TEM and an introduction to some of the terminology used, see *Cisco Prime Fulfillment Theory of Operations Guide 6.2*.

This chapter includes the following sections:

Getting Started

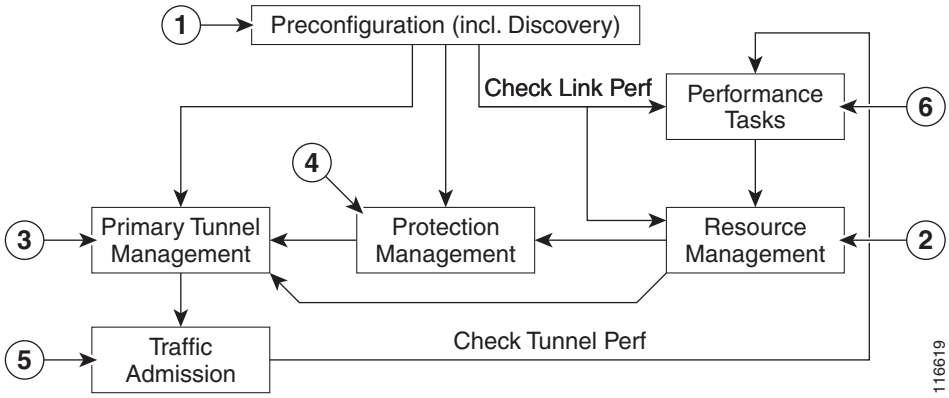
This section describes the installation procedure for Prime Fulfillment. The general installation procedure for Cisco Prime Fulfillment (Prime Fulfillment) is described in the *Cisco Prime Fulfillment Installation Guide 6.2*.

It includes the following sections:

Process Overview

The main components and flows in TEM are shown in Figure 7-1.

Figure 7-1 Main Process Flows in TEM

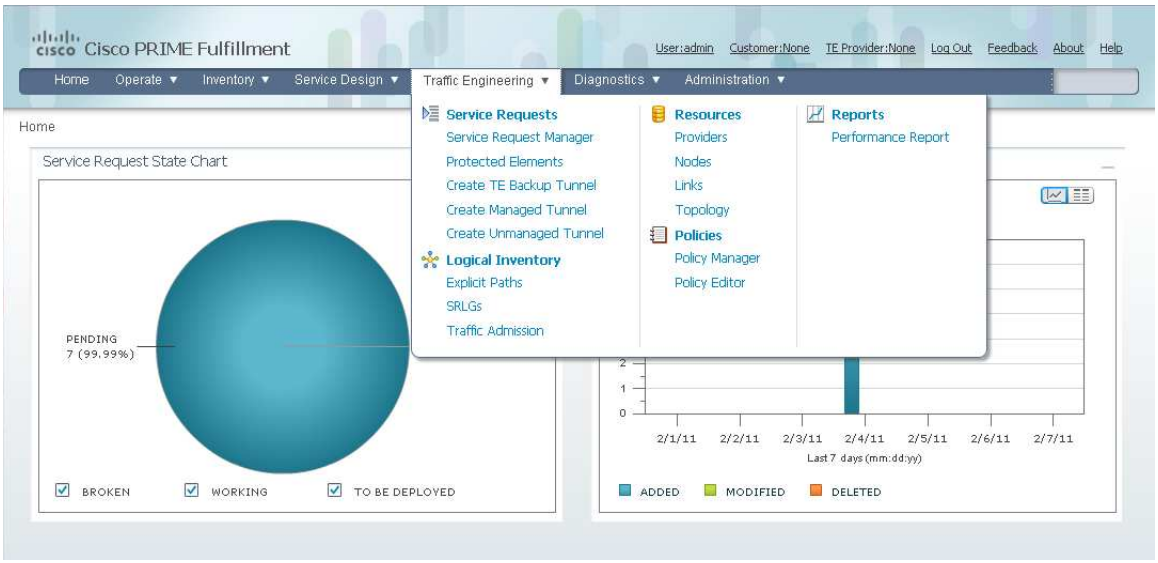


The illustration includes the following components:

1. Preconfiguration—Sets up key parameters that enable the system to collect TE network information (TE Discovery) and subsequently deploy TE configurations on the chosen network. (See Getting Started, page 7-1)
2. Resource Management—Tuning of certain properties on the TE interfaces to optimize the tunnel placement. (See TE Resource Management, page 7-20)
3. Primary Tunnel Management—Create and manage primary tunnels, either unmanaged (see Basic Tunnel Management, page 7-27) or managed. (see Basic Tunnel Management, page 7-27 or Advanced Primary Tunnel Management, page 7-44)
4. Protection Management—Protect selected elements in the network (links, routers, or SRLGs) against failure. (See Advanced Primary Tunnel Management, page 7-44)
5. Traffic admission—Assign traffic to traffic-engineered tunnels. (See TE Traffic Admission, page 7-66)
6. Performance Tasks—Calculates interface/tunnel bandwidth utilization using the Simple Network Management Protocol (SNMP). (See Administration, page 7-70)

The Traffic Engineering menu options in the Prime Fulfillment user interface are shown in Figure 7-2.

Figure 7-2 Traffic Engineering Menu Options



Prerequisites and Limitations

The current release of Prime Fulfillment involves certain prerequisites and limitations, which are described in this section.

See the *Cisco Prime Fulfillment Installation Guide 6.2* for general system recommendations.

General Limitations

The present release of Prime Fulfillment has the following limitations:

- Although concurrent use of Prime Fulfillment is supported in the Planning portion of the current implementation (see the section Multiple Concurrent Users in the *Cisco Prime Fulfillment Theory of Operations Guide 6.2*), multiple browsers on the same machine are still not recommended due to a limitation in Browser Session Attributes.
- JRE version 1.6.0_07 or higher should be installed on the client computer for launching Java applications and Applets. This can be done via Java's Control Panel. If you do not already have Java installed, you can use the links on the Topology Tool page to install the version that is bundled with Prime Fulfillment.
- If your repository predates the ISC 4.1 release and has been upgraded to a 4.1 or later repository, you need to run a TE Discovery task to collect software version information from the devices before deploying service requests.
- Let issued service requests finish deployment before issuing other service requests to avoid conflicts. This is described in more detail in the tunnel provisioning sections.

Feature-Specific Prerequisites and Limitations

Prime Fulfillment has the following feature-specific prerequisites and limitations:

- Some features might only be available with a particular license. In addition, the number of nodes provided by the license limits the size of the network. For more information, see Licensing, page 14-6.
- A number of specific requirements are associated with the TE Discovery task. These are described in TE Discovery Prerequisites and Limitations, page 7-12.
- Prime Fulfillment manages a single OSPF area or IS-IS level. Prime Fulfillment also supports multiple OSPF areas, however it does not discover tunnels between areas. Each OSPF area is mapped to a TE provider and is discovered area by area independently.
- Prime Fulfillment only supports MPLS-TE topology with point-to-point links.

Non-Cisco Devices and TEM

Prime Fulfillment does not manage non-Cisco devices and Prime Fulfillment cannot be used to provision them.

Prime Fulfillment will, however, discover non-Cisco devices and store them in the repository. Tunnels can be run through these devices, the bandwidth consumed can be accounted for, but the devices are not otherwise managed by Prime Fulfillment. TE tunnels originating from non-Cisco devices will not be discovered.

Sorting can be performed on different attributes in various parts of the Prime Fulfillment GUI. However, due to the added support for non-Cisco devices, sorting cannot be performed on Device Name and MPLS TE ID in the TE Nodes List window.

Supported Platforms

For supported devices and IOS platforms, see the *Cisco Prime Fulfillment Installation Guide 6.2*.

Error Messages

Warnings and violations that are invoked when using the TE planning tools in Prime Fulfillment are documented in Warnings and Violations, page 7-98

Elixir warning messages might appear when performing deployments in Prime Fulfillment:

```
WARNING Elixir.ServiceBlade Unable to load support matrix for the platform or platform
family. The default support matrix is loaded instead for role: TunnelHead.
WARNING Elixir.ConfigManager Attribute - lockdown of Command - Tunnel_PathOption can NOT
be retrieved from the input SR - SKIPPING.
```

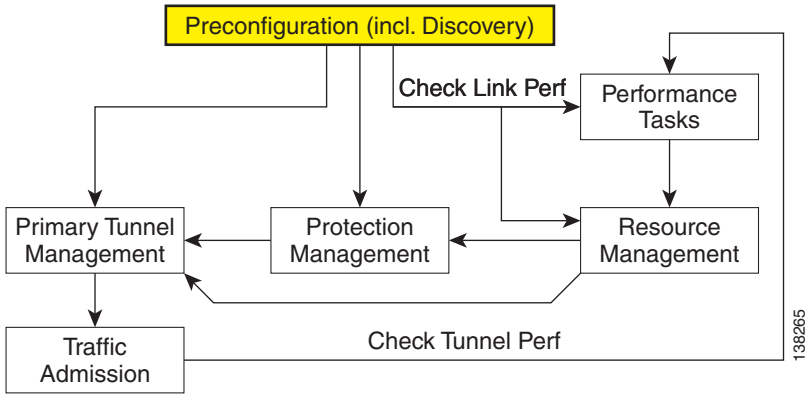
The deployments will, however, be successful and these messages can be safely ignored.

Preconfiguration Process Overview

The preconfiguration process sets up key parameters that enable the system to collect TE network information and subsequently deploy TE configurations on the chosen network.

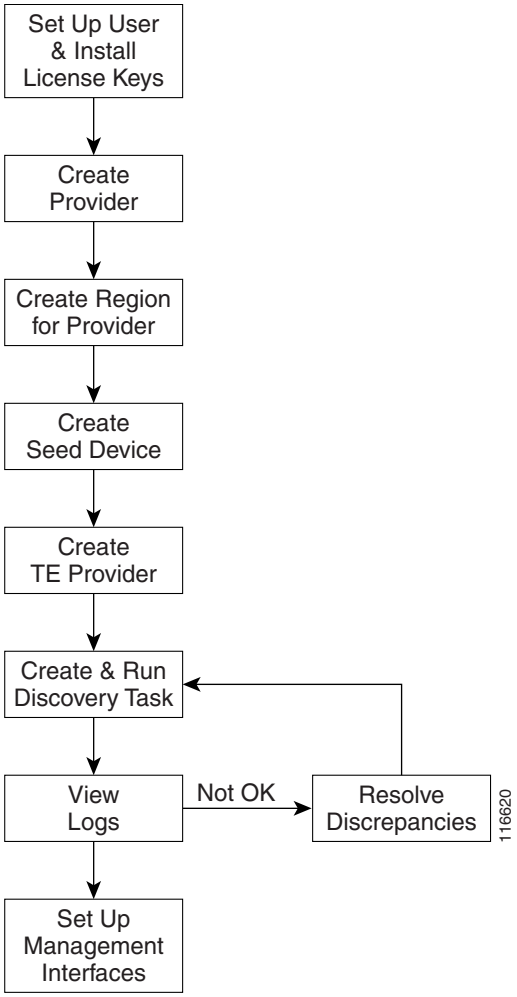
The highlighted box in Figure 7-3 shows where in Prime Fulfillment the preconfiguration steps take place.

Figure 7-3 Prime Fulfillment Process Diagram - Preconfiguration



The different steps in the preconfiguration process are provided in Figure 7-4.

Figure 7-4 Preconfiguration Process



Before commencing the preconfiguration process, MPLS-TE needs to be enabled on the network devices by making sure that the IP addresses used as devices' TE IDs are accessible from the management station (this step is not supported by TEM).

The preconfiguration process includes the following steps:

1. **Set up new user and install license keys**—To run the TEM blade of Prime Fulfillment, it is necessary to create a new user and install license keys. These keys allows you to view and manage the TE tunnels and resources using Prime Fulfillment. (See TEM Setup and Installation, page 7-6)
2. **Create a provider**—The provider is a concept designed to allow many different operators to work on Prime Fulfillment simultaneously, each working on different networks. Thus, each provider has to be defined and used as a reference operator for future work on the system. (To create a provider, see Providers, page 2-15.)
3. **Create a region for the provider**—The region is important because a single provider could have multiple networks. The region is used as a further level of differentiation to allow for such circumstances. (To create a region, see Provider Regions, page 2-16.)
4. **Create a seed device**—This IOS or IOS XR device will be the seed router for TE Discovery. The network discovery process uses the seed router as an initial communication point to discover the MPLS TE network topology. (To create a seed router, see Devices, page 2-1.)
5. **Create a TE Provider**—Providers can be defined as TE provider, if they are supporting MPLS TE in their network. To enable a TE network to be managed, it is necessary to create a TE provider. All TE related data associated with a given network is stored under a unique TE provider. A provider and region uniquely define a TE provider (See Creating a TE Provider, page 7-7.)
6. **Run a TE Discovery Task**—Discover the TE network for a particular TE provider to populate the repository with a view to creating primary and backup tunnels. (See TE Network Discovery, page 7-10.)
7. **Set Up Management Interfaces**—Set up management interfaces for discovered devices or update server host file with resolution for all discovered devices. This step is only necessary if the devices in the TE network are not accessible via their hostnames (See Setting Up Management Interfaces, page 7-19.)

Note If Telnet is selected to communicate with the seed router, Telnet must also be used for the other network devices. Likewise, if SSH is selected for the seed router, SSH must be used for all other devices.

TEM Setup and Installation

Before setting up Prime Fulfillment, the Prime Fulfillment software must be installed. To do so, see the *Cisco Prime Fulfillment Installation Guide 6.2*.

To set up a new Prime Fulfillment user, one or more users with a TE role must be created. For step by step instructions, see Users, page 14-9.

Licensing information, including the Prime Fulfillment licensing options and the procedure needed to install licenses is described in Licensing, page 14-6.

Editing DCPL Properties (Optional)

The Prime Fulfillment Dynamic Component Properties Library (DCPL) includes a wide variety of properties that are accessible from the GUI, some of which can be modified.

The various DCPL properties, including those pertaining to Prime Fulfillment, and the process for editing these properties are described in Manage Control Center, page 14-2.

Warning **Do not attempt to modify the DCPL properties unless you fully understand the implications.**

In the Prime Fulfillment GUI, the DCPL properties are found in **Administration > Hosts**. Check a check box for a specific host and click the **Config** button.

The DCPL properties pertaining to TEM are found in the following folders:

- **Provisioning > Service > TE**
- **TE**
- **TE Topology**

Creating a TE Provider

Before TE Discovery or any manipulation of TE data can take place, at least one TE provider has to be created. For example, an OSPF area can be assigned as a TE provider. Prior to this, a provider and a region for that provider must have been set up (see Preconfiguration Process Overview, page 7-4).

One region can be assigned as the default region as a place for discovered routers. These routers can then subsequently be placed in any region. For more information, see the section multiple hosts in the *Cisco Prime Fulfillment Theory of Operations Guide 6.2*.

To create a TE provider, use the following steps:

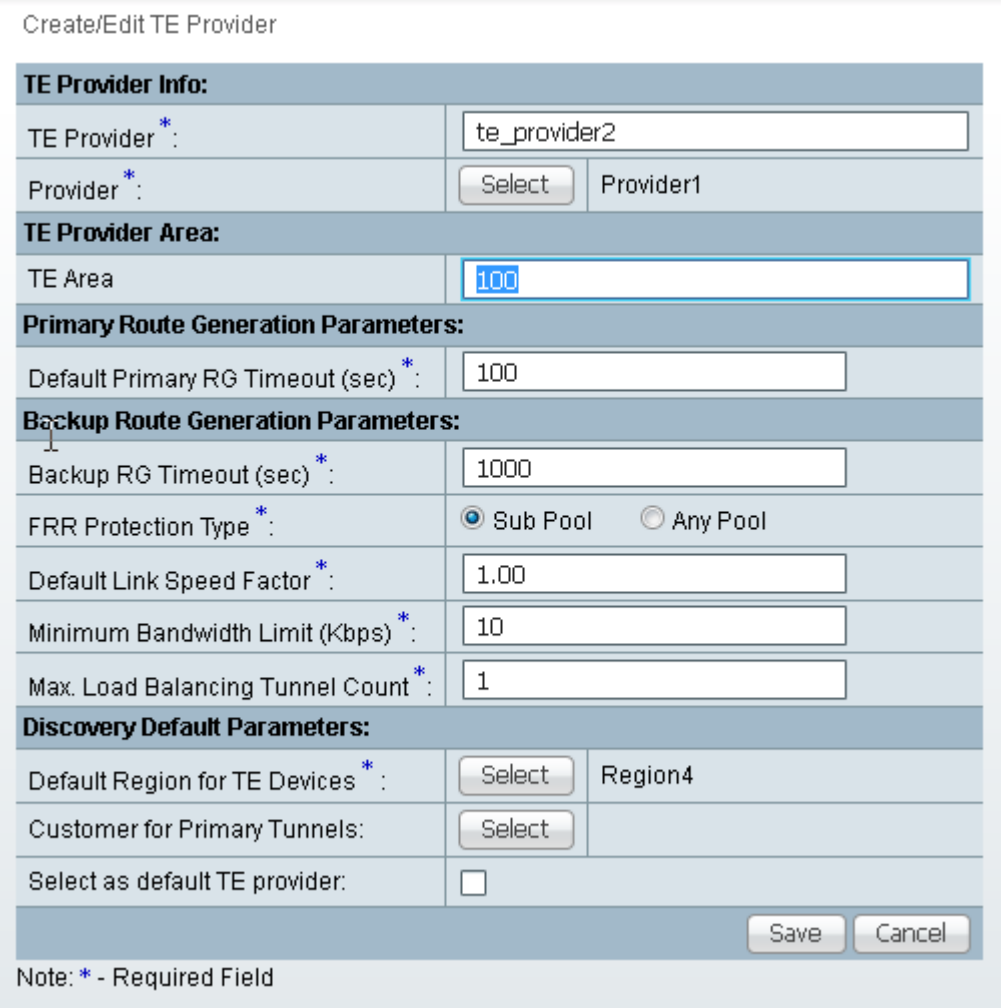
Step 1 Choose **Traffic Engineering > Providers**.

The TE Providers window appears.

Step 2 Click **Create** to create a TE provider.

The Create/Edit TE Provider window in Figure 7-5 appears.

Figure 7-5 Create/Edit TE Provider



The Create/Edit TE Provider window includes the following fields:

- **TE Area—**OSPF area assigned to the TE provider. This can be any positive integer from 0 to 4294967295 or a dot notation address of the form x.x.x.x where x is a number between 0 and 255.
- **Default Primary RG Timeout**—Default computation timeout for primary tunnels.
- **Backup RG Timeout**—Computation timeout per element for backup tunnels (for each protected element, the timer is reset to zero before the Prime Fulfillment attempts to protect it).
- **FRR Protection Type**—Fast Re-Route (FRR) protection type:
  - **Sub Pool**—Protect only sub pool primary tunnels.
  - **Any Pool**—Protect both sub pool and global pool primary tunnels.

  For a definition of pool types, see the section on bandwidth pools in the *Cisco Prime Fulfillment Theory of Operations Guide 6.2*.

- **Default Link Speed Factor**—Default multiplication factor to be applied to the link speed in order to determine move affected tunnels. that needs to be protected. The link's bandwidth is multiplied by the link speed factor, then the RSVP bandwidth reserved for the link (sub pool or global pool depending on the FRR protection type) is subtracted, and the resulting bandwidth is then available to FRR backup tunnels.

  Interpretation of the link speed factor:

  > 1.0 (overbooking)—more backup bandwidth than the link has available.

  < 1.0 (underbooking)—less backup bandwidth than the link has available.
- **Minimum Bandwidth Limit**—Minimum bandwidth allowed for backup tunnels.
- **Max. Load Balancing Tunnel Count**—This is the maximum number of backup tunnels needed to protect a flow through a protected element. Here, a flow is defined as follows:

  There are two flows in a protected link, one in each of the directions that traffic can flow. For a node, the number of flows depends on the number of neighbouring nodes for a particular node. There is a flow for each neighbour pair. So a node with 3 neighbours, A, B, and C, has 6 flows through it – A->B, A->C, B->A,B->C, C->A, C->B.
- **Default Region for TE Devices**—The default provider region is the one assigned by TE Discovery to a newly discovered device. If the device already exists in the repository and has a region defined, TE Discovery keeps that setting. It is possible to change the region of a device after TE Discovery.
- **Customer for Primary Tunnels**—Name of customer for primary TE tunnels.

Step 3 In the **TE Provider** field, enter a name for the new TE provider.

Step 4 To select a provider to be this TE provider, click the **Select** button next to the **Provider** field.

The Select Provider window appears.

Step 5 Select the desired provider using the radio buttons or search for a provider with search criteria matching a provider name and click **Find**.

Step 6 Click **Select** to select the desired provider.

The Select Provider window closes. The selected provider name is displayed in the **Provider** field.

Step 7 In the **TE Area** field, specify the number of the OSPF area to act as TE area.

Both dot notation and decimal notation are supported for the area identifier.

Note The **TE Area** field can be left blank if the seed router used for TE Discovery is not an Area Border Router, and it will be automatically populated on discovery.

Depending on the seed router used for TE Discovery, the area identifier should be set as follows:

- **Seed router is an ABR:** The area identifier field in TE provider must be set to indicate which of the two or more areas on the ABR is to be discovered.
- **Seed router is NOT an ABR:** Leave blank.

Note If you do not set the Area Identifier in TE Provider, TE Discovery will set it. After it is set, it cannot be changed.

Step 8 Add primary and backup route generation parameters.

When the FRR (Fast Re-Route) protection type is equal to Sub Pool, the backup tunnels generated by the tool will protect only the sub pool primary tunnels. When it is equal to Any Pool, the backup tunnels generated by the tool will protect both sub pool and global pool primary tunnels.

For more information on Fast Re-Route (FRR) protection pools, see the section on bandwidth pools in the *Cisco Prime Fulfillment Theory of Operations Guide 6.2*.

Step 9 Fill in the remaining required fields (marked '*') and any optional fields as desired.

Step 10 For the required **Default Region for TE Devices** field, click the corresponding **Select** button.

The Region for Create TE Provider window appears.

Step 11 Select the desired region using the radio buttons.

Step 12 Click **Select** to select the desired default region.

The Region for Create TE Provider window closes. The selected region name is displayed in the **Default Region for TE Devices** field.

Step 13 For the optional **Customer for Primary Tunnels** field, click the corresponding **Select** button.

The Customer for Create TE Provider window appears.

Step 14 If desired, select a customer using the radio buttons or search for a customer by entering customer search criteria in the **Show Customers with Customer Name matching** field and click **Find**.

Step 15 Click **Select** to select the desired customer.

The Select Customer for Create TE Provider window closes. The selected customer name is displayed in the **Customer for Primary Tunnels** field of the Create/Edit TE Provider window.

Step 16 Click **Save**.

The created TE provider appears in the TE Provider window and can now be used to perform TE discovery and other TE functions.

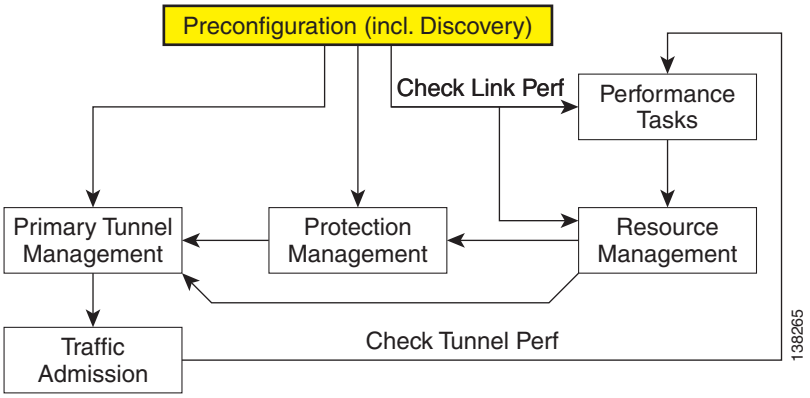
To switch between TE providers, go to the to the top of the Prime Fulfillment window above the menu toolbar (Figure 7-2) and click the **TE Provider** link.

TE Network Discovery

After completing the preconfiguration process and creating a seed router, you can discover the TE network for a particular TE provider. This populates the repository with the network topology. Also, you might need to set up the management interfaces. The necessary steps are described in this section.

The highlighted box in Figure 7-3 shows where in Prime Fulfillment the preconfiguration steps takes place.

Figure 7-6 Prime Fulfillment Process Diagram - Preconfiguration



The purpose of the TE discovery process is to populate the repository with the TE topology, TE tunnels, explicit paths, and static routes to tunnels present in the live network.

The TE discovery process uses a seed device to discover the MPLS TE network topology using either Telnet or SSH. All the Traffic Engineering routers in the network should be accessible via their TE ID.

TE Discovery is a schedulable task that can be run once or on a periodic basis. Any inconsistencies between the repository and the network are reported in the Discovery log. The service state information is updated incrementally by logging tunnel in-use Label Switched Paths (LSPs) and updating the service request (SR) state.

This section includes the following:

TE Discovery Prerequisites and Limitations

The following prequisites apply mainly to TE discovery.

For an overview of the general Prime Fulfillment prerequisites and limitations, see Prerequisites and Limitations, page 7-3.

Accessing TE Routers for TE Discovery

To successfully run a TE discovery task, the seed router must be directly accessible from the management station.

All TE routers must be accessible from the Prime Fulfillment machine via their TE router ID. This is often the loopback IP address, but not always.

For Telnet/SSH, there must be direct Telnet/SSH access from the Cisco Prime Fulfillment Traffic Engineering Management (TEM) management station to each device.

See Preconfiguration Process Overview, page 7-4 for instructions on how to select Telnet or SSH when setting up a seed router.

Note After performing a TE discovery, it is recommended that you do not manually reconfigure RSVP graceful restart on the device. This affects the synchronization with the database and can cause deployment failure, in which case a new TE discovery needs to be performed.

Memory Shortage on Large Networks

When running TE Discovery on a large network (250+ devices or 5000+ tunnels, for example) or an OutOfMemoryException is encountered, it is recommended that the memory setting be changed.

To do this, use the following steps:

Step 1 Choose **Administration > Hosts**.

Step 2 Select a host and click the **Config** button.

Step 3 Select **watchdog > server > worker > java > flags**.

Step 4 Change the first part of the property string, for example to **-Xmx1024m** instead of the default value **-Xmx512m**.

This increases the heap size of the **TE Discovery** task, which will clear up the OutOfMemoryException problem.

Step 5 Revert the **watchdog.server.worker.java.flags** property back to its original value to reduce the resource usage when no longer needed.

Note Alternatively, the same memory increase can be achieved by editing the **watchdog.server.worker.java.flags** property in the **vpnsc.properties** file.

IOS XR and Enable Passwords

If an IOS XR device is to be used as a seed device, the enable password should be set in its device record even though IOS XR does not require an enable password, for itself. That way IOS devices in the network, which do require an enable password, can be fully discovered.

When creating an IOS XR device through the **Devices** tab (**Inventory > Devices**) to act as a seed device for an initial discovery, it is not necessary to specify the enable password - TEM will be able to log in and get all the data it needs.

However, if there are other IOS devices in the same network, TEM will not be able to enter enable mode for those devices. As a result, these are not fully discovered in the sense that the inability to enter enable mode stops TEM from gathering all the relevant data. These other IOS routers will show up as '**unknown**' devices in the **Devices** window.

Limitations

Simultaneous TE Discovery in the same TE Provider is not supported. Only one user can run a TE Discovery per TE Provider at a time.

Creating a TE Discovery Task

In the Task Manager, you can run two types of TE Discovery tasks:

- TE Incremental Discovery, page 7-13
- TE Full Discovery, page 7-14

TE Incremental Discovery

This rediscovery process can take a long time to complete for a larger OSPF area.

In TE Incremental Discovery, the discovery tasks are run in increments whenever changes occur in the network, such as when a new device or link is added, causing a much smaller memory overhead than a TE Full Discovery.

To create a TE Discovery task on the TE network, use the following steps:

Step 1 Choose **Operate > Task Manager**.

The Task Manager window appears.

Step 2 Choose **Create > TE Incremental Discovery**.

The Task Creation wizard appears.

Step 3 Optionally, alter the **Name** and/or **Description** fields and click **Next**.

The TE Provider window appears.

Step 4 Select a TE provider and click **Next**.

The Device/Link Discovery Information window appears.

You can perform either of the following:

- Device discovery—A new device added to the network can be discovered using Device Discovery. For device discovery, non-Cisco devices, if any, are excluded from the list.

A device can be selected by clicking the Select button which shows the list of devices added in Inventory.

The prerequisite here is that the device which needs to be discovered needs to be added with its management IP address. The credentials of the device need not be the same as the credentials of other devices already populated in the repository. The device is successfully discovered only if it falls under the same OSPF area that is mentioned for the TE provider.

- Link discovery—A new link added to the network can be discovered using Link Discovery. Any explicit paths, primary, and backup tunnels traversing through that link will also be discovered.

  End Device A and End Device B can be selected from the list of devices which have already been (TE Nodes). You must specify Interface A and Interface B.

Step 5 Select the seed device for discovering the network and click **Next**.

The Task Schedules window appears.

Step 6 Create a task schedule in one of two ways:

- Click **Now** to schedule the task to run immediately, in which case the schedule information is automatically filled into the Task Schedules list.
- Click **Create** to create a scheduler for this task, in which case the Task Schedule window appears.

Step 7 In the Task Schedule window, make your selections to define when and how often the task should be run.

Note The default setting is to schedule a single **TE Discovery** task to take place immediately ("**Now**").

Step 8 Click **OK**.

The scheduled task should now appear in the Task Schedules table.

Step 9 Click **Next**.

A summary of the scheduled task appears.

Step 10 Click **Finish**.

This will add the task to the list of created tasks in the Tasks window.

TE Full Discovery

In a TE Full Discovery, the discovery task runs without stopping until all devices have been discovered.

To create a TE Discovery task on the TE network, use the following steps:

Step 1 Choose **Operate > Task Manager**.

The Task Manager window appears.

Step 2 Create a new task by selecting **Create > TE Full Discovery**.

The Create Task window appears.

Step 3 Optionally, alter the **Name** and/or **Description** fields and click **Next**.

The Select TE Provider window appears.

Step 4 Select a TE provider and click **Next**.

The Select Seed Device window appears. Non-Cisco devices, if any, are excluded from the list.

Step 5 Select the seed device for discovering the network and click **Next**.

The Task Schedules window appears.

Step 6 Create a task schedule in one of two ways:

- Click **Now** to schedule the task to run immediately, in which case the schedule information is automatically filled into the Task Schedules list.
- Click **Create** to create a scheduler for this task, in which case the Task Schedule window appears.

Step 7 In the Task Schedule window, make your selections to define when and how often the task should be run.

Note The default setting is to schedule a single **TE Discovery** task to take place immediately ("**Now**").

Step 8 Click **OK**.

The scheduled task should now appear in the Task Schedules table.

Step 9 Click **Next**.

A summary of the scheduled task appears.

Step 10 Click **Finish**.

This will add the task to the list of created tasks in the Tasks window.

Managing Per Area Discovery

Before running a per area TE discovery, it is helpful to understand how multiple OSPF areas are managed by Prime Fulfillment.

For background information on this topic, see the section Multiple OSPF Areas in the *Cisco Prime Fulfillment Theory of Operations Guide 6.2*.

This section describes the following:

- Performing a Per Area TE Discovery, page 7-15
- Running a Per Area TE Discovery Through an ABR, page 7-16.

Performing a Per Area TE Discovery

When a TE Discovery is run against an area with a selected TE provider, all tunnels and explicit paths associated with that area will be imported into the Prime Fulfillment database.

To initiate a per area TE discovery, use the following steps:

Step 1 Create an Provider.

Step 2 Create an Region.

Step 3 Create a TE Provider.

Step 4 Create a seed device from the Devices window.

Step 5 Choose **Operate > Task Manager > Create > TE Full Discovery**.

Specify a name for the TE Discovery task or accept the default and click **Next**.

Step 6 Select a TE Provider and click **Next**.

Step 7 Select a seed device and click **Next**.

Step 8 Select a schedule for the TE Discovery and click **Next**.

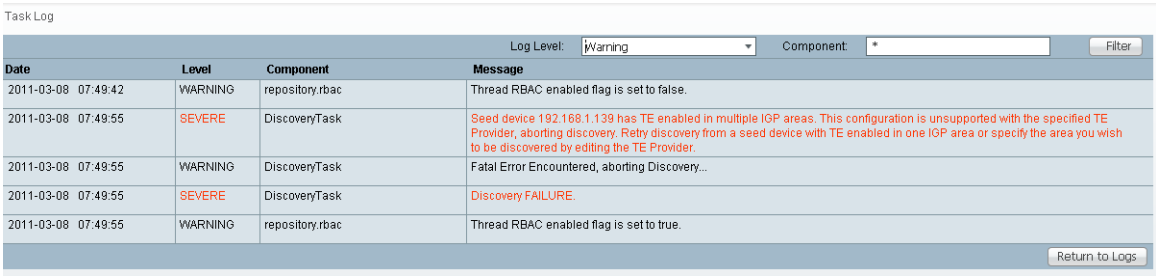
Step 9 Review the summary of the discovery task.

If it is acceptable, click **Finish** to start the TE Discovery process.

Running a Per Area TE Discovery Through an ABR

If no area identifier is specified in the TE provider configuration and the seed device is an ABR, TE Discovery will abort with the warning message shown in Figure 7-7 informing you to either specify an area identifier for the TE provider or use a non-ABR device as the seed.

Figure 7-7 TE Discovery Through an ABR with no TE Area Identifier Specified



Verifying a TE Discovery Task

The result of running the **TE Discovery** task can be assessed in four ways:

- Task Logs—View a summary log of any changes that have occured in the network.
- TE Topology—Display the latest TE Topology from the repository.
- View Network Element Types—In the Traffic Engineering Management GUI, go to **TE Nodes**, **TE Links**, **TE Primary Tunnels**, and so on to verify the state of specific network element types.
- Viewing the state of discovered devices—Go to the Service Requests window to examine whether the state of the discovered devices is as expected.

Task Logs

The TE Discovery log captures the state of the network and compares it with the most recent snapshot of the repository.

To view the task log for a **TE Discovery** task, use the following steps:

Step 1 Choose **Operate > Task Logs**.

The Task Logs window appears.

The status of the task is shown in the **Status** column. This updates automatically and indicates when the TE Discovery process is complete.

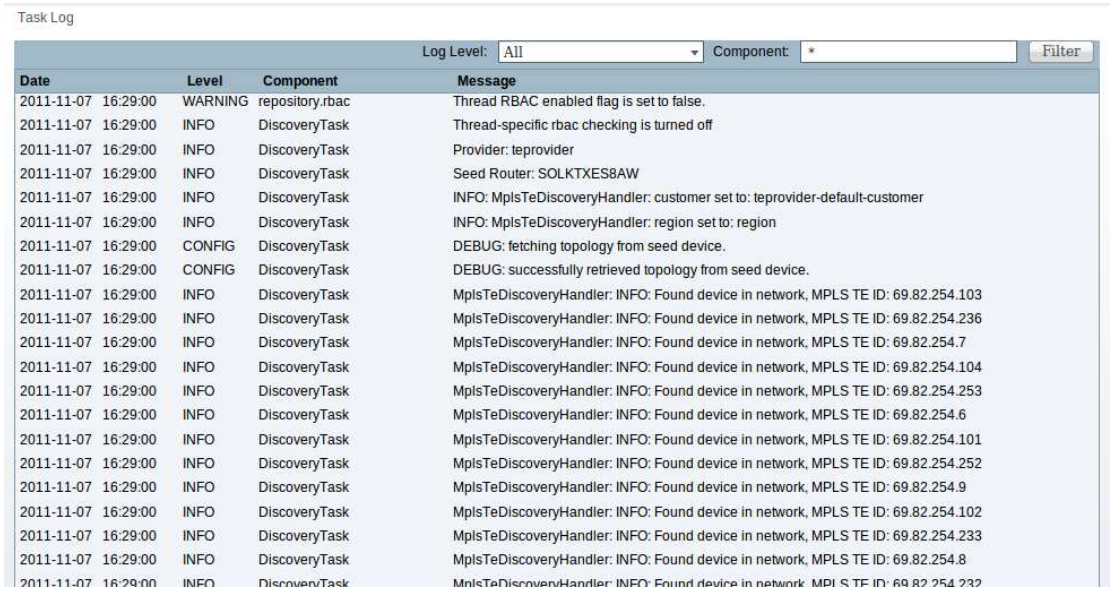
If the task is not completed and **Auto Refresh** is selected, the table continues to update periodically until it is completed.

Step 2 To view the log for a particular task, go to **Operate > Task Manager,** select the desired task, and then click the **View Log** button.

A copy of a TE Discovery log is shown in the following screenshots, starting with Figure 7-8. This first example shows the TE-enabled devices and links that TE Discovery has found in the topology. Once each device is identified, a set of debug, informational, warning and error logs are built up for each device to facilitate identification of errors.

Note To find the summary of changes in the network depicted in the following screenshots, scroll to the bottom of the log.

Figure 7-8 TE Discovery Task Log - Example 1


Task Log

Log Level: All Component: * Filter

Date	Level	Component	Message
2011-11-07 16:29:00	WARNING	repository.rbac	Thread RBAC enabled flag is set to false.
2011-11-07 16:29:00	INFO	DiscoveryTask	Thread-specific rbac checking is turned off
2011-11-07 16:29:00	INFO	DiscoveryTask	Provider: teprovider
2011-11-07 16:29:00	INFO	DiscoveryTask	Seed Router: SOLKTXES8AW
2011-11-07 16:29:00	INFO	DiscoveryTask	INFO: MplsTeDiscoveryHandler: customer set to: teprovider-default-customer
2011-11-07 16:29:00	INFO	DiscoveryTask	INFO: MplsTeDiscoveryHandler: region set to: region
2011-11-07 16:29:00	CONFIG	DiscoveryTask	DEBUG: fetching topology from seed device.
2011-11-07 16:29:00	CONFIG	DiscoveryTask	DEBUG: successfully retrieved topology from seed device.
2011-11-07 16:29:00	INFO	DiscoveryTask	MplsTeDiscoveryHandler: INFO: Found device in network, MPLS TE ID: 69.82.254.103
2011-11-07 16:29:00	INFO	DiscoveryTask	MplsTeDiscoveryHandler: INFO: Found device in network, MPLS TE ID: 69.82.254.236
2011-11-07 16:29:00	INFO	DiscoveryTask	MplsTeDiscoveryHandler: INFO: Found device in network, MPLS TE ID: 69.82.254.7
2011-11-07 16:29:00	INFO	DiscoveryTask	MplsTeDiscoveryHandler: INFO: Found device in network, MPLS TE ID: 69.82.254.104
2011-11-07 16:29:00	INFO	DiscoveryTask	MplsTeDiscoveryHandler: INFO: Found device in network, MPLS TE ID: 69.82.254.253
2011-11-07 16:29:00	INFO	DiscoveryTask	MplsTeDiscoveryHandler: INFO: Found device in network, MPLS TE ID: 69.82.254.6
2011-11-07 16:29:00	INFO	DiscoveryTask	MplsTeDiscoveryHandler: INFO: Found device in network, MPLS TE ID: 69.82.254.101
2011-11-07 16:29:00	INFO	DiscoveryTask	MplsTeDiscoveryHandler: INFO: Found device in network, MPLS TE ID: 69.82.254.252
2011-11-07 16:29:00	INFO	DiscoveryTask	MplsTeDiscoveryHandler: INFO: Found device in network, MPLS TE ID: 69.82.254.9
2011-11-07 16:29:00	INFO	DiscoveryTask	MplsTeDiscoveryHandler: INFO: Found device in network, MPLS TE ID: 69.82.254.102
2011-11-07 16:29:00	INFO	DiscoveryTask	MplsTeDiscoveryHandler: INFO: Found device in network, MPLS TE ID: 69.82.254.233
2011-11-07 16:29:00	INFO	DiscoveryTask	MplsTeDiscoveryHandler: INFO: Found device in network, MPLS TE ID: 69.82.254.8
2011-11-07 16:29:00	INFO	DiscoveryTask	MplsTeDiscoveryHandler: INFO: Found device in network, MPLS TE ID: 69.82.254.232

Figure 7-9 and Figure 7-10 show a sample device debug and information section.

Figure 7-9 TE Discovery Task Log - Example 2

```
2011-11-07  16:47:30  INFO    DiscoveryTask  <---------------------------->
                             Information summary for Te Router, Te Id: , Host name: WJRDUT307AW
                             -
                             - NEW: Te Router created, Mpls Te Id: 69.82.254.103
                             -
                             - Device Interfaces:
                             -
                             - EXISTING: Interface found with no changes, Name: MgmtEth0/RP0/CPU0/0, IP Address: 10.141.218.17
                             - EXISTING: Interface found with no changes, Name: TenGigE0/4/3/0, IP Address: 69.82.120.81
                             - EXISTING: Interface found with no changes, Name: Loopback10, IP Address: 10.214.254.103
                             - EXISTING: Interface found with no changes, Name: MgmtEth0/RP1/CPU0/0, IP Address: 10.141.218.18
                             - EXISTING: Interface found with no changes, Name: TenGigE0/4/1/0.1100, IP Address: 69.82.122.140
                             - EXISTING: Interface found with no changes, Name: TenGigE0/3/3/0, IP Address: 69.82.120.79
                             - EXISTING: Interface found with no changes, Name: Loopback0, IP Address: 69.82.254.103
                             - EXISTING: Interface found with no changes, Name: TenGigE0/4/4/0.1100, IP Address: 69.82.122.142
                             - EXISTING: Interface found with no changes, Name: TenGigE0/13/0/0, IP Address: 69.82.122.134
                             - EXISTING: Interface found with no changes, Name: TenGigE0/10/0/0, IP Address: 69.82.122.132
                             - EXISTING: Interface found with no changes, Name: TenGigE0/4/4/0.1250, IP Address: 69.82.77.128
                             - EXISTING: Interface found with no changes, Name: TenGigE0/3/0/0, IP Address: 69.82.77.48
                             - EXISTING: Interface found with no changes, Name: TenGigE0/4/1/0.1250, IP Address: 69.82.77.132
                             - EXISTING: Interface found with no changes, Name: TenGigE0/4/2/0, IP Address: 69.82.77.54
                             - EXISTING: Interface found with no changes, Name: TenGigE0/3/2/0, IP Address: 69.82.77.52
                             - EXISTING: Interface found with no changes, Name: TenGigE0/4/0/0, IP Address: 69.82.77.50
                             -
                             - Te Links:
                             -
2011-11-07  16:47:30  CONFIG  DiscoveryTask  <---------------------------->
                             -
                             Debug summary for Te Router, Te Id: 69.82.254.236, Host name: TWBGOHAA81W
                             -
                             DEBUG: Calling device for show version output: 69.82.254.236
```

Figure 7-10 TE Discovery Task Log - Example 3

```
2011-11-07  16:47:30  CONFIG  DiscoveryTask  <---------------------------->
                             -
                             Debug summary for Te Router, Te Id: 69.82.254.103, Host name: WJRDUT307AW
                             -
                             DEBUG: Calling device for show version output: 69.82.254.103
                             DEBUG: MplsTeShowVersionCallback: XDE show version invocation completed normally for device:
                             69.82.254.103
                             DEBUG: MplsTeShowVersionCallback: Device: 69.82.254.103, has an OS with version: 4.0.1[Default]
                             DEBUG: MplsTeShowVersionCallback: Device: 69.82.254.103, is running Cisco IOS XR.
                             DEBUG: Calling device for show running-config output: 69.82.254.103
                             DEBUG: Calling device for show primary tunnels output: 69.82.254.103
                             DEBUG: Calling device for show backup tunnels output: 69.82.254.103
                             DEBUG: MplsTeShowRunningCallback: XDE show running config invocation , MPLS TE ID:
                             69.82.254.103, completed normally.
                             DEBUG: MplsTeShowRunningCallback: Device has the following flags: rsvp graceful restart: false, te
                             enabled: true, conformant: true, supports FRR true, snmp traps enabled: true
                             DEBUG: Calling device for show auto-bw output: 69.82.254.103
                             DEBUG: MplsTeShowTunnelsCallback: show tunnels command completed successfully on device:
                             69.82.254.103, found tunnels: 1000 1001 1003 1004 1005 1006 1008 1009 1010 1013 1014 1017 1020
                             1023 1024 1025 1028 1029 10100 10101 10200 10201 10300 10301 10400 10401 10500 10501
                             10600 10601 10700 10701 10800 10801 10900 10901 11000 11001 11100 11101 11200 11201 11400
                             11401 11500 11501 11600 11601 11700 11701 11800 11801 11900 11901 12100 12101 12300 12301
                             12500 12501 12700 12701 14100 14101 14200 14201 15800 15801 16100 16101 16200 16201 16300
                             16301 16400 16401 18100 18101 18200 18201
                             DEBUG: Calling device for show supports subpool output: 69.82.254.103
                             DEBUG: MplsTeShowTunnelsBackupCallback: show backup tunnels command completed successfully
                             on device: 69.82.254.103, found backup tunnels: 1000 1001 1003 1004 1005 1006 1010 1013 1014
                             1017 1020 1023 1024 1025 1028 1029
                             DEBUG: MplsTeShowAutoBwCallback: XDE show auto bw invocation for device, MPLS TE ID:
                             69.82.254.103, completed normally.
                             DEBUG: MplsTeShowAutoBwCallback: Device: 69.82.254.103, supports auto bandwdith.
                             DEBUG: MplsTeShowSubpoolCallback: show supports subpool command completed successfully on
                             device: 69.82.254.103
                             DEBUG: MplsTeShowSubpoolCallback: this device supports subpool.
                             DEBUG: MplsTeShowRunningCallback: Device: WJRDUT307AW, has TE enabled interfaces:
                             TenGigE0/4/3/0, TenGigE0/4/1/0.1100, TenGigE0/3/3/0, TenGigE0/4/4/0.1100, TenGigE0/13/0/0,
                             TenGigE0/10/0/0
                             Device: WJRDUT307AW, has non TE enabled interfaces: MgmtEth0/RP0/CPU0/0, Loopback10,
                             MgmtEth0/RP1/CPU0/0, Loopback0, TenGigE0/4/4/0.1250, TenGigE0/3/0/0, TenGigE0/4/1/0.1250,
                             TenGigE0/4/2/0, TenGigE0/3/2/0, TenGigE0/4/0/0
                             DEBUG: MplsTeShowRunningCallback: Device: WJRDUT307AW, has explicit paths: WJRDUT307AW-
                             AURSCOTY7AW-1 WJRDUT307AW-AURSCOTY7AW-3 WJRDUT307AW-CLSPCOYK8AW-1
                             WJRDUT307AW-CLSPCOYK8AW-2 WJRDUT307AW-CLSPCOYK8BW-1 WJRDUT307AW-
                             HCHLILMT7AW-2 WJRDUT307AW-HLBOOR387AW-1 WJRDUT307AW-HLBOOR387AW-2
                             WJRDUT307AW-OMALNEXU7AW-4 WJRDUT307AW-RCKLCAIG7AW-1 WJRDUT307AW-
                             RCKLCAIG7AW-2 WJRDUT307AW-RCKLCAIG7AW-3 WJRDUT307AW-RCKLCAIG8AW-3
                             WJRDUT307AW-RCKLCAIG8AW-4 WJRDUT307AW-RCKLCAIG8BW-1 WJRDUT307AW-
                             RCKLCAIG8BW-2 WJRDUT307AW-RCKLCAIG8BW-3 WJRDUT307AW-RDMEWA227AW-1
                             WJRDUT307AW-RDMEWA227AW-3 WJRDUT307AW-RDMEWA227AW-4 WJRDUT307AW-
                             SCRMCAGN81W-1 WJRDUT307AW-SOLKTXES8AW-2 WJRDUT307AW-SOLKTXES8BW-1
                             DEBUG: MplsTeShowRunningCallback: Device: WJRDUT307AW, has tunnels: 1003 1004 1001 10500
```

289256

Step 3 Click **Return to Logs** to quit the current log with the option to open another log.

TE Topology

The TE Topology tool provides a visual snapshot of the current state of the network. It cannot be used to determine changes that have taken place in the network.

The steps required to generate a topology graph of the network are described in TE Topology, page 7-81.

View Network Element Types

Another way to check the state of the network after running TE discovery is to go to the Traffic Engineering menu options and select the type of elements you want to verify.

For example, to check the status of the nodes after running TE discovery, choose **Traffic Engineering > Nodes**. Look at the updated list of TE nodes to assess which nodes are in the network.

Do the same for TE Links, TE Primary Tunnels, TE Backup Tunnels, and so on.

Setting Up Management Interfaces

Before commencing tunnel management operations, you need to set up management interfaces. However, this step is only necessary if the network devices are not accessible by the hostname from the management station.

For a detailed description of how to set up management interfaces on specific devices, see Devices, page 2-1.

MPLS-TE Management Process

The MPLS-TE management process involves the following steps:

1. Enable MPLS-TE on the network devices and make sure that the IP addresses used as the devices TE IDs are accessible from the management station (this step is not supported by TEM).
2. Prepare the repository for discovering MPLS-TE network.
3. Set up management interfaces for the discovered devices or update the server host file with resolution for all discovered devices. Again, this is not needed if the hostnames are already accessible from the management station.
4. Discover the MPLS-TE network.

You will then be in a position to run the other MPLS-TE functions available in TEM.

Note When the repository is empty, or when the management IP addresses are not configured for current devices in the TE network, make sure that the router MPLS TE ID can be reached from the management station. In other words, the TE discovery process does not support seed passthrough.

Configuring Ethernet Links

Only point-to-point links are supported in TEM. POS links are point-to-point by default but otherwise Ethernet links need to be configured as point-to-point.

For IOS, enter the following command:

```
(config-if)# ip ospf network point-to-point
```

For IOS XR, enter the following command:

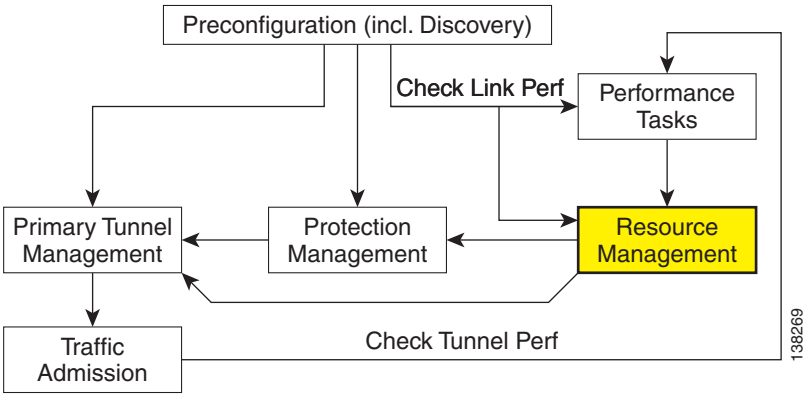
```
# router ospf <id> area <area identifier> interface <name> network point-to-point
```

TE Resource Management

TE resource management is defined as the tuning of certain properties on the TE interfaces to optimize the tunnel placement.

The highlighted box in Figure 7-3 shows where in Prime Fulfillment resource management occurs.

Figure 7-11 Prime Fulfillment Process Diagram - Resource Management



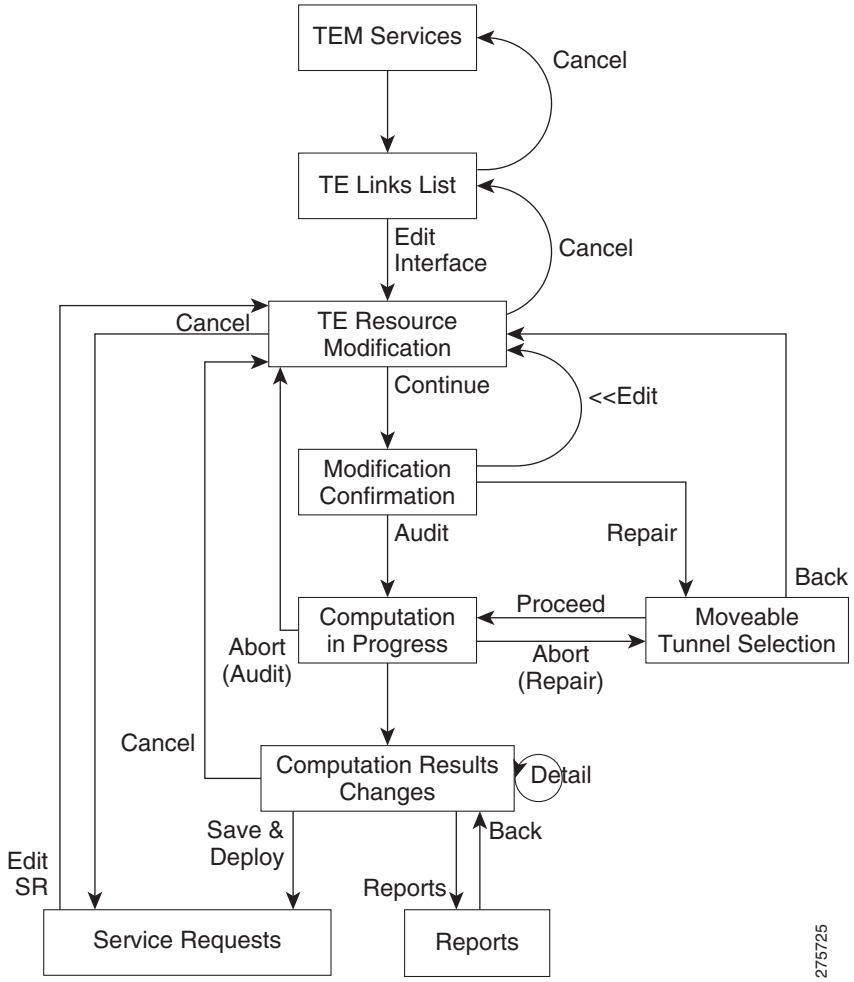
When a tunnel placement is attempted and there is insufficient bandwidth, sometimes the resources on the TE links can be changed and the tunnel placement retried.

Network resources in this context are understood to be routers in the TE network, the interfaces that connect them, and the RSVP bandwidths and other properties configured on the links. Because Prime Fulfillment relies on the discovery process to add the network elements to the repository, the resources must be discovered before resource management can be performed.

TE resource management is a manual process that should be performed on an as needed basis. If the original configuration is already optimal, there is no need to do any resource management tasks. If subsequent discovery unveils any discrepancy, or if you experience difficulty achieving desired results in protection planning or placing primary tunnels, adjustments on the resources might be warranted.

An overview of the resource management process is provided in Figure 7-12.

Figure 7-12 Resource Management Processes



This section includes the following:

- Modifying Network Resources, page 7-21
- Changing Link Status, page 7-23
- Deleting TE Links, page 7-24
- Deleting TE Tunnels, page 7-25
- Deleting TE Nodes, page 7-26.

Modifying Network Resources

The resource management tasks are mainly carried out from the TE Links List window.

Note Certain attributes, such as Description, that do not impact the computation carried out by these tools and updates to these are, therefore, not displayed in the computation results window.

To modify a TE link, use the following steps:

Step 1 Choose **Traffic Engineering > Links**.

The TE Links List window appears.

The links list shows the current active links in the TE network. Use the arrows to page forward as needed.

Step 2 Select the desired link in the links list.

Note **Admin Status**—Indicates whether the link is **UP** or **DOWN**. This is local to Prime Fulfillment. It is not the network interface status.

Step 3 Click **Edit > Interface A** or **Edit > Interface B** to edit one of interfaces on the link.

Note If a non-Cisco interface is selected for editing, changes made in the Edit window will be saved in the ISC repository but they will not be deployed.

The TE Resource Modification window appears. It includes the following fields:

- **Max Global (BC0) Reservable**—Maximum amount of bandwidth in kbps that can be reserved by TE Tunnels.
- **Max Sub Pool (BC1) Bandwidth**—Maximum amount of bandwidth in kbps that can be reserved by sub pool TE Tunnels. The range is from 1 to the value of **Max Global Reservable**.
- **Attribute Bits**—Links attributes to be compared to a tunnel's affinity bits during selection of a path. Valid values are from 0x0 to 0xFFFFFFFF, representing 32 attributes (bits) where the value of an attribute is 0 or 1.
- **TE Metric**—Metric used to override the Interior Gateway Protocol (IGP) administrative weight (cost) of the link.
- **Propagation Delay**—The time it takes for traffic to travel along a link from the head interface to the tail interface.
- **Max Delay Increase**—Used in computations of FRR backup-tunnels to constrain the propagation delay of a backup-tunnel for the link. A max delay increase for a link might need to be set to loosen the delay constraint when generating backup tunnels, as it is difficult to find backup tunnel paths where there is no increase in the delay compared with the flow being protected.
- **Link Speed Factor**—Multiplication factor corresponding to the amount (percentage) of link speed available for primary and backup traffic. This is typically set to 1.

Step 4 Make the desired modifications and click **Continue** to proceed to the confirmation page to verify the changes or click **Cancel** to quit without saving.

Step 5 Click **Edit** to return to the editable window or proceed in one of the following ways:

- **Proceed with Changes** —Perform Tunnel Audit or Tunnel Repair.

  For a detailed explanation of Tunnel Audit and Tunnel Repair, see Advanced Primary Tunnel Management, page 7-44

  If a non-Cisco device is edited, **Proceed with Changes** will be disabled. Instead, **Save & Deploy** is enabled and the changes can be saved (not deployed).

- **Save & Deploy**—If the changes made do not affect tunnel placement, click **Save & Deploy** to proceed. In this case, there is no need for performing Tunnel Audit or Tunnel Repair.

Note When you click **Save & Deploy**, a background process is started. To avoid a potential conflict with another deployment, wait until the service request (SR) has completed the Requested and Pending states before deploying another SR with Save & Deploy. To see the state of deployment, go to **Operate > Service Request Manager** or open **Operate > Task Manager**.

Note In Prime Fulfillment, service requests (SRs) are generally deployed from each TE service, not from the **Operate > Service Request Manager** page with the exception of the TE Traffic Admission SR.

After deployment, the SR status can be viewed from the SR window at **Operate > Service Request Manager**.

For more information on working with service requests, see the managing service requests part elsewhere in this guide.

If the SR does not go to the **Deployed** state, go to the Task Log to see the deployment log (**Operate > Task Manager > Logs**). Task logs are further described in Task Logs, page 7-16.

Changing Link Status

From the TE Links List window, you can also find out what effect it will have if a link is taken offline. This approach can be used to move tunnels off a link before actually shutting down the interface.

Note Link status in Prime Fulfillment is of local significance. Changing link status as described in this section is not provisioned down to the network.

To change the link status, use the following steps:

Step 1 Choose **Traffic Engineering > Links**.

The TE Links List window appears.

Step 2 Select one or more links and click the **Change Status** button.

Step 3 Select **Enable** or **Disable** to enable or disable the selected link.

As an example, selecting **Disable** will change the link status to **DOWN**.

Similarly, use **Enable** to change the status back to **UP**.

Step 4 Click **Proceed with Changes** to assess any impact on tunnel placement using Tunnel Audit or Tunnel Repair and deploy the changes.

For a detailed explanation of Tunnel Audit and Tunnel Repair, see Advanced Primary Tunnel Management, page 7-44.

Deleting TE Links

The TE Link List window includes a delete function (the **Delete** button), which allows you to delete a TE link and the TE interfaces at each end of the link from the repository. It does not make any change to the physical link in the network.

Link deletion can be selected based on a specific TE provider. When deleting different links belonging to different providers, first choose the appropriate provider and then mark the links to be deleted.

Also, simultaneous deletion of multiple links of the same provider is supported.

Restrictions

The Prime Fulfillment GUI prevents you from deleting a link if any TE object is still using that link.

It checks the following objects:

- strict explicit paths
- protected interfaces of backup tunnels
- SRLGs
- protected elements
- TE resource SRs.

If there are any primary or backup tunnels traversing the path options, an error report will be displayed. Otherwise, a message will be displayed seeking confirmation that the above set of associated objects should be deleted.

Use Case

In this example, we will look at the procedure required when attempting to delete a link that could be traversed by primary or backup tunnels.

Use the following steps:

Step 1 Choose **Traffic Engineering > Links**.

Step 2 Select a link by checking the corresponding check box.

Step 3 Click the **Delete** button.

Step 4 Two things can happen:

- A tunnel with path option traverses the link: The link deletion will fail and you will be prompted to reroute or delete those tunnels before trying link deletion again. This will take you to the TE Links List page.
- No tunnels with path option traverses the link: A list of TE associated objects will be displayed for that link and you will be prompted to confirm whether you agree to the automatic deletion of TE Link associated objects or have second thoughts and would like to cancel the link deletion transaction.

Step 5 After any necessary tunnels have been rerouted/deleted and link deletion is attempted, a list of objects that are still associated will be displayed.

Step 6 If you want to delete associated TE objects listed after rerouting/deleting primary tunnels, you will get directed to a new window showing the progress of the transaction only when there are tunnels offering backup link protection/protecting multiple interfaces. If there are no tunnels offering backup link protection/protecting multiple interfaces, you are directed to the TE Links window on successful/failure transaction from the associated TE objects list page.

See the note below on associated TE objects.

Step 7 After all the associated objects have been deleted, you will be directed to the TE Links List window.

Note on Associated TE Objects

Associated TE objects can be any of the following:

- strict explicit paths and loose explicit paths (with strict hop type) traversing the link;
- backup tunnels offering link protection;

Note The link will be removed from any SRLGs (if SRLG has more than one link) or both the link and the SRLG will be removed if the link marked for deletion is the only one in the SRLG.

- resource SRs;
- protected elements.

The associated TE objects in the above list vary depending on the way the link is configured in TEM.

For example, if associated TE objects have backup tunnels offering link protection, you will be directed to the Link Deletion Progress window where protected interfaces will be updated accordingly for the available TE links and backup tunnel SRs will get re-deployed. Otherwise, if no backup tunnels offering link protection qualify as associated TE objects, the remaining TE objects will automatically be removed from the window showing the associated TE objects.

Deleting TE Tunnels

TE Tunnels can be deleted in the TE Links List window or in the individual primary or backup tunnel SR windows (see Delete Primary Tunnel, page 7-38 or Delete Backup Tunnel, page 7-44).

In the TE Links window, the reason for wanting to delete a tunnel will often be a need to delete a link that is traversed by one or more tunnels.

To delete a tunnel in the TE Links List window, use the following steps:

Step 1 Choose **Traffic Engineering > Links.**

Step 2 Select the link for the tunne that you wish to delete and click the **Show Tunnels** button.

This brings up a tunnel filter where you can select the category of tunnel you wish to display (**All**, **Managed**, **Unmanaged**, **Backup**).

Step 3 Select one of these tunnel categories.

This brings up a list of all tunnels in the selected filter category, which traverses the link.

Step 4 Select one or more tunnels that you wish to delete and click the **Delete** button.

This will delete the tunnels selected by starting a new provisioning operation.

Deleting TE Nodes

You can also delete a TE node. This works in a very similar way to deleting a link but is done from the PE devices screen. By deleting the corresponding PE device, you effectively delete the TE node.

Similar restrictions apply as in the case of TE links. The delete operation can only be succeed if no TE objects are using the node.

Restrictions

The Prime Fulfillment GUI prevents you from deleting a node if any TE object is still using that node.

As with TE links, it checks the following objects:

- strict explicit paths
- protected interfaces of backup tunnels
- SRLGs
- protected elements
- TE resource SRs.

In addition, the node deletion checks that no managed, unmanaged, or backup tunnel starts or ends at the node in question.

If any of these objects is using the node, an attempt to delete the node will result in an error message and the node and its interfaces remain unchanged.

Use Case

An example of this feature is when a TE router is to be decommissioned from the network and replaced by one or more new TE routers as part of a major topology change.

The steps needed to enable you to delete this node might include the following:

1. Reroute all managed tunnels away from this node using Tunnel Repair.
2. Reroute all unmanaged and backup tunnels using the node as part of their path away from it.
3. Delete any backup tunnels that protect either of the intefaces that make up the node.
4. Delete any explicit paths that use the node.
5. Delete the node from the repository from the TE Links List window.
6. Outside Prime Fulfillment, during a suitable outage window, physically decommission the node, and set up its replacement(s).
7. Run a new TE discovery task, which result in the newly added nodes being added to the repository.
8. Depending on the FRR requirements of the network, protect the new node(s) using Compute Backup. (See Compute Backup, page 7-63.)
9. Run network grooming (see Grooming, page 7-58) to optimise the managed tunnels, so that they will make use of the new node(s).

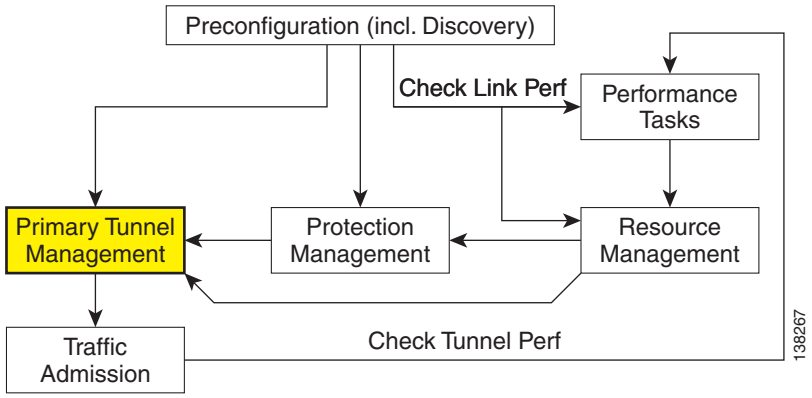
If this check succeeds, the TE node and all TE links and TE interfaces starting at that node are removed from the repository.

Basic Tunnel Management

This section describes the processes involved in creating primary and backup tunnels with Prime Fulfillment. To create a tunnel, certain steps must first be performed as described in previous sections.

The highlighted box in Figure 7-3 shows where in Prime Fulfillment primary tunnel management occurs.

Figure 7-13 Prime Fulfillment Process Diagram - Primary Tunnel Management



Primary tunnels are characterized by carrying traffic during normal operation. They have a prioritized list of possible paths, by which traffic can be routed. At any one time, the highest priority path available will be used to route traffic. If this fails, traffic will normally be rerouted via the next available path until a higher priority path becomes available again.

Prior to setting up the tunnel, a TE policy governing the traffic must be defined. An explicit path is created to establish the route and, in the case of a primary tunnel, it is created as either a managed or an unmanaged tunnel.

The purpose of a backup tunnel is to carry Fast Re-Route (FRR) protected traffic around a failed element until the routing in the network has reconverged. It is intended to protect traffic travelling along primary tunnels. There can be many backup tunnels protecting the same traffic through the use of load balancing.

If the network fails to reconverge, the backup tunnel will remain in place.

The difference between managed and unmanaged tunnels is described in the section on Managed/Unmanaged Primary Tunnels in the *Cisco Prime Fulfillment Theory of Operations Guide 6.2*.

The concept of bandwidth pools from which tunnels reserve bandwidth is important to understand. This is described in the section on Bandwidth Pools in the *Cisco Prime Fulfillment Theory of Operations Guide 6.2*.

This section includes the following:

- Create TE Policy, page 7-28
- Create Explicit Path, page 7-29
  - Delete Explicit Path, page 7-31

Create TE Policy

To create a primary tunnel, each primary tunnel must be associated with a policy. A policy can be used by multiple tunnels.

For backup tunnels, this step is not necessary. In this case, proceed to Create Explicit Path, page 7-29.

For other TE policy management operations, see TE Policies, page 7-70.

The TE policy is a set of rules governing the TE network and defines the Class-of-Service (for example, gold, silver, bronze) for primary tunnel traffic.

Prime Fulfillment has a notion of **Managed** and **Unmanaged** policies. **Managed** policies have setup/hold priorities of 0/0 and can have additional routing constraints such as protection level and max delay. Tunnels with **Unmanaged** policies are provisioned by the system, but the system only tracks the deployment, not the operation of the tunnel. **Unmanaged** policies cannot have a setup/hold priority of zero.

For more information about managed and unmanaged primary tunnels, see the section on Managed/Unmanaged Primary Tunnels in the *Cisco Prime Fulfillment Theory of Operations Guide 6.2*.

Policies are managed under **Policies** in **Service Design**. For a more detailed explanation of the **Policies** GUI, see TE Policies, page 7-70.

To create a TE policy, use the following steps:

Step 1 Choose **Traffic Engineering > Policy Manager**.

The Policy Manager window appears.

Step 2 Click **Create** and select **TE Policy** to set up a new TE policy.

To edit an existing policy, select the policy that you want to modify and click **Edit**. The TE Policy Editor window appears.

Note A policy that is being used by a tunnel cannot be modified. However, the name and ownership of an in-use policy can be changed.

For an explanation of the various window elements, see TE Policies, page 7-70.

Step 3 Fill in the required fields marked with an asterisk (*) and any optional fields.

If you intend to use the TE policy for managed tunnels, make sure to check the **Managed** check box.

When setting up a policy for a managed tunnel, the **Setup** and **Hold** priorities are automatically set to zero (highest priority). In the case of a policy for an unmanaged tunnel, you can specify the desired **Setup** and **Hold** priority settings.

Step 4 Click **Save**.

Create Explicit Path

Paths are defined between source and destination routers, possibly with one or more hops in between. Paths are used for primary and backup tunnels in the explicit path option(s).

If you intend to create an explicit path for managed tunnels, the path should not contain any non-TE enabled interfaces. Paths with non-TE enabled interfaces will be filtered out by the tunnel path chooser of the tunnel editor for managed tunnels and backup tunnels (not unmanaged tunnels).

To create or edit an explicit path, use the following steps:

Step 1 Choose **Traffic Engineering > Explicit Paths**.

The TE Explicit Path List window appears.

Step 2 To create an explicit path in the **TE Explicit Path List**, click **Create**.

The New TE Explicit Path window appears.

To edit an explicit path in the explicit path list, select the explicit path that you want to modify and click **Edit**. This opens the TE Explicit Path Editor window.

Note An explicit path that is being used by a tunnel cannot be modified. However, use Edit to view the path.

The New TE Explicit Path window includes the following GUI elements:

- **Path Name**—Name of explicit path.
- **Head Router**—Name of the head router.
- **Path Type**—Three types of explicit paths are supported:
  - **STRICT**—All strict hops are defined in the path.
  - **LOOSE**—Any loose hops (pure loose path or a combination of loose and strict hops) are defined in the path.
  - **EXCLUDE**—All exclude hops are defined in the path.
- **Links** (table)—Lists the links added for the current path and includes the following information:
  - **Device**—Hostname of the TE device that the path originates from.
  - **Outgoing Interface**—Interface name of the outgoing interface from the originating device.
  - **Outgoing IP**—IP address of the outgoing interface.
  - **Next Hop**—Hostname of the next hop device.
  - **Incoming Interface**—Incoming interface name on the next hop device.
  - **Incoming IP**—Incoming interface IP address on the next hop device.

- **Provision Preference**—Preference for provisioning the **next-address** subcommand of the **ip explicit-path** command. Choose between **Outgoing Interface** and **Incoming Interface**.
  - **Outgoing Interface**—Outgoing interface on the router.
  - **Incoming Interface**—Incoming interface on the router.

Note If a path is used by any tunnel, no modifications are possible. The **Outgoing Interface** and **Incoming Interface** links are not selectable and the Provision Preference line and the **Add Link**, **Delete Link**, and **Save** buttons disappear.

Step 3 Specify a pathname and select a head router.

Step 4 Select a path type:

- **Strict**: If **Strict** is chosen, use the current panel that lists the connected links one by one until destination is reached.
- **Loose**: If **Loose** is selected, a new hop is added by entering the IP address. If **Strict** is selected, you are allowed to select from TE Links list only.

Note For IOS XR, the **Loose** type is only available if the head device is running IOS XR 3.4 or later.

Note If **Loose** is chosen, a new panel that adds a loose hop definition one by one is listed. Because a combination of strict and loose hops is allowed for a loose explicit path definition, the flexibility of including strict hops is provided with a constraint of at least a loose hop presence in the path.

- **Exclude**—**Exclude** allows you to specify an exclude IP address. See Step 6.

Step 5 If **Strict** was selected, click the **Add Link** button to add a blank line to the hop list table.

If **Loose** or **Exclude** was selected, an **Add Hop** button appears, which when clicked opens a pop-up window where you specify an IP address.

Step 6 Now an interface must be selected for the head router.

Depending on the path type selection, you will see one of the following windows:

A. Strict path type:

Click the **Add Link** button, then click **Add Interface**. The Select Next Hop window appears.

The next hop list contains all the possible next hops of the router, excluding the ones already included in the explicit paths (to avoid path loops).

The next hop list contains TE interfaces and at most one non-TE interface for each router (if the loopback interface is used as the MPLS TE ID of the device). For TE interfaces, the **Outgoing Interface** and **Outgoing IP** columns are populated by the application.

Note If a non-TE interface is selected, **Provision Preference** is set to **Incoming Interface**. The provision preference cannot be set manually.

Select an interface and click **Select**. The corresponding link information is added to the new explicit path in the **Links** table.

In the New TE Explicit Path window, both the incoming and outgoing interface fields are populated.

B. Loose path type:

Click the **Add Hop** button. The Loose Hop Definition window appears.

In this window, specify an IP address for the desired loose hop and click **OK**. The Loose Hop Definition window closes.

The New TE Explicit Path window now displays the added loose hop.

C. Exclude path type:

Click the **Add Hop** button. The Exclude Hop Definition window appears.

In this window, specify an IP address for the desired exclude hop and click **OK**. The Exclude Hop Definition window closes.

The New TE Explicit Path window now displays the added exclude hop.

Step 7 To add another link, click either **Add Link** or **Add Hop**.

Step 8 For Strict hops, a **Provision Preference** can optionally be selected by clicking either the **Outgoing Interface** or the **Incoming Interface** radio button.

Note If you try to select the **Provision Preference** before adding a link when non-TE interfaces are present, the **Add Link** process overrides the **Provision Preference** and sets it to incoming.

Step 9 Click **Save** to keep the created TE explicit path or click **Cancel** to quit without saving.

Delete Explicit Path

Prime Fulfillment supports decommission of explicit paths when deleting/decommissioning primary/backup tunnels. This is only supported for IOS XR.

Whether an explicit path can be deleted in such situations depends on whether they are used by other global applications.

Explicit path deletion goes hand in hand with both SR tunnel deletion for primary managed/unmanaged tunnels, backup tunnels, and any non-conformant tunnels and is applicable to all path option types (STRICT, LOOSE, EXCLUDE).

An explicit path configuration will be automatically removed by Prime Fulfillment when the explicit path is no longer used by any tunnel in the system due to a change in tunnel configuration. This situation occurs when tunnels are deleted or when tunnels are rerouted in Prime Fulfillment.

When the explicit path configuration is removed from the device, the explicit path will still exist in the Prime Fulfillment database. Such explicit paths remaining in the database can be reused.

Explicit paths do not get deleted if you reroute or delete the tunnel(s) outside of Prime Fulfillment (through CLI on the device itself, for example). However, when a transaction reroutes, deletes, or modifies a tunnel using Prime Fulfillment so that an explicit path is no longer used by any tunnels, that explicit path configuration will automatically be removed from the device.

Primary Tunnel Operations

Prime Fulfillment allows you to perform a number of primary tunnel operations, which are described in the following sections.

Create Primary Tunnel

After a TE Policy and an explicit path have been set up, a primary tunnel can be created. There are two types of primary tunnels:

- Managed Primary Tunnels
- Unmanaged Primary Tunnels

Below, the GUI flow is described for creating unmanaged primary tunnels. It is very similar for managed primary tunnels and the few differences that exist are described in the section Managed/Unmanaged Primary Tunnels in the *Cisco Prime Fulfillment Theory of Operations Guide 6.2*.

To create a managed or an unmanaged primary tunnel, use the following steps:

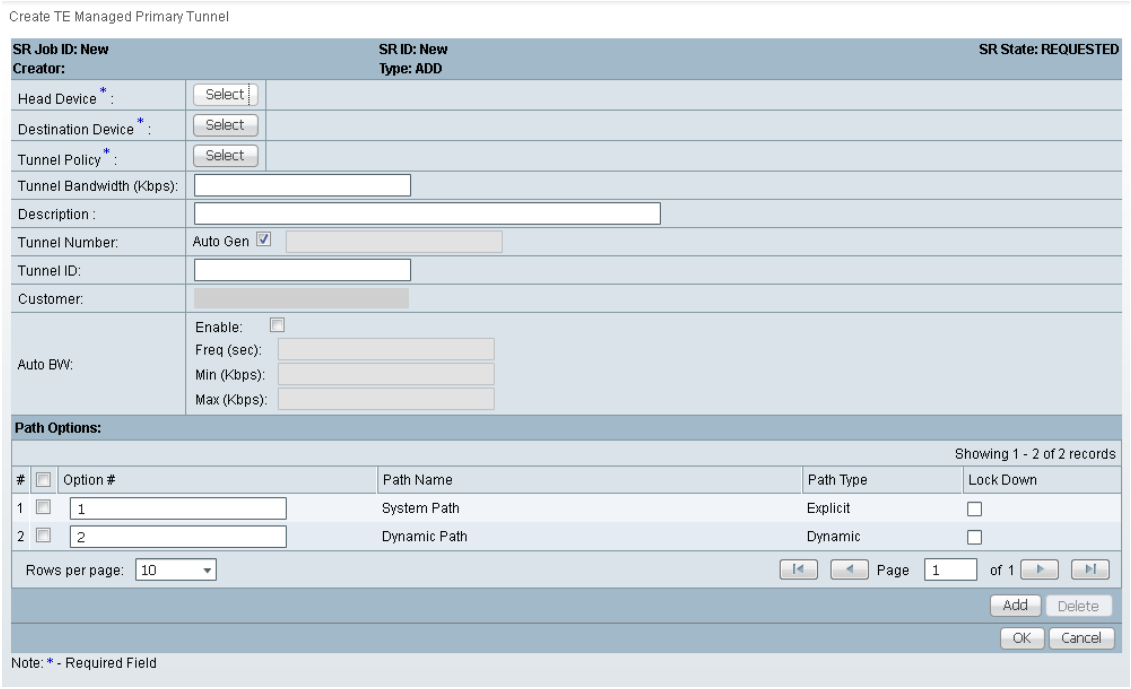
Step 1 Choose **Traffic Engineering**.

Step 2 Click **Create Managed TE Tunnel**. The TE Managed Primary Tunnels SR window appears as shown in Figure 7-14.

or

Click **Create Unmanaged TE Tunnel**. The TE Unmanaged Primary Tunnels SR window appears.

Figure 7-14 Create TE Managed Primary Tunnel



The TE Managed Primary Tunnels SR window includes the following elements:

- **Op**—SR operation on the tunnel. This can be one of the following:
  - **ADD**—Indicates a newly added tunnel.
  - **MODIFY**—Indicates a modified existing tunnel.
  - **DELETE**—Indicates an existing tunnel to be deleted.

  - **ADMIT**—Indicates an existing tunnel to be admitted by tunnel computation.
- **Tunnel ID**—Unique tunnel identifier used within Prime Fulfillment.
- **T#**—Tunnel number on the head router.
- **Head**—Hostname of the head router.
- **Dest**—Hostname of the destination router.
- **Policy**—TE policy for the tunnel.
- **BW**—The tunnel bandwidth. If the tunnel is auto-bw enabled, BW shows the higher of tunnel bandwidth and maximum automatic bandwidth.
- **AutoBW**—Auto Bandwidth enabled if **true**, otherwise **false**.
- **Deploy Status**—Tunnel deployment status.
- **Verified**—Indicates whether tunnel verification was successful (**succeed**, **failed**, or **unknown**).
- **Allow Reroute**—Specifies whether reroute is allowed (**true** or **false**). If reroute is not allowed, the tunnel cannot be set to movable, and hence cannot be rerouted by the operation (placement, grooming, or repair).
- **Head Region**—The region to which the head router belongs.
- **Tail Region**—The region to which the tail router belongs.

The following actions can be performed (buttons):

- **Display**—Open a Topology Display for the network and highlight the selected primary tunnel(s). Selected tunnels are marked in color with directional arrows.
- **Details**—Open the TE Tunnel Details window, which provides type, status, LSP, and other information about the tunnel.
- **Admit**—Admit selected tunnels not previously verified into the managed topology. This feature is used only for discovered tunnels that failed verification or for migrating unmanaged tunnels.
- **Create**—Create a managed primary tunnel.
- **Edit**—Edit a selected primary tunnel.
- **Delete**—Delete selected primary tunnels.
- **Import**—Import tunnel data from import XML file.
- **Placement Tools**—These tools are available only when no change has been made to the tunnels. Apply the following functions against the current topology and tunnels:
  - **Groom**—Analyse the managed tunnels in the network and reroute them to reduce the maximum link utilization.
  - **Tunnel Audit**—Determine if changes to previously made SRLGs or backup tunnels have caused constraint violations in managed tunnels (this can occur when managed tunnels have FRR protection constraints).
  - **Tunnel Repair**—Repair any managed tunnel constraint violations revealed by **Placement Tools > Tunnel Audit**.
- **Update Tunnel ID**—Update Tunnel ID(s) directly in the repository without deploying the corresponding tunnel(s).
- **Proceed with Changes** —For verifying changes in tunnels. When tunnels have been created, deleted, admitted, or their attributes altered, you can proceed with one of the following placement tools:
  - **Tunnel Audit**—Checks what constraint violations modifications to tunnels might cause.

  - **Tunnel Placement**—Admit new tunnels and modify tunnels already admitted into the network.
  - **Tunnel Repair**—Resolve inconsistencies caused by changes to bandwidth requirements or delay parameters of existing tunnels by moving as few existing tunnels as possible to accomodate the changes.

Note that for the unmanaged tunnels list, the last two columns in the managed tunnels list (Verified and Allow Reroute) are replaced by the Conformance column.

In the following example, an unmanaged tunnel is created.

Step 3 Click **Create**.

The Create TE Unmanaged Primary Tunnel window appears.

The Create TE Managed Primary Tunnel window and Create TE Unmanaged Primary Tunnel window have only minor differences and include the following elements:

- **Head Device**—Head device for the tunnel.
- **Destination Device**—Destination device for the tunnel.
- **Tunnel Policy**—A set of rules established for a tunnel.
- **Tunnel Bandwidth**—Total allocated bandwidth of the tunnel.
- **Description**—Descriptive text to help identify the tunnel.
- **Tunnel Number**—Tunnel number corresponding to the tunnel interface name.
  - **Auto Gen**—Check this box to generate the tunnel number automatically. Otherwise, enter a desired number.

Note If a manually entered tunnel number is too low, it could prevent deployment.

Note MPLS-TE tunnels can potentially interfere with multicast GRE tunnels. Prime Fulfillment creates new tunnels using auto-gen and this tunnel number might already be used by an MDT GRE tunnel. As a result, Prime Fulfillment uses high tunnel numbers to avoid any complications.

- **Tunnel ID—**Unique tunnel identifier used within Prime Fulfillment.
- **Customer**—Selected customer for the tunnel.
- **Auto BW**—A way to configure a tunnel for automatic bandwidth adjustment and to control the manner in which the bandwidth for a tunnel is adjusted.
  - **Enable**—Check this box to enable automatic bandwidth.
  - **Freq**—Interval between bandwidth adjustments.
  - **Min**—Minimum automatic bandwidth, in kbps, for this tunnel.
  - **Max**—Maximum automatic bandwidth, in kbps, for this tunnel.

Path options:

- **Option #**—Sequential number of available explicit paths.
- **Path Name**—Name of the explicit path. In case of an existing path, the name is a URL that links to the Explicit Path Viewer.

  - **System Path**—System generated explicit path. For managed tunnels, the first path has to be an explicit path. If a tunnel contains a system path, the planning function will generate an optimal path for the tunnel.
  - **Dynamic Path**—A dynamic path is provisioned by allowing the head router to find a path. The **dynamic** keyword is provisioned to the routers.
- **Path Type**—Path option type, Explicit or Dynamic.
- **Lock Down**—Disables reoptimization check on the tunnel, if checked, meaning the path cannot be changed.

Step 4 To select a **Head Device** in the Create TE Unmanaged Primary Tunnel window, click the corresponding **Select** button to open the Select Device for TE Head Router window.

Step 5 Select a device name and click **Select**.

The Select Device for TE Head Router window closes and the prompt returns to the Create TE Unmanaged Primary Tunnel window.

Step 6 To select a **Destination Device** in the Create TE Unmanaged Primary Tunnel window, click the corresponding **Select** button to open the Select Device for TE Tail Router window.

Step 7 Select a device name and click **Select**.

The Select Device for TE Tail Router window closes and the prompt returns to the Create TE Unmanaged Primary Tunnel window.

Step 8 To select a **Tunnel Policy** in the Create TE Unmanaged Primary Tunnel window, click the corresponding **Select** button to open the Select Unmanaged TE Tunnel Policy window.

Note When creating a managed tunnel, make sure that one or more managed tunnel policies are available. If that is not the case, go to **Policies** (see Create TE Policy, page 7-28) and make sure to check the **Managed** check box.

Step 9 Select a policy and click the **Select** button.

This brings you back to the tunnel editor.

Step 10 Click **Add** to set up path options for the tunnel. The Select TE Explicit Path window appears.

The **Path Options** section provides two path types:

Explicit Path—A fixed path from a specific head to a specific destination device that includes three types of paths: **Strict**, **Loose**, and **Exclude**.

Dynamic Path—A dynamic path is provisioned by allowing the head router to find a path. The **dynamic** keyword is provisioned to the routers.

Step 11 Select the desired TE Explicit Path unless you prefer dynamic path only.

If none is available, you can set one up first. To do so, see Create Explicit Path, page 7-29.

Step 12 Click **Select**.

The selected path appears in the **Path Options** section of the create window.

For explicit paths (<*head_device*>-<*destination_device*>), you can click the pathname to open the non-editable Explicit Path Viewer.

For an explanation of the various window elements, see Create Explicit Path, page 7-29.

Step 13 In the Create TE Unmanaged Tunnel window, click **OK** to accept the entered tunnel information or click **Cancel** to quit and return to the TE Unmanaged Primary Tunnels SR window.

The TE Unmanaged Primary Tunnel SR window appears with the newly created SR with the **Op** field set to ADD.

Note The added tunnel can be reverted from the ADD state to its original state by selecting it and clicking **Delete**. The tunnel is removed from the tunnel list.

Step 14 In the TE Unmanaged Primary Tunnel window, click **Save & Deploy** (see Note on page 36) to either deploy the new tunnel SR to the network or force deploy all tunnels, or you can create or edit more primary tunnels and then save and deploy all changes.

When you click **Save & Deploy**, Prime Fulfillment locks the TE routers effected, which will block any subsequent SRs which use that TE Router until the SRs are finished. It is safe to try and deploy other SRs in the system. If there is any conflict with the SR currently being processed, Prime Fulfillment will simply ask you to wait until it is complete.

To see the state of deployment, go to the Service Requests window at **Operate > Service Request Manager** or open **Operate > Task Manager**.

- **Save & Deploy**—For committing tunnel changes that do not impact tunnel placement. There are two options for saving and deploying SR tunnels to the network:
  - **SR Tunnels Only**—Deploy all tunnel changes that does not impact tunnel placement, or if no changes were made to the SR, use this to redeploy the SR that was in **Requested** or **Invalid** state.
  - **Force Deploy All Tunnels**—Force deployment of all tunnels in this SR. This could be useful when previous provisioning of the SR has failed, so that it is necessary to force through the deployment of all tunnels in the SR.

Note You might see Elixir Warnings during TE Tunnel deployment. The deployment will be successful and the warning messages can safely be ignored.

Note For managed tunnels, you cannot deploy the service request until you have used the **Proceed with Changes** button to perform either Tunnel Placement, Tunnel Audit, or Tunnel Repair (see Advanced Primary Tunnel Management, page 7-44).

Note With the exception of TE Traffic Admission SRs, TE SRs are always deployed immediately from the specific TE SR window, not from **Operate > Service Request Manager**.

The Service Requests window (**Operate > Service Request Manager)** appears and displays the state of the deployed SR (first REQUESTED, then PENDING, then DEPLOYED, if successful).

For more information on working with service requests, see the managing service requests part elsewhere in this guide.

If the SR does not go to the **Deployed** state, go to the Task Logs window to see the deployment log (**Operate > Task Manager > Logs**) as described in SR Deployment Logs, page 10-46.

To edit the service request from the **Service Request Manager** window, go back to the TE Managed Primary Tunnels SR or the TE Unmanaged Primary Tunnels SR window as described in Edit Primary Tunnel, page 7-37.

Edit Primary Tunnel

Primary tunnel attributes can be modified in the primary tunnel editor.

There are two ways to access the primary tunnel editor:

- from the managed or unmanaged primary tunnels SR window or
- from the Service Requests window.

Access from Primary Tunnel SR Window

To access the primary tunnel editor from the primary tunnel SR window (TE Managed Primary Tunnels SR or TE Unmanaged Primary Tunnels SR window) and edit a managed or an unmanaged primary tunnel, use the following steps:

Step 1 Choose **Traffic Engineering**.

Step 2 Click **Create Managed TE Tunnel**. The TE Managed Primary Tunnels SR window in Figure 7-14 appears.

or

Click **Create Unmanaged TE Tunnel**. The TE Unmanaged Primary Tunnels SR window appears.

Step 3 To edit a tunnel SR, select the desired SR and click **Edit**.

The Edit TE Managed Primary Tunnel or the Edit TE Unmanaged Primary Tunnel window appears.

The primary tunnel editor is identical to that of the create primary tunnel GUI. For an explanation of the various window elements, see Create Primary Tunnel, page 7-32.

Step 4 Make the desired changes and click **OK** to accept, or **Cancel** to discard the changes.

In the TE Unmanaged Primary Tunnel SR window, the **Op** field changes to MODIFY.

Note The modified tunnel can be reverted to its original state by selecting it and clicking **Delete**. The MODIFY flag in the Op column disappears.

Step 5 Click **Save & Deploy** to either deploy the new tunnel SR to the network or force deploy all tunnels, or you can create or edit more primary tunnels and then save and deploy all changes.

The Service Requests window (**Operate > Service Request Manager**) appears and displays the state of the deployed SR.

For more information on working with service requests, see the managing service requests part elsewhere in this guide.

Access from Service Requests Window

To access the primary tunnel editor from the Service Requests window, assuming that the SR has been created, use the following steps:

Step 1 Choose **Operate > Service Request Manager**.

Step 2 To edit the desired tunnel SR, select the SR in question and click **Edit**.

Depending on whether a managed or an unmanaged tunnel has been selected, the TE Managed Primary Tunnel SR or the TE Unmanaged Primary Tunnel SR window appears displaying the SR selected in the Service Requests window.

Step 3 Select the tunnel SR and click **Edit**.

The Edit TE Unmanaged Primary Tunnel window appears.

Go to Access from Primary Tunnel SR Window, page 7-37 and continue the process from Step 4.

Delete Primary Tunnel

TE tunnels can be deleted either from the TE Links List window (see Deleting TE Tunnels, page 7-25) or in the primary or backup tunnels SR windows.

To delete a managed or an unmanaged primary tunnel from the TE Managed Primary Tunnels SR or TE Unmanaged Primary Tunnels SR window, use the following steps:

Step 1 Choose **Traffic Engineering**.

Step 2 Click **Create Managed TE Tunnel**. The TE Managed Primary Tunnels SR window appears.

or

Click **Create Unmanaged TE Tunnel**. The TE Unmanaged Primary Tunnels SR window appears.

Step 3 To delete a tunnel, select the desired tunnel(s) and click **Delete**.

The **Op** field status changes to **DELETE**.

For an explanation of the various window elements, see Create Primary Tunnel, page 7-32.

Note The deleted tunnel can be reverted to its original state by selecting it and clicking **Delete**. The DELETE flag in the Op column disappears.

Step 4 Click **Save & Deploy** to either deploy the new tunnel SR to the network or force deploy all tunnels, or you can create or edit more primary tunnels and then save and deploy all changes.

The Service Requests window (**Operate > Service Request Manager)** appears and displays the state of the deployed SR.

For more information on working with service requests, see the managing service requests part elsewhere in this guide.

Backup Tunnel Operations

Prime Fulfillment allows you to perform a number of backup tunnel operations, which are described in this section.

The *Cisco Prime Fulfillment Theory of Operations Guide 6.2* contains a section on Connectivity Protection (CSPF) Backup Tunnels, which is one of the techniques used to provide backup protection.

Create Backup Tunnel

Backup tunnels are created in much the same way as primary tunnels. In both cases, building an explicit path is not required when an existing path already traverses the desired routers. A path can be used for any number of tunnels within its bandwidth capacity.

A precondition for creating a backup tunnel is the presence of an explicit path. To create an explicit path, see Create Explicit Path, page 7-29.

To create a backup tunnel, use the following steps:

Step 1 Choose **Traffic Engineering > Create TE Backup Tunnel**.

The TE Protection SR window appears.

The TE Protection SR window includes the following elements:

The columns in the tunnel list provides the following information:

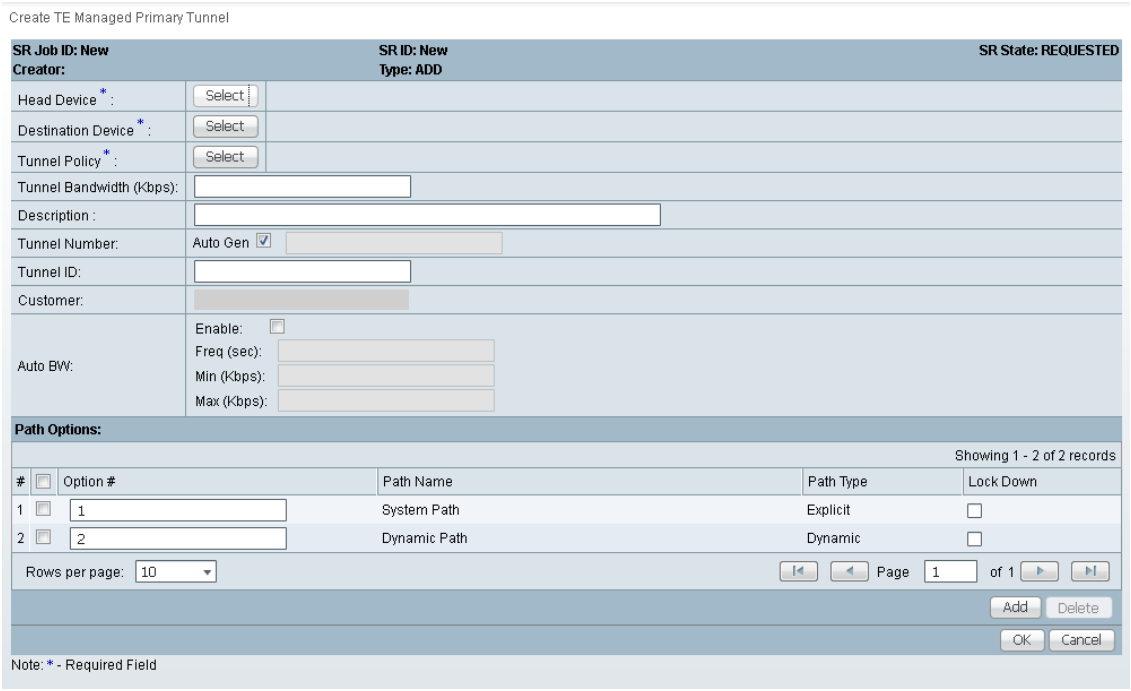
- **Op**—Current SR operation on the tunnel. This can be one of the following:
  - **ADD**—Indicates a newly added tunnel, either calculated by the system or entered by the user.
  - **MODIFY**—Indicates a modified existing tunnel.
  - **DELETE**—Indicates an existing tunnel to be deleted, either computed by the system or originated by the user.
- **Tunnel ID**—Unique tunnel identifier used within Prime Fulfillment.
- **T#**—Tunnel number on the head router.
- **Head**—Hostname of the head router.
- **Dest**—Hostname of the destination router.
- **BW Quota**—Amount of bandwidth that this backup tunnel can protect. The router limits the LSPs that can use this backup tunnel so that the sum of the bandwidth of the LSPs does not exceed the specified amount of bandwidth. If there are multiple backup tunnels, the router will use the best-fit algorithm.
- **Deploy Status**—Tunnel deployment status.
- **Conformance**—Indicates whether the tunnel is found to be conformant when running discovery. A tunnel is non-conformant if it has a non-zero bandwidth reservation and a zero hold or setup priority. If a tunnel is entered through TEM, it is always conformant. A connectivity protection tunnel is marked Conformant = true if it has zero tunnel bandwidth, unlimited backup bandwidth, and an 'exclude address' first path option. Otherwise, it is marked Conformant = false.
- **Backup Type**—Can be either bandwidth protected backup tunnels (**BW Protected**) or CSPF-routed backup tunnels (**CSPF**). For more information about these types of backup tunnels, see the *Cisco Prime Fulfillment Theory of Operations Guide 6.2*.
- **Head Region**—The region to which the head router belongs.

- **Tail Region**—The region to which the tail router belongs.

Step 2 Click **Create**.

The Create TE Backup Tunnel window in Figure 7-15 appears.

Figure 7-15 Create TE Backup Tunnel



The Create TE Backup Tunnel window includes the following elements:

- **Head Device**—Head device for the tunnel.
- **Destination Device**—Destination device for the tunnel. The selection window is very similar to the Head Device selection window.
- **Protected Interface(s)**—Interface(s) on the head router that this backup tunnel protects.
- **Description**—Descriptive text to help identify the tunnel.
- **Backup Bandwidth Limit**—Bandwidth protected by the backup tunnel.
  - **Any Pool BW**—Bandwidth set aside for the protection of either the Sub Pool or the Global Pool.
  - **Sub Pool (BC1) BW**—Bandwidth set aside for the Sub Pool.
  - **Global Pool (BC0) BW**—Bandwidth set aside for the Global Pool.

  For a definition of pool types, see the *Cisco Prime Fulfillment Theory of Operations Guide 6.2*.
- **Tunnel Number**—Tunnel number corresponding to the tunnel interface name.
  - **Auto Gen**—Check this box to generate the tunnel number at provisioning time. Otherwise, enter a desired number.

  Note If a manually entered tunnel number is too low, it could prevent deployment.
- **Tunnel ID—**Unique tunnel identifier used within Prime Fulfillment.
- **Tunnel Bandwidth**—Total allocated bandwidth of this backup tunnel (display only).
- **Tunnel Pool Type**—Tunnel bandwidth pool type for this policy (display only). For a definition of pool types, see the *Cisco Prime Fulfillment Theory of Operations Guide 6.2*.
  - **Global Pool (BC0)**—Bandwidth will be reserved from Global Pool.
  - **Sub Pool (BC1)**—Bandwidth will be reserved from Sub Pool.
- **Setup Priority (0-7), Hold Priority (0-7), Affinity, Affinity Mask**—All manually created backup tunnels should have setup and hold priorities of 0 and affinity value and mask of 0x0 for them to be able to protect an element.

Path options:

- **Option #**—Sequential number of available explicit paths.
- **Path Name**—Name of the explicit path.
- **Path Type**—Explicit path type (**Explicit** or **Dynamic**)
- **Lock Down**—Disables reoptimization check on the tunnel, if checked.

Step 3 Select, at a minimum, a **Head Device**, a **Destination Device**, and a **Protected Interface**.

Also, specify a **Backup Bandwidth Limit** greater than zero. Add other tunnel information as desired.

Step 4 Click **Add** to add just one path.

The Select TE Explicit Path window appears.

Step 5 Select an explicit path.

It must match the head and destination of an existing path. If none is available, you first must set one up. To do so, see Create Explicit Path, page 7-29.

Step 6 Click **Select**.

The selected path appears in the **Path Options** section of the page as shown in the Select TE Explicit Path window.

For explicit paths, you can click the pathname to open the Explicit Path Viewer.

Step 7 In the Create TE Backup Tunnel window, click **OK** to accept the entered tunnel information or click **Cancel** to quit the window without saving it.

In the TE Protection SR window, a new backup tunnel is added in the tunnel list with the **Op** field set to ADD.

Note The added tunnel can be reverted to its original state by selecting it and clicking **Delete**. The tunnel is removed from the tunnel list.

Step 8 Click **Save & Deploy** to either deploy the new tunnel SR to the network or force deploy all tunnels, or you can create or edit more backup tunnels and then save and deploy all changes.

The **Save & Deploy** button provides two options:

- **SR Tunnels Only**—Deploy all tunnel changes that does not impact tunnel placement, or if no changes were made to the SR, use this to redeploy the SR that was in **Requested** or **Invalid** state.
- **Force Deploy All Tunnels**—Force deployment of all tunnels in this SR. This could be useful when previous provisioning of the SR has failed, so that it is necessary to force through the deployment of all tunnels in the SR.

When you click **Save & Deploy**, Prime Fulfillment locks the TE routers effected, which will block any subsequent SRs which use that TE router until the SRs are finished. It is safe to try and deploy other SRs in the system. If there is any conflict with the SR currently being processed, Prime Fulfillment will simply ask you to wait until it is complete. To see the state of deployment, go to the Service Requests window under Inventory and Connection Manager or open the Task Manager under Monitoring.

Note You might see Elixir Warnings during TE Tunnel deployment. The deployment will be successful and the warning messages can safely be ignored.

Note With the exception of TE Traffic Admission SRs, TE SRs are always deployed immediately from the specific TE SR window, not from the **Operate > Service Request Manager** page.

The Service Requests window (**Operate > Service Request Manager)** appears and displays the state of the deployed SR.

For more information on working with service requests, see the managing service requests part elsewhere in this guide.

If the SR does not go to the **Deployed** state, go to the Task Logs window to see the deployment log (**Operate > Task Manager > Logs**) as described in SR Deployment Logs, page 10-46.

Edit Backup Tunnel

Backup tunnel attributes can be modified in the backup tunnel editor.

There are two ways to access the backup tunnel editor:

- from the Protection SR window or
- from the Service Requests window.

From the Protection SR Window

To access the Protection SR window to edit a backup tunnel, use the following steps:

Step 1 Choose **Traffic Engineering > Create TE Backup Tunnel**.

The TE Protection SR window appears.

Step 2 To edit a tunnel SR, select the desired SR and click **Edit**.

The Edit TE Backup Tunnel window appears. The backup tunnel editor is identical to that of the create backup tunnel GUI. For an explanation of the various window elements, see Create Backup Tunnel, page 7-39.

Step 3 Make the desired changes and click **OK**.

In the TE Protection window, the **Op** field changes to MODIFY.

Note The modified tunnel can be reverted to its original state by selecting it and clicking **Delete**. The MODIFY flag in the Op column disappears.

Step 4 In the TE Protection SR window, click **Save & Deploy** to either deploy the new tunnel SR to the network or force deploy all tunnels, or you can create or edit more backup tunnels and then save and deploy all changes.

The Service Requests window (**Operate > Service Request Manager**) appears and displays the state of the deployed SR.

For more information on working with service requests, see the managing service requests part elsewhere in this guide.

From the Service Requests Window

To edit a backup tunnel from the **Service Requests** window, assuming that the SR has been created use the following steps:

Step 1 Choose **Operate > Service Request Manager**.

Step 2 To edit the desired tunnel SR, select the SR in question and click **Edit**.

The TE Protection SR window appears displaying the SR selected in the Service Request Manager window.

Step 3 Select the tunnel SR and click **Edit**.

The Edit TE Backup Tunnel window appears.

Go to Edit Backup Tunnel, page 7-42 and continue the process from Step 3.

Delete Backup Tunnel

TE tunnels can be deleted either from the TE Links List window (see Deleting TE Tunnels, page 7-25) or in the primary or backup tunnels SR windows.

To delete a backup tunnel from the TE Protection SR window, use the following steps:

Step 1 Choose **Traffic Engineering > Create TE Backup Tunnel**.

The TE Protection SR window appears.

Step 2 To delete a tunnel SR, select the desired SR and click **Delete**.

The **Op** field status changes to **DELETE** for unmanaged tunnels.

For an explanation of the various window elements, see Create Backup Tunnel, page 7-39.

Note The deleted tunnel can be reverted to its original state by selecting it and clicking **Delete**. The DELETE flag in the Op column disappears.

Click **Save & Deploy** to either deploy the new tunnel SR to the network or force deploy all tunnels, or you can create or edit more primary tunnels and then save and deploy all changes.

The Service Requests window (**Operate > Service Request Manager**) appears and displays the state of the deployed SR.

For more information on working with service requests, see the managing service requests part elsewhere in this guide.

Purging a Service Request

The Purge operation in the Service Request Manager window is designed to remove a service request from the repository without affecting the network.

The **Purge** button has 2 options:

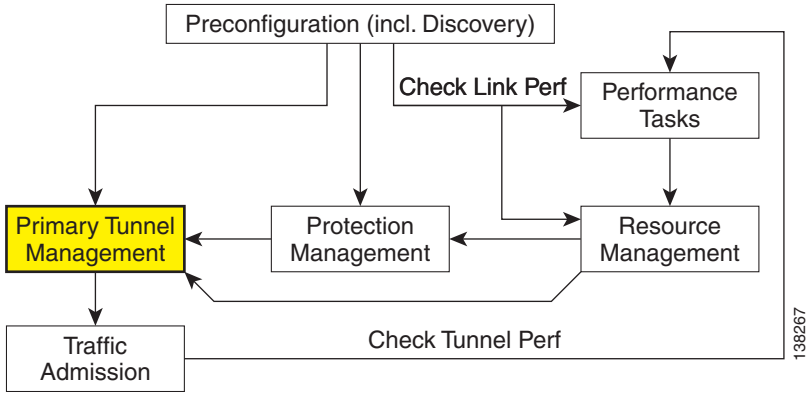
- **Purge**—The regular purge can only be used on the service request in **CLOSED** state. Therefore, it cannot be used on TE Resource, TE Tunnel, or TE Protection service requests because these cannot be decommissioned. These three types of service requests can only be force purged.
- **Force Purge**—During force purge, the repository checks the necessary dependency on the service request before it can be purged, so if a service request cannot be purged, there will be an error message.

Advanced Primary Tunnel Management

In addition to the basic tunnel management tools described in Basic Tunnel Management, page 7-27, Prime Fulfillment gives access to a set of advanced tunnel planning tools that provide optimal placement of tunnels to ensure efficient use of network resources.

The highlighted box in Figure 7-3 shows where in Prime Fulfillment primary tunnel management occurs.

Figure 7-16 Prime Fulfillment Process Diagram - Primary Tunnel Management



The advanced tools are available for managed tunnels only. The difference between managed and unmanaged tunnels is described in the section Managed/Unmanaged Primary Tunnels in the *Cisco Prime Fulfillment Theory of Operations Guide 6.2*.

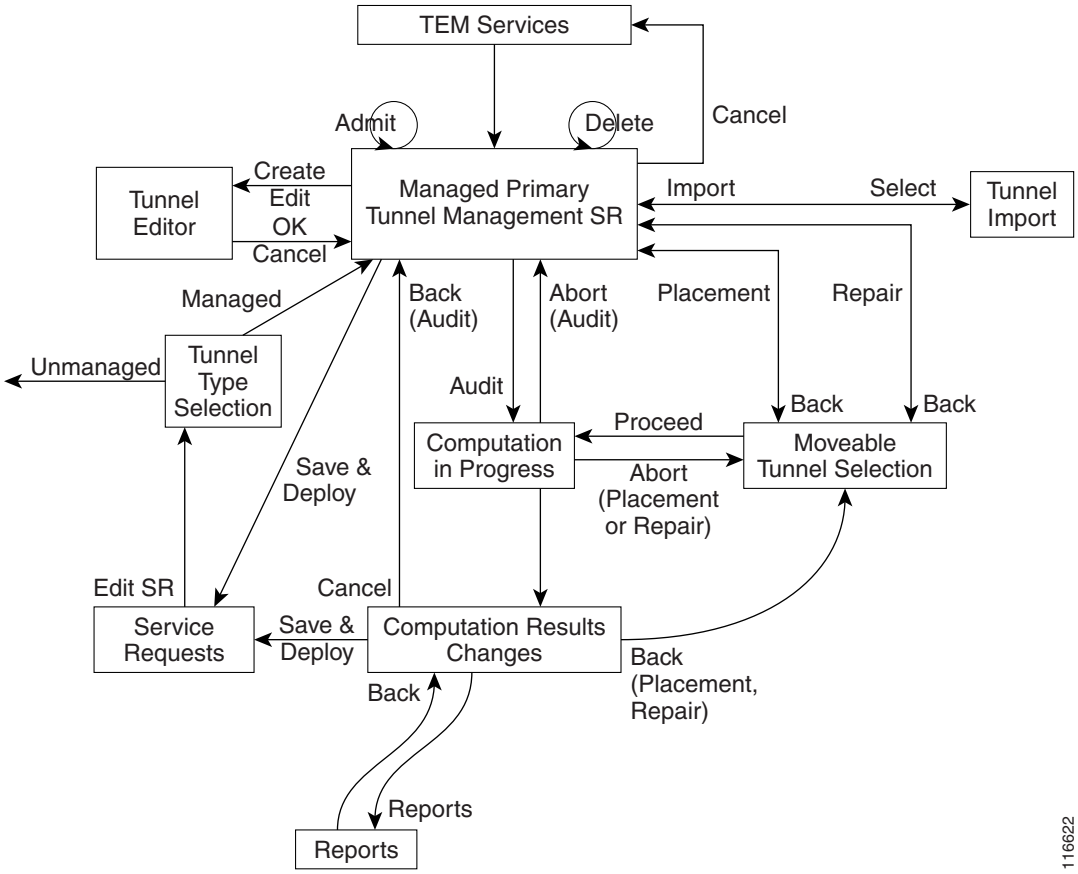
This section includes the following:

Tunnel Operations

This section explains the advanced tunnel operations in Prime Fulfillment that incorporate the planning tools.

An overview of the primary tunnel management process is provided in Figure 7-17.

Figure 7-17 Primary Tunnel Management Processes



For **Tunnel Type Selection**, when you select **Unmanaged** the TE Unmanaged Primary Tunnel SR window appears (see Basic Tunnel Management, page 7-27).

All other elements in Figure 7-17 are described in this section.

Create Primary Tunnel

To create a TE managed primary tunnel with the RG license installed, use the following steps:

Step 1 Choose **Traffic Engineering**.

Step 2 Click **Create Managed TE Tunnel**.

The TE Managed Primary Tunnels SR window appears.

For an explanation of the various window elements, see Create Primary Tunnel, page 7-32.

Step 3 Click **Create**.

The Create TE Managed Primary Tunnel window appears.

For an explanation of the various window elements, see Create Primary Tunnel, page 7-32.

The **Path Options** section provides three path types, **System Path**, **Explicit Path**, and **Dynamic Path**.

A **System Path** is an Prime Fulfillment system generated explicit path (immovable). The first path has to be an explicit path.

An **Explicit Path** is a fixed path from a specific head to a specific destination device.

A **Dynamic Path** is provisioned by allowing the head router to find a path. The **dynamic** keyword is provisioned to the routers.

Step 4 To select a **Head Device**, click the corresponding **Select** button to open the device selection window.

Select a head device and click **Select**.

Step 5 To select a **Destination Device**, click the corresponding **Select** button to open the device selection window.

Select a tail device and click **Select**.

Step 6 To select a **Tunnel Policy**, click the corresponding **Select** button to open the policy selection window.

Note If no tunnel policies are available, the reason could be that they are all unmanaged. To create a managed tunnel, first create a managed policy in **Service Design > Policy Manager** (see Create Policy, page 7-71) by making sure to check the **Managed** check box.

The Select Managed TE Tunnel Policy window includes the following elements:

- **Policy Name**—Name of the TE policy.
- **Pool Type**—Tunnel bandwidth pool type for this policy. For a definition of pool types, see the Bandwidth Pools section in the *Cisco Prime Fulfillment Theory of Operations Guide 6.2*.
  - **SUB_POOL**—Bandwidth will be reserved from Sub Pool.
  - **GLOBAL**—Bandwidth will be reserved from Global Pool.
- **Setup Priority**—Priority used when signaling an LSP for the tunnel to determine, which of the existing tunnels can be preempted. Valid values are from 0 to 7, where a lower number indicates a higher priority. Therefore, an LSP with a setup priority of 0 can preempt any LSP with a non-0 hold priority.
- **Hold Priority**—Priority associated with an LSP for the tunnel to determine if it should be preempted by other LSPs that are being signaled. Valid values are from 0 to 7, where a lower number indicates a higher priority.
- **Affinity**—Attribute values required for links carrying the tunnel (bit values are either 0 or 1).
- **Affinity Mask**—Attribute values to be checked. If a bit in the mask is 0, a link's attribute value of that bit is irrelevant. If a bit in the mask is 1, the link's attribute value and the tunnel's required affinity for that bit must match.
- **Delayed Constraint**—True or false value. If true, the tunnel has a maximum delay that its path must not exceed.
- **FRR Protection**—Used to enable an MPLS traffic engineering tunnel to use a backup tunnel in the event of a link failure if a backup tunnel exists.
  - **None**—No backup tunnel needed.
  - **Best Effort**—Use backup tunnel if available.
  - **Link and SRLG**—Specifies that primary tunnels should be routed only through links and SRLGs that are protected by FRR backup tunnels.
  - **Link, SRLG and Node**—Specifies that primary tunnels should be routed only through links, SRLGs and nodes that are protected by FRR backup tunnels.

- **MPLS IP Enabled**—Specifies whether MPLS IP has been configured for the corresponding tunnel.

Step 7 Specify a tunnel bandwidth greater than zero.

Step 8 Add other tunnel information as desired.

Step 9 Optionally, if you want to specify an explicit path rather than using the system path provided by Prime Fulfillment, delete the system path and subsequently add the explicit path.

For a more detailed explanation of this step, see Create Primary Tunnel, page 7-32.

Step 10 In the Create TE Managed Tunnel window, click **OK** to accept the entered tunnel information or **Cancel** to quit and return to the TE Managed Primary Tunnels SR window.

The TE Managed Primary Tunnel SR window appears displaying the new tunnel with the **Op** field set to ADD to signify that an SR has been added.

Note The added tunnel can be reverted to its original state by selecting it and clicking **Delete**. The tunnel is removed from the tunnel list.

Step 11 In the TE Managed Primary Tunnel SR window, you can create or edit more tunnels, or if you are done with all the changes, proceed in one of the following two ways depending on which of the following buttons are active (**Save & Deploy** is not available after the **Create** operation):

- **Proceed with Changes**: The changes you entered impacts tunnel placement. Click on this to continue with one of the planning flows described in the Placement Tools (see Placement Tools, page 7-52) until the SR can be saved and deployed.
- **Save & Deploy**: The changes you entered do not impact tunnel placement. Click on this to save and deploy the SR. This function is further described in Create Primary Tunnel, page 7-32.

When you click **Save & Deploy**, Prime Fulfillment locks the TE routers effected, which will block any subsequent SRs which use that TE router until the SRs are finished. It is safe to try and deploy other SRs in the system. If there is any conflict with the SR currently being processed, Prime Fulfillment will simply ask you to wait until it is complete. To see the state of deployment, go to the Service Requests window under Inventory and Connection Manager or open the Task Manager under Monitoring.

Note With the exception of TE Traffic Admission SRs, TE SRs are always deployed immediately from the specific TE SR window, not from the Service Requests page in **Inventory and Connection Manager**.

If **Save & Deploy** was selected in Step 11, the Service Requests window (**Operate > Service Request Manager)** opens and displays the state of the deployed SR.

For more information on working with service requests, see the managing service requests part elsewhere in this guide.

Note You might see Elixir Warnings during TE Tunnel deployment. The deployment will be successful and the warning messages can safely be ignored.

If the SR does not go to the **Deployed** state, go to the Task Logs window to see the deployment log (**Operate > Task Manager > Logs**) as described in Task Logs, page 10-26.

Edit Primary Tunnel

The only difference between creating and editing tunnels is that in the tunnel editor, the head and destination devices and tunnel number fields are not editable. Otherwise, you create and edit the same attributes.

Only **Proceed with Changes** or **Save & Deploy**, not both, are available depending on whether the changes you entered impacts tunnel placement.

To edit a primary tunnel, see Edit Primary Tunnel, page 7-37

Delete Primary Tunnel

To delete one or more tunnels, see Delete Primary Tunnel, page 7-38.

Admit Primary Tunnel

The Admit function is used to admit selected tunnels not previously verified into the managed topology. This feature is used only for discovered tunnels that failed verification. During the discovery process, verification is performed with the Tunnel Placement algorithm, as if the tunnels were admitted for the first time.

Verification means that the discovered managed tunnel is verified against the network topology and TEM checks if there is enough bandwidth along the tunnel path (both are specified in the tunnel).

In general, verification will fail if there is not enough bandwidth due to the existence of other tunnels or a limitation on link capacity/bandwidth.

More specifically, this can happen when a priority 0 tunnel is created independently of TEM and a TE Discovery task is run. If the tunnel does not satisfy all the managed tunnel constraints (that is, if it is reserving more bandwidth than is available in a link that it passes through) TE discovery will mark it as ‘verified = false’. It will not be managed by TEM until you use the Admit button to make it verified. Typically this would have to be accompanied with some other tunnel or resource change to ensure that the constraint is now satisfied.

To admit a primary tunnel, use the following steps:

Step 1 In the **TE Managed Primary Tunnel SR**, select one or more unverified tunnels to migrate.

Step 2 Click **Admit**.

The unverified tunnel(s) are verified and, if sucessful, an **ADMIT** flag will appear in the **Op** column.

Step 3 Choose **Proceed with Changes** > **Tunnel Placement** to determine if the tunnels can be placed. If not, edit the tunnels and try again.

Import Primary Tunnel

This feature allows you to update tunnels in bulk through a file-based import mechanism. The data is migrated into the managed primary tunnel service request.

Construct XML Import File

To import tunnels from a file, first construct an XML import file conforming to the structure defined in the system supplied Document Type Definition (DTD) file (see Document Type Definition (DTD) File, page 7-108), and save the XML file together with the DTD file on the Prime Fulfillment server under the same directory. To create a valid import file, use the provided command line validation tool (see Command Line Validation Tool, page 7-50).

The following files are necessary for importing data into the Prime Fulfillment application and are included in the installation:

- DTD file for the import file in
  < *installedDir*>/ *resources/java/xml/com/cisco/vpnsc/ui/te*
  - **TeImport.dtd**

    (a sample file, '**sample.xml**', is also included)
- Shell script for executing the command line validator in the <*installedDir*>*/bin* directory.
  - **ImportTeTunnels**

    Usage: **importTeTunnels <***importfile***>**

importfile is a XML file and must specify **TeImport.dtd** as its DTD. **TeImport.dtd** must be in the same directory as *importfile*.

Command Line Validation Tool

The purpose of a command line validator is to help construct a valid import file off-line that corresponds to **TeImport.dtd**. The tool helps screen out errors associated with files that are not well-formed and files that do not conform to the rules set by the DTD.

For instructions on how to use the DTD file, see the DTD file documentation.

The tool reads the import file line-by-line, echoes each line in on the output as it parses, and reports any parsing error it encounters. The parsing and validation continues even when parsing errors are encountered for as long as the file structure makes sense.

Note This tool does not check for cross field validation or data integrity errors with respect to the Prime Fulfillment application.

Import Procedure

The file-based import feature is only enabled when there are no uncommitted new, changed, or deleted tunnels in the service request.

It provides a way of adding, editing, deleting, or migrating many tunnels at a time.

To start the import procedure, use the following steps:

Step 1 Prepare the XML import file in accordance with the DTD file.

Step 2 Go to **Traffic Engineering**.

Step 3 Select provider if this has not been done earlier in the session.

Step 4 Click **Create Managed TE Tunnel**.

The TE Managed Primary Tunnels SR window appears.

Step 5 Click **Import** to start the import process.

The Select Import File window appears.

Note The Import button is only enabled when there are no uncommitted new, changed, or deleted tunnels in the service request.

The Select Import File window lists all the XML files and any directories under the directory name shown in the **Look in** field.

The default directory shown in the **Look in** field corresponds to the installation directory in which the DTD and sample XML files reside.

Step 6 Select the desired XML file to be used for the import operation.

The system then parses the file. If any error is detected, it will be reported in the Tunnel Import Error Status window.

The Tunnel Import Error Status window shows the URL of the file, its last modified timestamp, the import status, and any error/warning messages.

Step 7 If the import operation failed, click **Cancel** to return to the previous window.

If it is partially successful, the **Continue** button is enabled, thereby providing an additional option to accept system treatment for errors/warnings and continue with the import operation.

Step 8 If the file is parsed successfully or you click **Continue**, all valid tunnels in the file are added to the service request and the TE Managed Primary Tunnels SR window is re-displayed in the SR view. The imported tunnels are displayed with the appropriate tunnel **Op** type.

Planning Strategy

The main objective of using the planning tools is to achieve optimal overall network utilization while causing minimal impact on any existing traffic on the network.

In most cases, the following strategy can be applied:

- Attempt to admit the new traffic optimizing on utilization (Placement feature) without allowing existing traffic to be moved. This offers the possibility of accommodating the new traffic without any changes to the existing traffic, while still optimising reserved bandwidth utilization under the constraint that existing tunnels do not move.
- If this fails, attempt to admit the same new traffic minimizing change to existing traffic (Repair feature) to see if the new traffic can be accommodated without affecting any more existing tunnels than necessary.
- If this succeeds in placing the new traffic, but you feel that the overall reserved bandwidth utilization is higher than would be preferred, consider grooming the network.
- If the Repair fails, review the parameters that control how many changes can be considered. Alternatively the specification to the desired traffic could be changed, or resource modifications could be made.

This strategy reflects the different approaches taken by the different algorithms in searching for solutions. However, other combinations are possible.

Placement Tools

Planning tools for primary tunnels are available from the **Proceed with Changes** and **Placement Tools** buttons in the TE Primary Tunnel SR window depending on whether a change has been made to the managed primary tunnels.

- Proceed with Changes: Used when you have made changes (add/change/delete/admit) to the tunnels. Tunnel operations are described in Tunnel Operations, page 7-45. Then choose one of the placement tools to verify primary placement with the system and continue with deployment. This button is also available in Resource Management.
- Placement Tools: Used to perform planning functions on the existing network.
  - The Tunnel Audit option should be used to verify the constraint-based placement of existing managed primary tunnels with the existing network topology. You can use this option to find out the optimality of your primary placement. If you are requiring protection levels above "Best Effort" on your primary tunnels, it is also important to perform an audit after any changes have been made in the protection network. If the audit results in warnings/violations, you can use the Tunnel Repair option help you find a solution.
  - The Groom option is used for optimizing your primary placement. In all primary computation, a quality report is produced which displays the optimality and utilization of the bandwidth pools. You can perform a Tunnel Audit first to determine if grooming is needed on your network.

The planning tools are described in detail in the following sections.

Note If tunnel attributes that are not supported by the placement tools (such as auto-bw frequency) are changed in conjunction with attributes that are supported, the attributes appear correctly in the TE Computation Results window. But if only unsupported attributes are changed, the TE Computation Results window still shows no achieved changes and the **Save & Deploy** button is grayed out so the change cannot be deployed.

Tunnel Audit

When any type of change is required, whether tunnel modifications or TE resource modifications, a Tunnel Audit is run to determine what inconsistencies the change might cause, if any. Tunnel Audit can also be used anytime to check for the optimality of network utilization.

The audit can be performed from the primary tunnel window or from the TE Links List window. (See TE Resource Management, page 7-20.)

To perform an audit on the created tunnel, use the following steps:

Step 1 Choose **Traffic Engineering**.

Step 2 Click **Create Managed Tunnel**.

The TE Managed Primary Tunnels SR window appears.

Tunnel Audit can be used in two ways:

- When one or more tunnels have been created or their attributes altered (see Create Primary Tunnel, page 7-46), Tunnel Audit can be activated by selecting **Proceed with Changes** .
- When no changes have taken place, Tunnel Audit can be accessed by selecting **Placement Tools**.

As an example, assume that a new primary tunnel SR has been created.

The TE Managed Primary Tunnel SR window appears.

Step 3 Choose **Proceed with Changes > Tunnel Audit**.

The Computation In Progress window appears temporarily. Then the TE Primary Tunnel Computation Results - Changes window appears.

This window includes the following elements:

Status section (top):

- **Computation Status**—Indicates whether the computation succeeded or failed.
- **Tunnels:**
  - **unplaced**—Number of unplaced tunnels out of the total.
  - **moved**—Number of tunnels that were moved.
- **Bandwidth - unplaced**—Amount of tunnel bandwidth that was not placed out of the total bandwidth of all existing and new tunnels.
- **Global Util.**—Global Pool bandwidth utilization percentage.

  The utilization values can be the following:
  - **Global Pool**—Comparison data for various Global Pool attributes.
  - **Sub Pool**—Comparison data for various Sub Pool attributes.
  - **Median**—Utilization of the link that is the middle link when all links are ordered by utilization.
  - **Max. Modifiable**—Utilization value for the most utilized link that has movable tunnels passing through it.
  - **Mean**—Average link utilization for the network as a whole.
  - **Max.**—Utilization value for the most utilized link in the topology.
- **Sub Pool Util.**—Sub Pool bandwidth utilization percentage.
- **Solution**—Utilization for the generated solution.
- **Original**—Utilizations for the original placement.

Changes section (left):

- **Changes**—Number of changes achieved out of the total number of changes.
  - **Achieved**—Indicates whether a specific change is successful (**Yes** or **No**).
  - **Origin**—The originator of the change. Can be **user** (change by user) or **compute** (from a computation, e.g. rerouting of a tunnel).
  - **Type**—The type of change requested: **Tunnel Add Change**, **Tunnel Modify Change**, **Tunnel Remove Change**, or **Element Modify Change**.
  - **Object ID**—A tunnel or link ID.

Note Certain attributes, such as Description, that do not impact the computation carried out by the placement tools and updates to these are not displayed in the computation results window.

Step 4 To obtain detailed information about the tunnel and whether the change request was achieved, select the specific tunnel and click **Details**.

A **qualityReport** is always generated. If the computation was successful, this will be the only report.

If a warning or a violation was encountered, one or more warning or violation reports will also be generated.

Step 5 To view an audit report, click **View Report** .

In some cases, both a **qualityReport** and a violation report is generated.

Step 6 To view the contents of the **qualityReport**, select the **qualityReport** and click the **Details** button.

The qualityReport fields in the right window pane include the following elements:

Status section (top): described above.

Report section (left):

- **Report Type**—There are three basic report types: a **qualityReport** (generated every time), warning reports, and violation reports.
- **Summary Info**—Summary information about the findings of the report.

Information section (right):

- **Report Type**—See description above.
- **Description**—Specific information about the report.
- **Achievement**—Success or failure of the computation attempt/solution (**SUCCESS** or **CONSTRAINT_VIOLATIONS_REPORTED**).
- **Solution**—Indicates whether a solution was found (**SOLUTION_FOUND**, **PARTIAL_SOLUTION_FOUND** or **NO_SOLUTION_FOUND**).
- **Termination**—Indicates whether the computation was completed:
  - **COMPLETED**—The computation completed processing before the time limit.
  - **TIMED_OUT**—The computation was not able to complete processing within the time limit. The solution presented is the best solution it was able to find in the time available.

- **Optimality**—Indicates whether the computation was optimal:
  - **OPTIMAL_FOR_ALL_CRITERIA**—The solution generated has proven to be the best for all optimization criteria.
  - **NO_OPTIMALITY_PROOF**—The solution's optimality is unknown.
  - **OPTIMAL_FOR_DEMAND_SELECTION**—The solution generated has proven to be the best in terms of total bandwidth placed, but utilization optimality is unknown.

OPTIMAL_FOR_SUB_POOL_PATH_SELECTION—The solution generated has proven to be the best in terms of total bandwidth placed and maximum sub pool utilization, but has not proven to be optimal in terms of global pool utilization.

Step 7 To view the contents of the violation report, select the violation report and click the **Details** button.

The TE Primary Tunnel Computation Results - Report (Details) window appears.

The report fields in the right window pane are described for each report in Warnings and Violations, page 7-98

Step 8 Click **View Result** to return to the Changes window.

If the proposed changes were achieved, you can click on **Save & Deploy** to save the achievable changes to the repository and implement the tunnel modifications on the network.

Note **Save & Deploy** will discard any changes that were not achievable.

The Service Requests window (**Operate > Service Request Manager**) appears and displays the state of the deployed SR.

For more information on working with service requests, see the managing service requests part elsewhere in this guide.

Tunnel Placement

The Placement feature supports the admission of new tunnels into the network and the modification of tunnels already admitted into the network. Prime Fulfillment will attempt to implement the changes in such a way that network utilization is optimized.

To place a created tunnel, use the following steps:

Step 1 Choose **Traffic Engineering**.

Step 2 Click **Create Managed TE Tunnel**.

The TE Managed Primary Tunnels SR window appears.

Step 3 When one or more tunnels have been created or their attributes altered (see Create Primary Tunnel, page 7-46), select **Proceed with Changes > Tunnel Placement**.

The Movable Tunnel Selection (Placement) window appears.

Step 4 Set the movable and unmovable managed tunnels.

You can specify whether, when admitting a new tunnel, existing tunnels can be moved (rerouted). This is configurable by you. The default is that managed tunnels are not movable.

Step 5 Click **Proceed**.

The Computation In Progress window shown appears temporarily. Then the TE Primary Tunnel Computation Results - Changes window appears.

Note Certain attributes, such as Description, that do not impact the computation carried out by the placement tools and updates to these are not displayed in the computation results window.

Step 6 To obtain detailed information about the tunnel and whether the placement request was achieved, select the specific tunnel and click **Detail**.

The detail section in the right side of the window appears.

If the placement request succeeded (**Achieved**: **yes**), the Detail pane will contain a computed **Path** that is selectable.

To view the path information, click the blue link in the computed **Path** field. The TE Explicit Path window appears.

Step 7 To view the placement report(s), click **View Report** in the Changes window.

The TE Primary Tunnel Computation Results - Report window appears.

A **qualityReport** is always generated. If the computation was successful, this will be the only report.

If a warning or a violation was encountered, one or more warning or violation reports will be generated as well.

Step 8 To view the contents of a placement report, select one of the reports and click the **Details** button.

In the case of a **qualityReport**, the TE Primary Tunnel Computation Results - Report (details) window appears in the report pane on the right.

Step 9 Click **View Result** to return to the Changes window and click **Save & Deploy** to save the change to the repository and implement the tunnel modifications on network.

The Service Requests window (**Operate > Service Request Manager)** appears and displays the state of the deployed SR.

For more information on working with service requests, see the managing service requests part elsewhere in this guide.

Tunnel Repair

As changes are made to the bandwidth requirements or delay parameters of existing tunnels, inconsistencies can arise with the Tunnel Placement. You can run a Tunnel Repair to address such inconsistencies. The objective of Tunnel Repair is to try to move as few existing tunnels as possible to accommodate the changes.

The repair operation can be performed from the primary tunnel window or from the TE Links List window. (See TE Resource Management, page 7-20.)

In the following, we will seek to repair an edited tunnel:

Step 1 Choose **Traffic Engineering > Create Managed Tunnel**.

The TE Managed Primary Tunnels SR window appears.

Tunnel Repair can be used in two ways:

- When one or more tunnels have been created or their attributes altered (see Create Primary Tunnel, page 7-46), Tunnel Repair can be activated by selecting **Proceed with Changes > Tunnel Repair**.
- When no changes have taken place, Tunnel Repair can be accessed by selecting **Placement Tools > Tunnel Repair**.

Step 2 In this example, a new primary tunnel SR has been created.

Run Tunnel Repair on the modified tunnels from the TE Managed Primary Tunnels SR window by navigating

Proceed with Changes > Tunnel Repair

The Movable Tunnel Selection window appears.

Step 3 Set the tunnels that should be movable.

Tunnel Repair will only move existing tunnels if it has to. If you do not want certain tunnels to be moved during Tunnel Repair, these tunnels should be explicitly excluded from the selected list of movable tunnels.

You can also specify a limit on the maximum number of tunnel moves that are acceptable using the **Maximum number of tunnel moves** field.

Note It is not necessary to set modified tunnels to be movable as these are movable by default.

Step 4 Click **Proceed**.

The Computation In Progress window shown appears temporarily. Then the TE Primary Tunnel Computation Results - Changes window s appears.

Note Certain attributes, such as Description, that do not impact the computation carried out by the placement tools and updates to these are not displayed in the computation results window.

Step 5 To obtain detailed information about the tunnel and whether the change request was achieved, select the specific tunnel and click **Detail**.

The detail section in the right side of the window appears.

Step 6 To view a repair report, click **View Report**.

The TE Primary Tunnel Computation Results - Report window appears.

A **qualityReport** is always generated. If the computation was successful, this will be the only report.

If a warning or a violation was encountered, one or more warning or violation reports will also be generated.

Step 7 To view the contents of the repair report, click the **Details** button.

In the case of a **qualityReport**, the TE Primary Tunnel Computation Results - Report (details) window appears.

The report fields in the right window pane are described for each report in Warnings and Violations, page 7-98

Step 8 Click **View Result** to return to the Changes window and click **Save & Deploy** to save the change to the repository and implement the tunnel modifications on network.

The Service Requests window (**Operate > Service Request Manager)** appears and displays the state of the deployed SR.

For more information on working with service requests, see the managing service requests part elsewhere in this guide.

Grooming

The purpose of grooming is to analyze the tunnel pathing with respect to the network elements and optimize resource allocation.

Grooming is not available when change requests have been created. In that case, only the placement tools under **Proceed with Changes** are available.

To perform grooming on the network, use the following steps:

Step 1 Choose **Traffic Engineering > Create Managed TE Tunnel**.

The TE Managed Primary Tunnels SR window appears.

Step 2 Run Grooming by navigating

Placement Tools > Groom

The Movable Tunnel Selection window appears.

Step 3 Set the tunnels that should be movable.

As with Tunnel Repair, Grooming will only move existing tunnels if it has to. If you do not want certain tunnels to be moved during the Grooming process, these tunnels should be explicitly excluded from the selected list of movable tunnels.

Step 4 Click **Proceed**.

The Computation In Progress window shown appears temporarily. Then the TE Primary Tunnel Computation Results - Changes window appears.

Note Certain attributes, such as Description, that do not impact the computation carried out by the placement tools and updates to these are not displayed in the computation results window.

Step 5 To obtain detailed information about the Grooming and whether it succeeded, select the specific tunnel and click **Detail**.

The detail section in the right side of the window appears.

Step 6 To view a Grooming report, click **View Report** .

The TE Primary Tunnel Computation Results - Report window appears.

A **qualityReport** is always generated. If the computation was successful, this will be the only report.

If a warning or a violation was encountered, one or more warning or violation reports will also be generated.

Step 7 To view the contents of the Grooming report, click the **Details** button.

In the case of a **qualityReport**, the TE Primary Tunnel Computation Results - Report (details) window appears.

The report fields in the right window pane are described for each report in Warnings and Violations, page 7-98

Step 8 Click **View Result** to return to the Changes window and click **Save & Deploy** to save the change to the repository and implement the tunnel modifications on the network.

The Service Requests window (**Operate > Service Request Manager**) appears and displays the state of the deployed SR.

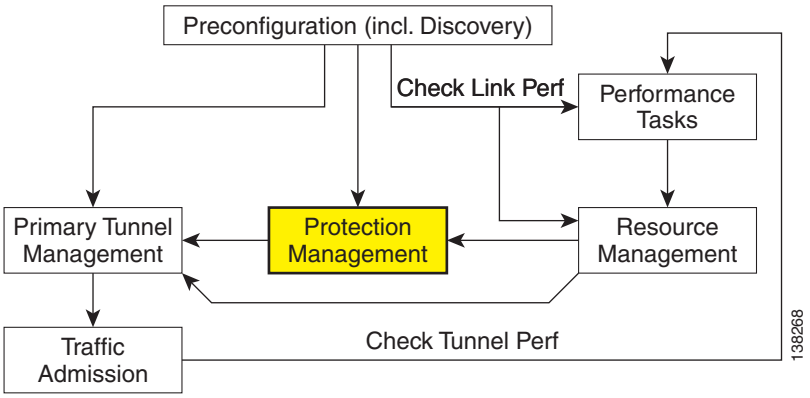
For more information on working with service requests, see the managing service requests part elsewhere in this guide.

Protection Planning

This section describes the process of creating and managing the protection of network elements using automated protection tools. See Basic Tunnel Management, page 7-27 for a description of the process using the basic tools.

The highlighted box in Figure 7-18 shows where in Prime Fulfillment protection management occurs.

Figure 7-18 Prime Fulfillment Process Diagram - Protection Management



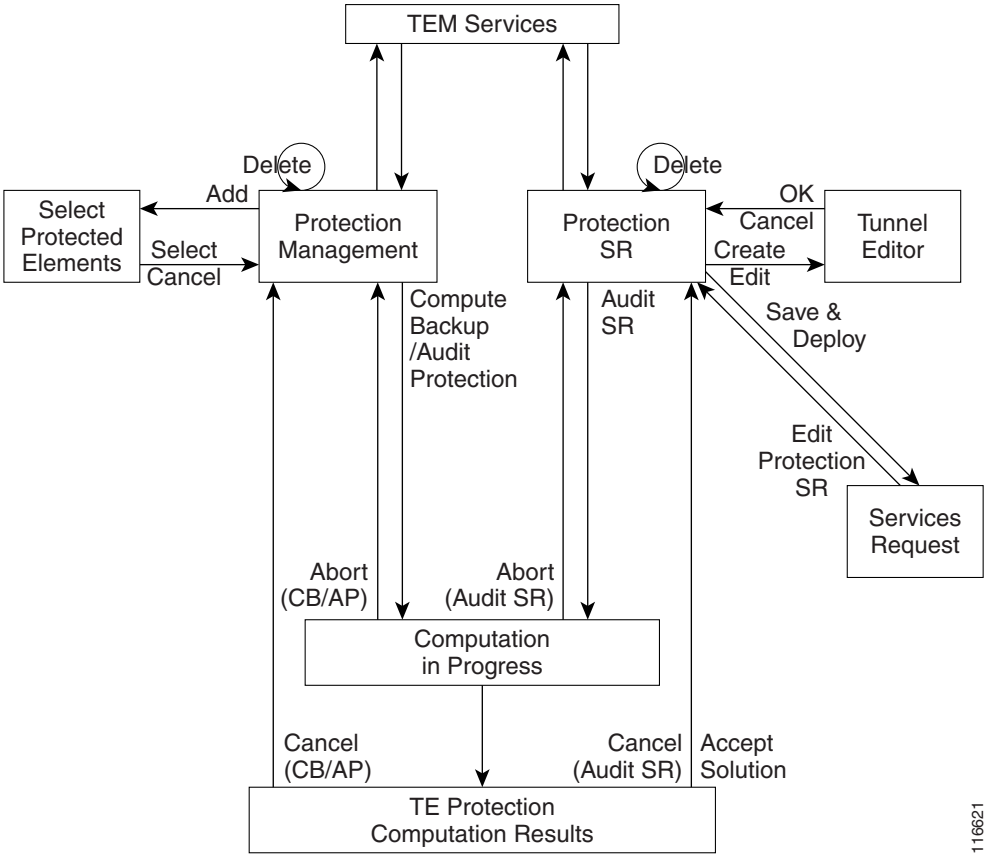
The purpose of protection planning is to protect selected elements in the network (links, routers, or SRLGs) against failure.

The first step is to identify the elements that must be protected and then invoke the protection tools to compute the protected tunnels. From the computation, the system responds for each element with either a set of tunnels that protect the element or a set of violations and warnings that help you determine why it could not be protected.

For successfully protected elements the tunnels can be deployed on the network. For elements that could not be protected, the protection is either ignored or the constraints are altered on the protection case. More specifically, this can involve changing the TE bandwidth settings of the links associated to the element and then rerunning the protection computation on the altered network.

An overview of the protection management processes is provided in Figure 7-19.

Figure 7-19 Protection Management Processes



This section includes the following:

- SRLG Operations, page 7-61
  - Create SRLG, page 7-61
  - Edit SRLG, page 7-61
  - Delete SRLG, page 7-62
- Configure Element Protection, page 7-62
- Protection Tools, page 7-62
  - Compute Backup, page 7-63
  - Audit Protection, page 7-64
  - Audit SR, page 7-65.

SRLG Operations

It is not uncommon for links to have identical physical characteristics, such as being physically located in the same conduit, or being connected to the same hardware. As a result, they could fail as a group during a single failure event. A Shared-Risk Link Group (SRLG) addresses this problem by identifying links that could fail together.

After SRLG modifications (create, edit, delete), use the protection planning functions in the **TE Protection Management** window to ensure that adequate protection is available on the network.

Create SRLG

Creating an SRLG is only necessary if a shared risk link group has been identified and it must be protected.

To create an SRLG, use the following steps:

Step 1 Choose **Traffic Engineering > SRLGs**.

The TE SRLG List window appears.

Step 2 To create an SRLG in the **TE SRLG List**, click **Create**.

The TE SRLG Editor window appears.

Step 3 Specify an **SRLG Name**.

Step 4 Click **Add Link**.

The Links associated with SRLG window appears.

Step 5 Select one or more links and click **Select**.

The corresponding link information is added to the link list and the Select window closes and returns to the SRLG editor.

Step 6 Click **Save** to save the SRLG.

This closes the SRLG editor and brings back the TE SRLG List as the active window, where the newly created SRLG is listed.

Edit SRLG

To edit an SRLG, use the following steps:

Step 1 Choose **Traffic Engineering > SRLGs**.

The TE SRLG List window appears.

Step 2 To edit an SRLG in the TE SRLG List, from the TE SRLG List window select the SRLG that you want to modify and click **Edit**.

The TE SRLG Editor window appears.

Step 3 Use **Add Link** and **Remove Link** to adjust to the desired set of links for the selected SRLG.

Step 4 Click **Save** to save the changes.

Delete SRLG

To delete an SRLG, use the following steps:

Step 1 Choose **Traffic Engineering > SRLGs**.

The TE SRLG List window appears.

Step 2 To delete an SRLG in the TE SRLG List, from the TE SRLG List window select the SRLG(s) that you want to delete and click **Delete**. The Delete Confirm window appears.

Step 3 Click **Delete** to confirm.

The Delete Confirm window closes. After the TE SRLG List window has been updated, the deleted SRLG no longer appears in the SRLG list.

Configure Element Protection

Before a protection computation can be performed, it is necessary to configure the network element protection.

To do so, use the following steps:

Step 1 Choose **Traffic Engineering > Protected Elements**.

The TE Protection Management window appears.

Explanation of the **Protection Status** field:

Protection Status—The protection status displayed is determined from the last time an audit was performed. The audit is performed either explicitly by the user or when the protection SR is deployed. The protection status is stated for each network element as either **Protected**, **Not Fully Protected**, or **Unknown**. Click on the column header, **Protected**, to sort elements according to protection status

Step 2 First, decide which network elements must be protected.

In the TE Protection Management window, click **Add** to add a protection element (link, node, or SRLG). The Select Protection Elements window appears.

Links that are connected to non-Cisco devices cannot be protected and will, therefore, not show in the Select protection elements window. Likewise, non-Cisco devices and SRLGs that contain links to non-Cisco devices cannot be protected and are excluded from the selection.

Step 3 Select one or more elements to be protected and click **Select.**

The Select Protection Element window closes and the TE Protection Management window reappears.

Next, decide which protection tools should be applied. These are described in Protection Tools, page 7-62.

Protection Tools

Relying on manual creation of backup tunnels as described in Basic Tunnel Management, page 7-27 has its limitations, not just for larger and more complicated networks.

The protection tools available in Prime Fulfillment provide a number of tools that automatically compute and verify protection of specified network elements.

Note Certain attributes, such as Description, that do not impact the computation carried out by these tools and updates to these are, therefore, not displayed in the computation results window.

Compute Backup

Compute Backup is used to let Prime Fulfillment automatically compute the necessary backup tunnels to protect specified network elements. The manual process is described in Basic Tunnel Management, page 7-27

To run Compute Backup, use the following steps:

Step 1 Choose **Traffic Engineering > Protected Elements**.

Step 2 Configure the necessary protection elements as described in Configure Element Protection, page 7-62.

Step 3 If you only want to perform Compute Backup on selected elements, select one or more elements on which to calculate a backup path.

Step 4 Click **Compute Backup** and select one of the following:

- All Elements
- Selected Elements

First the Computation In Progress window appears and then the TE Protection Computation Results window appears.

The **Element:** table displays the outcome of the computation for each element in the protection computation. The status for each element is indicated by at least one row per element in the table. If the status is not valid, the table will contain one row per warning or violation.

The **Element:** table contains the following columns:

- **Element Name**—Name of the network element to be protected.
- **Type**—Network element type (node, link, or SRLG).
- **Report**—Warning or violation associated with an element, if any, as reported by the computation engine.
- **Status**—Computation status of the network element:
  - Valid Tunnels—The element is fully protected by backup tunnels.
  - InvalidTunnels—An Audit Protection detected that the element was not fully protected by the existing backup tunnels.
  - No Solution Exists—A Compute Backup has proven that it is not possible to fully protect the element.

Note Certain attributes, such as Description, that do not impact the computation carried out by the protection tools and updates to these are not displayed in the computation results window.

Step 5 Select a row corresponding to a specific warning or violation and click **Detail** to display a detailed description in the right pane and backup tunnels associated with the selected item in the bottom pane.

For a description of warnings and violations, see Warnings and Violations, page 7-98

Explanation of the **Protection Type** column:

- **Protection Type**—Protection side-effect from activating the tunnel. There are three protection types:
  - **Protection tunnels**—Tunnels that can be activated to provide protection for a specified element.
  - **Side-effect tunnels**—Tunnels that are activated to protect a neighboring element, but which are also activated when a specified element fails.
  - **Activated tunnels**—Tunnels that are activated when a specified element fails, and which might or might not provide protection for the specified element or its neighbors.

The **Backup Tunnel** table displays which new protection tunnels are required and any existing tunnels that should be kept or deleted for each element.

Step 6 If the proposed protection solution is acceptable, click **Accept Solution**.

The TE Protection SR window appears with all tunnel additions and deletions computed by the system.

For an explanation of the various window elements, see Create Backup Tunnel, page 7-39.

Optionally, you can make tunnel changes here and then run **Audit SR** to ensure that you have the desired level of protection before you deploy (see Audit SR, page 7-65).

Step 7 Click **Save & Deploy** to deploy the new tunnel SR to the network.

When you click **Save & Deploy**, Prime Fulfillment locks the TE routers effected, which will block any subsequent SRs which use that TE router until the SRs are finished. It is safe to try and deploy other SRs in the system. If there is any conflict with the SR currently being processed, Prime Fulfillment will simply ask you to wait until it is complete. To see the state of deployment, go to the Service Requests window under Inventory and Connection Manager or open the Task Manager under Monitoring.

Note With the exception of TE Traffic Admission SRs, TE SRs are always deployed immediately from the specific TE SR window, not from the **Service Requests** page in **Inventory and Connection Manager**.

The Service Requests window (**Operate > Service Request Manager**) opens and displays the state of the deployed SR.

For more information on working with service requests, see the managing service requests part elsewhere in this guide.

If the SR does not go to the **Deployed** state, go to the Task Logs window to see the deployment log (**Monitoring > Task Manager > Logs**) as described in SR Deployment Logs, page 10-46.

Audit Protection

As opposed to the Compute Backup tool described on page 63, Audit Protection does not attempt to create a backup solution. It seeks to verify protection of specified network elements with the current set of backup tunnels and reports any warnings or violations that are discovered. It is recommended that any time a change has been committed on the TE topology such as resources on TE links or SRLG memebership, a protection audit be run to verify the protection status on all elements.

The computation will display the same computation results page as for Compute Backup. When you return from the computation results page, the Protection Status column in the TE Protection Management window is updated to show the level of protection for each element.

This section describes the necessary steps to perform Audit Protection on one or more network elements.

To run Audit Protection, use the following steps:

Step 1 Choose **Traffic Engineering > TE Protected Elements**.

The TE Protection Management window appears.

Explanation of the **Protection Status** field:

Protection Status—The protection status displayed is determined from the last time an audit was performed. The audit is performed either explicitly by the user or when the protection SR is deployed. The protection status is stated for each network element as either **Protected**, **Not Fully Protected**, or **Unknown**. Click on the column header, **Protected**, to sort elements according to protection status

Step 2 If you only want to perform Audit Protection on selected elements, select one or more tunnels on which to calculate a backup path.

Click **Audit Protection** and select one of the following:

- All Elements
- Selected Elements

The Computation In Progress window appears.

Then the TE Protection Computation Results window appears.

For an explanation of the various window elements, see Compute Backup, page 7-63.

Note Certain attributes, such as Description, that do not impact the computation carried out by the protection tools and updates to these are not displayed in the computation results window.

Step 3 To view the backup tunnels for a particular element, select the element and click **Details**.

The TE Protection Computation Results window appears.

For an explanation of the various window elements, see Compute Backup, page 7-63.

Step 4 Select a row corresponding to a specific warning or violation and click **Details** to display a detailed description in the right pane and backup tunnels associated with the selected item in the bottom pane.

Tunnels associated with a warning or violation are flagged in the **Report** column in the **Backup Tunnels** table in the bottom pane.

The **Accept Solution** button is greyed out because the audit does not provide a solution but rather an evaluation.

For a description of warnings and violations, see Warnings and Violations, page 7-98

Step 5 Click **Cancel** to return to the TE Protection Management window.

The protection status is updated in the Protection Status column.

Audit SR

Audit SR audits protection of all elements in the **TE Protection Management** window against backup tunnels in the TE Protection SR window.

This feature can be used to audit the protection for manually added, modified, and deleted tunnels in the TE Protection SR window before deploying them.

To audit a TE backup tunnel SR, use the following steps:

Step 1 Choose **Traffic Engineering**.

Step 2 Click **Create TE Backup Tunnel**.

The **TE Protection SR** window appears. For an explanation of the various window elements, see Create Backup Tunnel, page 7-39.

Step 3 To audit the protection SR, click **Audit SR**.

Note Audit SR will only be enabled if there are elements in the TE Protection Management window. If this is not the case, the **Audit SR** button will be disabled (grayed out).

The FRR Audit process begins and the TE Protection Computation Results window appears.

See Audit Protection, page 7-64 for a description of the rest of the process. Detail and report windows are identical in these two processes.

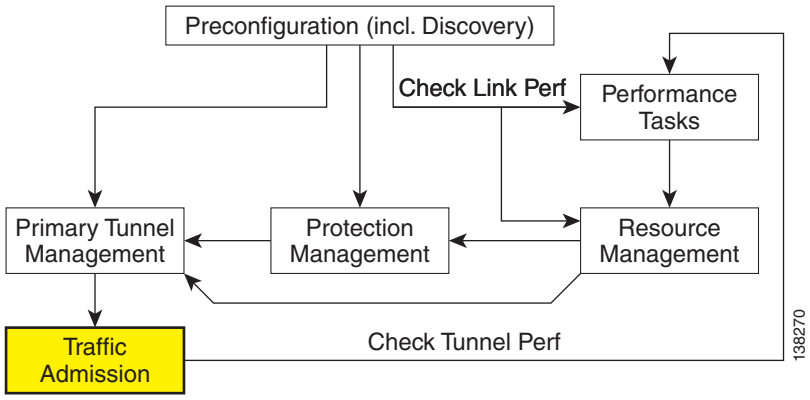
TE Traffic Admission

TE Traffic Admission is the first step towards enabling services on TE tunnels. There are a number of mechanisms that can be used for forwarding traffic into a tunnel to provide basic IP connectivity. The current implementation of Cisco Prime Fulfillment Traffic Engineering Management (Prime Fulfillment) uses both static routing and autoroute announce to inform the routing protocol of the tunnel's presence. Autoroute announce can be also used as part of the routing protocol calculations.

The TE Traffic Admission tool is used to assign traffic to traffic-engineered tunnels.

The highlighted box in Figure 7-3 shows where in Prime Fulfillment TE Traffic Admission occurs.

Figure 7-20 Prime Fulfillment Process Diagram - TE Traffic Admission



Static routing is perhaps the simplest way of forwarding traffic into a tunnel. Traffic that matches a target destination prefix is routed into a particular tunnel.

While this achieves the basic goal of directing traffic into a given tunnel, this approach has limitations. First, the offering of differentiated Class-of-Service (CoS) treatment is limited to destination-based CoS. As each source PE serves as an aggregation point for a number of traffic flows, there is no way to restrict which traffic receives preferential treatment to a destination because access to a tunnel is through general routing. Secondly, it does not generally provide a scalable solution because the static routing mechanism must capture both the large number of subnets that can be served by each PE router, and it must be able to further capture CoS treatment for each of these subnets.

Static routing works best if there is no need to provide differentiated CoS treatment by destination. That is, all packets destined for one or more particular prefixes all receive the same CoS.

This section includes the following:

Creating a TE Traffic Admission SR

The TE traffic admission tool in Cisco ISC TEM only displays primary tunnels (managed or unmanaged) when they are associated with a TE provider and the tunnels are not already associated with a TE Admission SR. That is, the tool is only intended for admitting new traffic onto tunnels currently not carrying any traffic.

To create a TE Traffic Admission SR, use the following steps:

Step 1 Choose **Traffic Engineering**.

Step 2 Click **TE Traffic Admission**.

The TE Traffic Admission Tunnel Selection window appears.

Note If this window does not open, either no tunnels are associated with a TE provider or any tunnels associated with a TE provider are already tied to a TE Admission SR.

The TE Traffic Admission Tunnel Selection window lists all primary tunnels, both managed and unmanaged, that are not already associated with an admission SR.

The **Deploy Status** can be **Pending**, **Deployed**, or **Functional**.

Note Backup tunnels are not displayed in the TE Traffic Admission Tunnel Selection window.

Step 3 Select a TE tunnel by clicking the corresponding radio button and clicking **Select**.

The TE Traffic Admission SR window appears.

The main TE Traffic Admission SR window includes the following fields:

- **Tunnel**—Tunnel name.
- **Description**—Service request description.
- **EXP** [IOS devices only]—Class marking bits for CBTS.

- **Policy** [IOS XR devices only]—Policy marking bits for PBTS.
- **Autoroute announce**—Used to specify that the Interior Gateway Protocol (IGP) should use the tunnel (if the tunnel is up) in its enhanced shortest path first (SPF) calculation.
  - **On**—Autoroute announce is enabled.
  - **Off**—Autoroute announce is disabled.
- **Autoroute Metric**—Used to specify the Multiprotocol Label Switching (MPLS) traffic engineering tunnel metric that the Interior Gateway Protocol (IGP) enhanced shortest path first (SPF) calculation uses.
  - **Absolute**—Absolute metric mode; you can enter a positive metric value.
  - **Relative**—Relative metric mode; you can enter a positive, negative, or zero value.
- **Static Routes**—Lists any static routes that the tunnel uses.
- **Destination**—Name of the static route for the tunnel destination.
- **Distance**—Administrative distance (cost).

Note If TE Traffic Admission SR attributes such as PBTS attributes are changed outside Prime Fulfillment and a TE discovery task is run, the discovery task logs will not report a discrepancy warning and the repository will be updated with the new configuration from the device.

Step 4 When filling out the form, if **Autoroute Announce** is set to **On**, indicate whether **Autoroute Metric** should be **Absolute** or **Relative**.

Step 5 You can also set an optional autoroute metric.

For the relative metric, the range is -10 to 10, for the absolute metric, the range is 1 to 2147483647.

Note CBTS is supported in IOS and PBTS is supported in IOS XR. If the tunnel head router is running IOS XR, the **EXP** fields will not be present and are replaced with the **PBTS** fields.

When clicking the **Add** button, the Add TE Static Route window appears.

Step 6 In the Add TE Static Route window, specify at a minimum a **Destination** IP address (w.x.y.z/n).

Optionally specify an administrative **Distance**. It is recommended that you either define one or more static routes or, alternatively, that you define an autoroute.

Step 7 Click **OK** to accept the entries or **Cancel** to exit the window.

In the main TE Traffic Admission SR window, you can add another TE Static Route or edit existing routes.

Step 8 Click **Save** to save the service request.

The Service Requests window appears with the TE Traffic Admission SR in **REQUESTED** state and the Operation Type set to **ADD**.

For more information on working with service requests, see the managing service requests part elsewhere in this guide.

To deploy the service request from the Service Requests window, see Deploying a TE Traffic Admission SR, page 7-69.

Deploying a TE Traffic Admission SR

As opposed to the TE Primary Tunnel SR, Backup Tunnel SR, and TE Resource Modification windows, a TE Admission SR must be deployed from the general Service Request Manager window.

To deploy a TE Admission SR, use the following steps:

Step 1 Choose **Operate > Service Request Manager**.

The Service Requests window appears.

The Service Requests window includes the following elements:

- **Job ID**—Job ID for the SR.
- **Data Files**—This field is used for variable substitutions via templates and currently do not apply to TEM SRs.
- **State**—Indicates whether the tunnel state is DEPLOYED or NOT DEPLOYED and whether it is Conformed or Not Conformed.
- **Type**—The type of service request, indicating which service issued the request. For a detailed description of the possible service types, see the managing service requests part elsewhere in this guide.
- **Operation Type**—SR operation on the tunnel, can be either **ADD, MODIFY, DELETE,** or **ADMIT**. Applicable only to tunnels in the current SR.
- **Creator**—ID for the user who created the SR.
- **Customer Name**—Name of the customer to which the SR applies.
- **Policy Name**—Name of the policy associated with the SR.
- **Last Modified**—Date and time when the SR was last modified.
- **Description**—SR description provided by the user.

Step 2 Select the desired service request and click **Deploy**.

A drop-down menu appears under the **Deploy** button. In the drop-down menu, select **Deploy** or **Force Deploy**. After having been successfully deployed, the **State** of the SR changes to **Deployed**.

The Service Requests window (**Operate > Service Request Manager)** appears and displays the state of the deployed SR.

For more information on working with service requests, see the managing service requests part elsewhere in this guide.

Other Traffic Admission SR Operations

As opposed to other service requests, TE Traffic Admission SRs can be decommissioned in the Service Requests window.

Edit and decommission operations for TE Traffic Admission service requests are handled in the Service Request Manager window. These operations are described in the managing service requests part elsewhere in this guide.

Viewing the SR State

To view a service request state, go to **Operate > Service Request Manager**.

If the SR does not enter the **Deployed** state, go to the **Task Logs** window to see the deployment log (**Operate > Task Manager > Logs**) as described in SR Deployment Logs, page 10-46.

Administration

A number of administrative features in Cisco Prime Fulfillment Traffic Engineering Management (TEM) are common to Prime Fulfillment. Instructions on how to use these features are described in detail starting in Manage Active Users and User Account, page 14-1.

In this section, only TE-specific administrative features are described.

This section includes the following:

TE User Roles

A TE user role can be a predefined or a user-specified role defining a set of permissions. For a detailed description of user roles in Prime Fulfillment and how to use them, see User Roles, page 14-16.

To access the User Roles window and locate the TE user roles, choose **Administration > Roles**. The User Roles window appears.

There are two pre-defined TEM user roles:

- **TERole**—Grants full permission to TEM operations.
- **TEServiceOpRole**—Grants permission only to manage the TE Admission SR.

TE Policies

Policies are used to define common tunnel attributes. Attributes such as bandwidth pools, hold and setup priority, and affinity bits, are set manually during policy creation as described below.

This section describes the following policy operations:

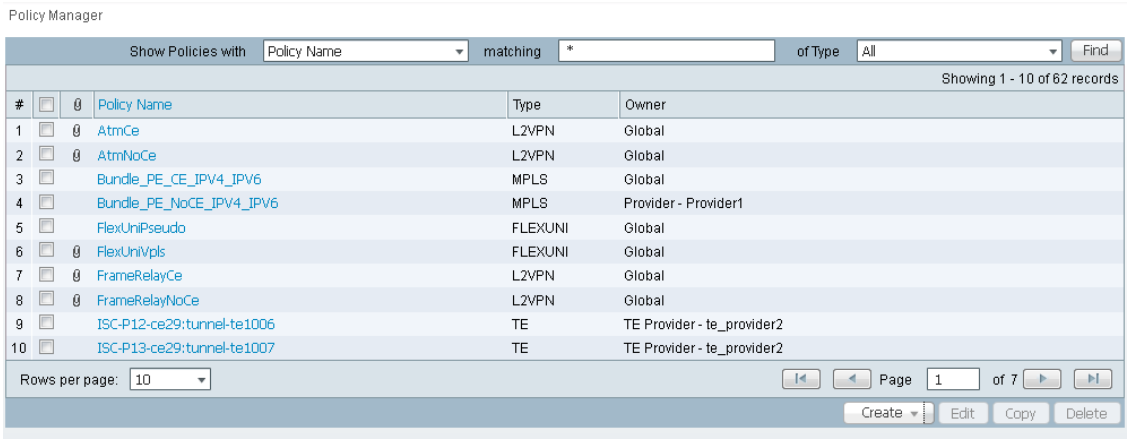
Create Policy

Prime Fulfillment allows you to create TE-specific policies in a manner similar to other policies. To create a TE policy, use the following steps:

Step 1 Choose **Service Design > Policy Manager**.

The Policy Manager window in Figure 7-21 appears.

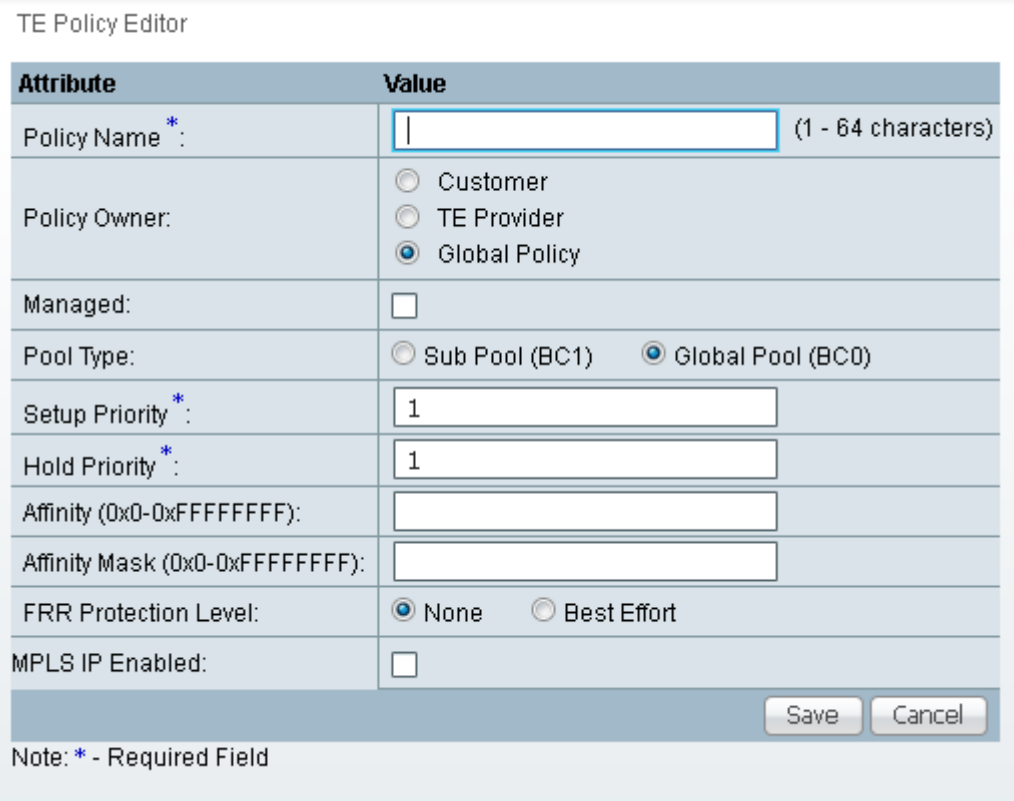
Figure 7-21 Policy Manager



Step 2 Click **Create** and select **TE Policy** to set up a new TE policy.

The TE Policy Editor window in Figure 7-22 appears.

Figure 7-22 TE Policy Editor



The TE Policy Editor window includes the following fields:

- **Policy Name**—Name of the TE policy chosen by the user.
- **Owner**—The owner of the TE policy.
- **Managed**—Check this box to make the policy to be used by managed tunnels. When clicked, both the setup and hold priorities are set to zero and these are not editable. If the box is unchecked, the setup/hold priorities can be set to a value between 1 and 7.

  Clicking the **Managed** check box will add some extra fields in the TE Policy Editor corresponding to two additional protection levels for **FRR Protection Level** (Fast Re-Route) and a new field, **Delay Constraint**.
- **Pool Type**—Tunnel bandwidth pool type for this policy. For a definition of pool types, see the Bandwidth Pools section in the *Cisco Prime Fulfillment Theory of Operations Guide 6.2*.
  - **Sub Pool (BC1)**—Bandwidth will be reserved from Sub Pool.
  - **Global Pool (BC0)**—Bandwidth will be reserved from Global Pool.
- **Setup Priority**—Priority used when signaling an LSP for the tunnel to determine, which of the existing tunnels can be preempted. Valid values are from 0 to 7, where a lower number indicates a higher priority. Therefore, an LSP with a setup priority of 0 can preempt any LSP with a non-0 hold priority.
- **Hold Priority**—Priority associated with an LSP for the tunnel to determine if it should be preempted by other LSPs that are being signaled. Valid values are from 0 to 7, where a lower number indicates a higher priority.

- **Affinity**—Attribute values required for links carrying the tunnel (bit values are either 0 or 1).
- **Affinity Mask**—Which attribute values should be checked. If a bit in the mask is 0, a link's attribute value of that bit is irrelevant. If a bit in the mask is 1, the link's attribute value and the tunnel's required affinity for that bit must match.
- **FRR Protection Level**—Level of Fast Reroute protection required on the primary tunnel.
  - **None**—No backup tunnel needed.
  - **Best Effort**—Use backup tunnel if available.
  - **Link & SRLG**—Primary tunnel must pass through only links or SRLGs that are FRR-protected
  - **Link, SRLG & Node**—Primary tunnel must pass through only intermediate nodes and links or SRLGs that are FRR-protected.
- **MPLS IP Enabled**—This configures the tunnel with the **mpls ip** command if enabled.

Edit Policy

A policy can be edited only if it is not associated with a tunnel.

To edit a TE policy, use the following steps:

Step 1 Choose **Service Design > Policy Manager**.

The Policies window in Figure 7-22 appears.

Step 2 Select the desired policy and click **Edit**.

The TE Policy Editor window appears. The policy editor is described in Create Policy, page 7-71. The only difference between the create and edit processes is that the policy name and owner are not editable when editing a policy.

Step 3 Make the desired changes to the policy attributes and click **Save**.

If the save operation succeeds, the new TE policy now appears in the Policies window. If not, the **Status** box will indicate the type of error that occurred and, when possible, the corrective action required.

Delete Policy

A policy can be deleted only if it is not associated with a tunnel.

To delete a TE policy, use the following steps:

Step 1 Choose **Service Design > Policy Manager**.

The Policies window in Figure 7-22 appears.

Step 2 Select the desired policy and click **Delete**.

The Confirm Delete window appears

Step 3 Check the policy marked for deletion and click **OK**.

The Policies window refreshes and the selected policy disappears.

TE Tasks

Prime Fulfillment currently offers three TE-specific tasks that are used in a manner similar to other tasks:

- **TE Discovery (Full and Incremental)**—Populates the repository with data from the TE network. Discrepancies are reconciled and/or reported.
- **TE Functional Audit**—Performs functional audit on TE Primary or Backup SRs in certain states.
- **TE Interface Performance**—Calculates the interface/tunnel bandwidth utilization.

This section focuses on describing how to create TE Functional Audit and TE Interface Performance tasks. Instructions on how to create a TE Discovery task are included in TE Network Discovery, page 7-10.

Creating a TE Task

TE tasks are managed in the **Task Manager**, which is accessed by selecting **Operate > Task Manager**.

The Tasks window appears.

For a detailed description of the window elements in the Tasks window, see Task Manager, page 10-23.

This page shows all collection and deployment tasks that have been executed. Note that a task could be scheduled to happen once or there could be several scheduled runs of a task. The schedule can be viewed by selecting a task and clicking **Schedules**.

Creating a TE Functional Audit Task

For each tunnel in the SR, the TE Functional Audit task checks the LSP currently used on a router against the LSP stored in the repository:

- tunnel down—Ignore (do not check)
- tunnel up—Check the LSP used on the router against the one stored in the repository:
  - If they are the same, the tunnel and the SR are both set to **Functional**.
  - If they are different, both the tunnel and the SR are set to **Broken**.
- tunnel missing from router—SR left untouched. The tunnel state is set to **Lost**.

This task only performs functional audit on TE Primary or Backup SRs, which are not in one of the following states:

- **Closed**
- **Requested**
- **Invalid**
- **Failed Deploy**

For more information on working with service requests, see the managing service requests part elsewhere in this guide.

To create a TE Functional Audit task, use the following steps:

Step 1 Choose **Operate > Task Manager**.

Step 2 Click **Audit > TE Functional Audit** to open the Create Task window.

For a detailed description of the window elements in the Create Task window, see Task Manager, page 10-23.

Step 3 Modify the **Name** or **Description** fields as desired and click **Next**.

The Task Service Requests window appears.

Step 4 Click **Add** to add a task service request.

The Select Service Request(s) window appears.

Step 5 Select an SR using the **Select** button.

Note Only SRs of type TE Tunnel or TE Protection will be accepted.

The Selected Service Request(s) window closes and the selected task(s) now appears in the Task Service Requests window. To add other SRs, repeat the procedure in Step 4 and Step 5.

Step 6 In the Task Service Requests window, click **Next**.

The Task Schedules window appears.

Step 7 Click **Now** to start the task immediately or **Create** to create a task schedule.

When selecting **Now**, a line is added to the **Task Schedules** window. When selecting **Create**, the Task Schedule window appears.

Step 8 In the Task Schedule window, indicate when and how often to run the task.

Step 9 Click **OK**.

The scheduled task should now appear in the **Task Schedules** table.

Note The default setting is to schedule a single TE Functional Audit task to take place immediately ("**Now**").

Step 10 Click **Next**.

The Task Schedule window now shows the new task in its list of created tasks. A summary of the scheduled task appears.

Step 11 Click **Finish**.

This adds the task to the list of created tasks in the Tasks window.

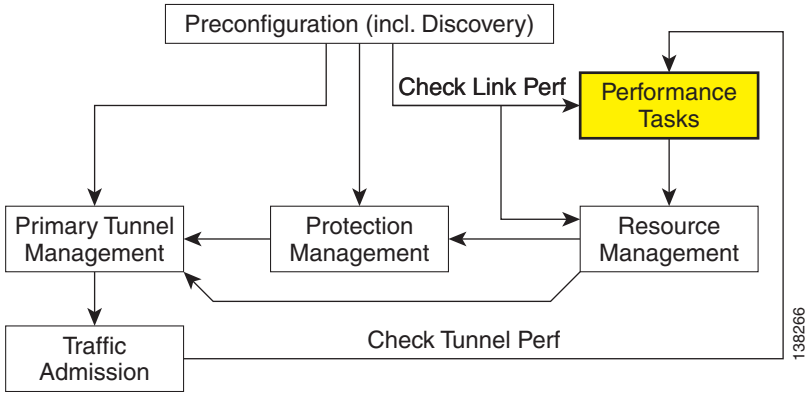
To view the task logs for the created tasks, see Viewing a Task Log, page 10-46.

Creating a TE Interface Performance Task

This task calculates interface/tunnel bandwidth utilization using the Simple Network Management Protocol (SNMP).

The highlighted box in Figure 7-3 shows where in Prime Fulfillment traffic admission occurs.

Figure 7-23 Prime Fulfillment Process Diagram - TE Interface Performance



Calculating utilization depends on how data is presented for the object you want to measure. Interface utilization is the primary measure used for network utilization. Because MIB-II variables are stored as counters, you must take two poll cycles and figure the difference between the two (hence, the delta used in the equation).

Three variables are required:

- task duration—how long the task will run (in seconds)
- frequency—how frequent the data will be collected (in seconds)
- interval—the distance between two poll cycles (in milliseconds).

The following explains the variables used in the formulas:

- delta(traffic in)—the delta between two poll cycles of collecting the SNMP input object, which represents the number of inbound units of traffic
- delta(traffic out)—the delta between two poll cycles of collecting the SNMP output object, which represents the number of outbound units of traffic
- bandwidth—the speed of the interface.

A more accurate method is to measure the input utilization and output utilization separately, using the following formula:

$$\text{Input utilization} = \frac{\text{delta(traffic in)} \times 8 \times 100}{(\text{number of seconds in delta}) \times \text{bandwidth}}$$

$$\text{Output utilization} = \frac{\text{delta(traffic out)} \times 8 \times 100}{(\text{number of seconds in delta}) \times \text{bandwidth}}$$

To create a TE Interface Performance task, use the following steps:

Step 1 Choose **Operate > Task Manager**.

Step 2 Click **Create > TE Interface Performance** to open the Create Task window for a new TE Interface Performance task.

For a detailed description of the window elements in the Create Task window, see Task Manager, page 10-23.

Step 3 Modify name and description if needed and click **Next**.

The Select TE Provider window appears.

Step 4 Click a radio button to select a TE provider.

Step 5 Click **Next**.

The TE Performance Collection window appears.

Step 6 Enter desired values in the **Task Duration**, **Task Frequency**, and **Task Interval** fields.

Note If the **Task Interval** field is set too low, the MIB might not be updated, in which case the TE Performance Report will not show any traffic. For tunnels or links on IOS routers, it is recommended to set the interval to 1000 ms; for IOS XR routers, a recommended interval is 5000 ms. Note that these values might need to be tuned to suit your specific environment.

Step 7 Use the **Add** button to select a tunnel or link on which to run the interface performance task:

- **TE Tunnel**—Add a TE tunnel. The Select Tunnel(s) window appears.
- **TE Link**—Add a TE link. The Select Link(s) window appears.

Step 8 Select one or more of tunnels and links and click **Next**.

The selected tunnels and links are added to the **Targets** list in the TE Performance Collection window. The Task Schedules window appears.

Step 9 Click **Now** or **Create** to create a task schedule.

When you select **Create** to customize the schedule, the Task Schedule window appears (with **Now**, this step is skipped).

Note The default setting is to schedule a single TE Interface Performance task to take place immediately ("**Now**").

Step 10 In the Task Schedule window, make your selections to define when and how often to run the task.

Step 11 Click **OK**.

The scheduled task should now appear in the **Task Schedules** table.

Step 12 Click **Next**.

A summary of the scheduled task appears.

Step 13 Click **Finish**.

This adds the task to the list of created tasks in the Tasks window.

To view the TE Performance Report that is generated for TE Interface Performance task(s), see TE Performance Reports, page 10-47.

To view the task logs for the created tasks, see Viewing a Task Log, page 10-46.

SR History and Configlets

The history and configlets associated with individual service requests can be viewed from the Service Requests window when you select a service request and click the **Details** button.

The history of a service request is essentially a state change report. It lists the various states that elements associated with an SR has transitioned between and reports relevant details pertaining to these state changes.

Configlets for devices associated with service requests are in simple scrollable text format.

For more information about these features and how to manage service requests, see the managing service requests part elsewhere in this guide.

Managing the Locking Mechanism

Whenever a task is performed that incurs a database update, which might affect the resource and hence the result of a tunnel computation, it locks the system before the update and releases it at completion of the update. If for some reason the lock is not released, other updates that require the lock are blocked.

The purpose of the lock feature is to prevent concurrent and mutually inconsistent planning activities from being committed to the database. Meaning, if each user takes the same snapshot of the the repository, performs computations, and tries to commit what he/she sees, the locking mechanism helps synchronize the commit and ensures that no commit invalidates other commits.

If the system is locked for prolonged periods of time, the administrator should check if anyone is performing long planning tasks and take note of which process locked the system and report it. If the administrator is sure that no one is using the system, it can be unlocked by using the lock manager.

Prime Fulfillment has two kinds of locks:

- TE provider lock—Locks managed tunnels, backup tunnels, resource SRs, and TE Discovery.
- TE router lock—Locks unmanaged tunnels.

Each system lock is linked to a TE provider. In the following, procedures for unlocking each system lock are listed.

Unlocking the TE Provider Lock

To unlock the TE provider, use the following steps:

Step 1 Choose **Traffic Engineering > Providers**.

The TE Providers window appears.

Step 2 Select a TE provider that is locked by checking the corresponding check box.

Step 3 Click **Manage Lock**.

The System Lock Management window appears.

The text fields in this window are read-only.

Step 4 To unlock, click the **Unlock** button.

The System Lock Management window closes and the **System Lock Status** field in the TE Providers window is updated accordingly.

Unlocking the TE Router Lock

To unlock the TE router lock, use the following steps:

Step 1 Choose **Traffic Engineering** > **Nodes**.

The TE Nodes List window appears.

Step 2 Select a TE node that is locked by clicking the corresponding check box.

Step 3 Click **Manage Lock**.

The System Lock Management window appears. The text fields in this window are read-only.

Step 4 To unlock, click the **Unlock** button.

The System Lock Management window closes and the **System Lock Status** field in the TE Nodes List window is updated accordingly.

Locking Operation Errors

TEM locks the TE Provider or TE Router object respectively for the duration of a save and deploy operation to ensure database consistency.

This section describes the following errors:

- Modifying Locked Object, page 7-79
- Modifying Object After Lock Is Released, page 7-80
- Deleting Link with Associated TE Object, page 7-80
- Deleting Link Without Associated TE Object, page 7-80

Modifying Locked Object

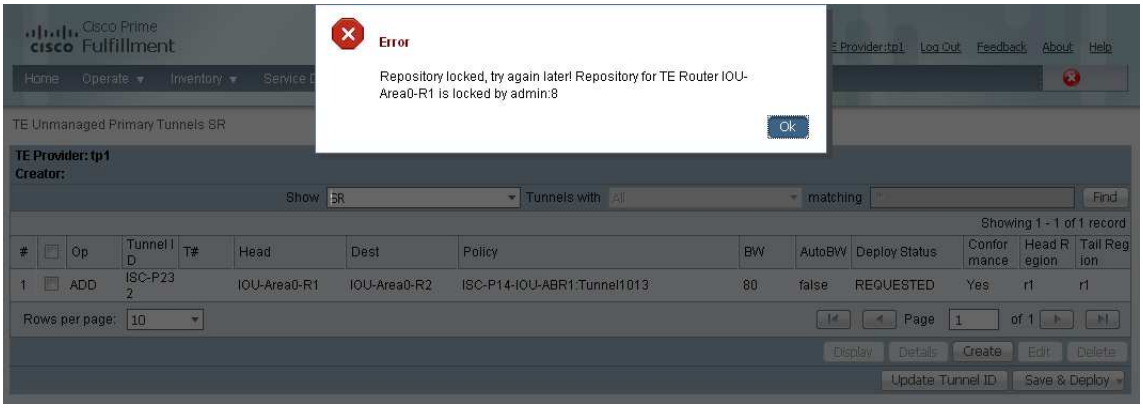
If you attempt to modify a locked object, you will be informed that the object cannot be modified because another user is making changes. You will receive the error message shown in Figure 7-24.

Figure 7-24 Modifying Locked Object

Modifying Object After Lock Is Released

If you attempt to modify an object after the lock is released, Prime Fulfillment will check that your current working version of the object is up to date. If not, you will be instructed to restart with a new version of the object as your data is now out of date. You will receive the error message shown in Figure 7-25.

Figure 7-25 Modifying Object After Lock Is Released

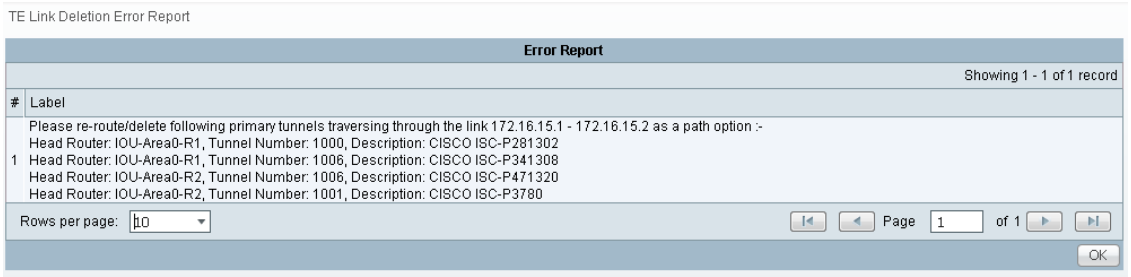


Deleting Link with Associated TE Object

Link removal is not allowed if the link is associated with an explicit path or is traversed by a tunnel.

If you try to delete a link with one or more associated objects, the error message in Figure 7-26 is displayed.

Figure 7-26 Deleting Link with Associated TE Object

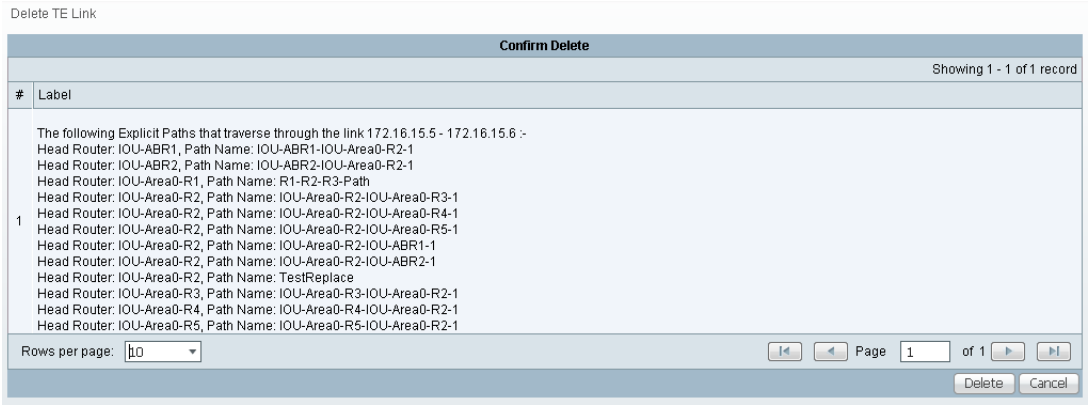


Deleting Link Without Associated TE Object

A link can be removed if it is not traversed by a tunnel, even if it is associated with an explicit path.

When you try to delete such a link, the type of report shown in Figure 7-27 will be displayed.

Figure 7-27 Deleting Link Without Associated TE Object



TE Topology

The TE Topology tool provides a graphical view of the network set up through the Cisco Prime Fulfillment web client. It gives a graphical representation of the various network elements, including devices, links, and tunnels. It also displays devices that Prime Fulfillment is unable to identify but which have been discovered with the TE Discovery tool to be part of the network.

The TE Topology tool is accessed from the Traffic Engineering menu.

The TE Topology tool is used to visualize the TE network based on the data contained in the repository. To that end, it provides a number of ways of manipulating the display, for example by applying algorithms to the graph layout, importing maps, and so on.

The tool is accessed from a TE Topology Interface Applet that displays the TE topology through a Java applet within the browser.

This section describes how to use the topology tool.

It includes the following sections:

Using the TE Topology Interface Applet

The TE Topology Interface Applet (Topology Applet) provides a means of visualizing the network and tunnels present in the network. The web-based GUI is the primary means of visualizing the network information. The Topology Applet simply augments the web-based GUI to provide you with a different presentation format.

The features offered through the Topology Applet are:

- TE Topology rendering
- Highlighting of network elements
- Tunnel overlay (unmanaged, primary, and backup)
- Topology layout persistence
- Integration with web page content.

To access the Topology Applet, use the following steps:

Step 1 Choose **Traffic Engineering > Topology**.

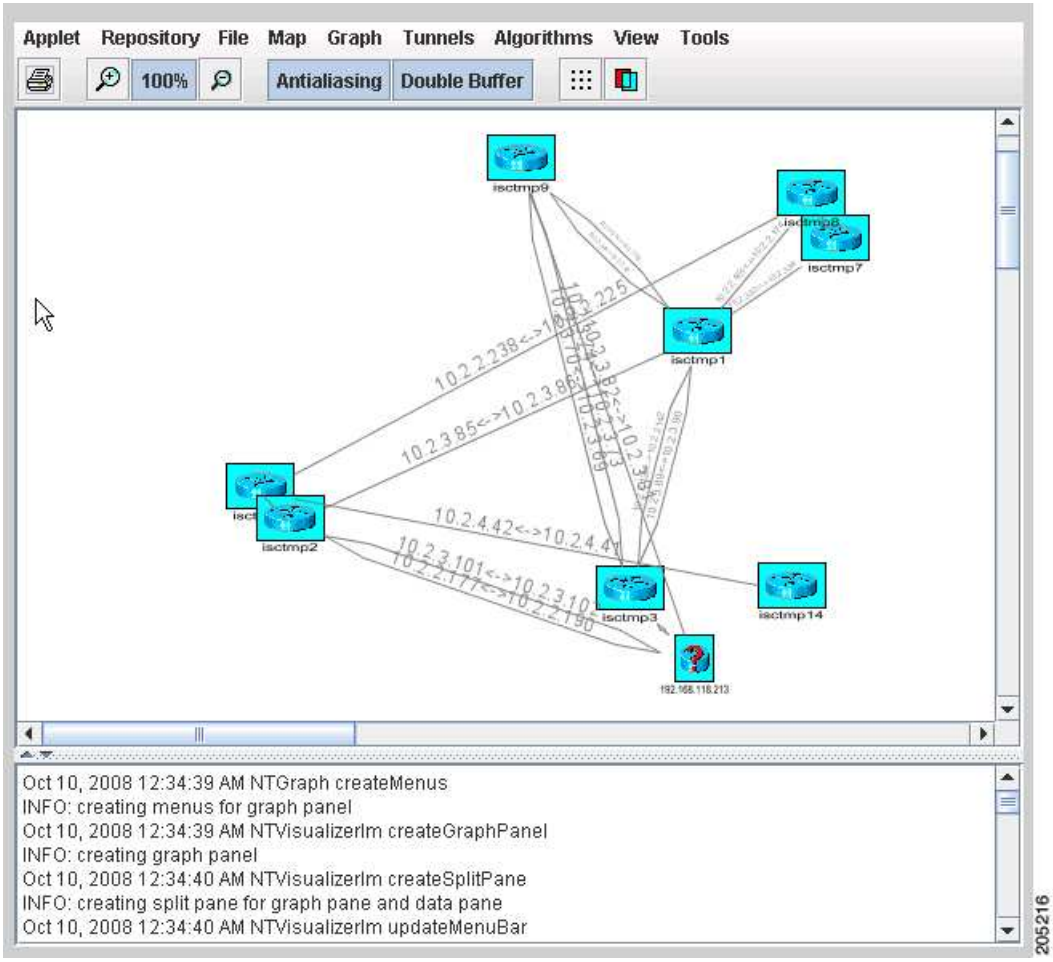
Step 2 Click **TEM Topology Interface Applet**.

If the security certificate for the topology applet has not been accepted previously, you might get a security warning window.

Step 3 Click **Yes** or **Always** to accept the authenticity of the security certificate.

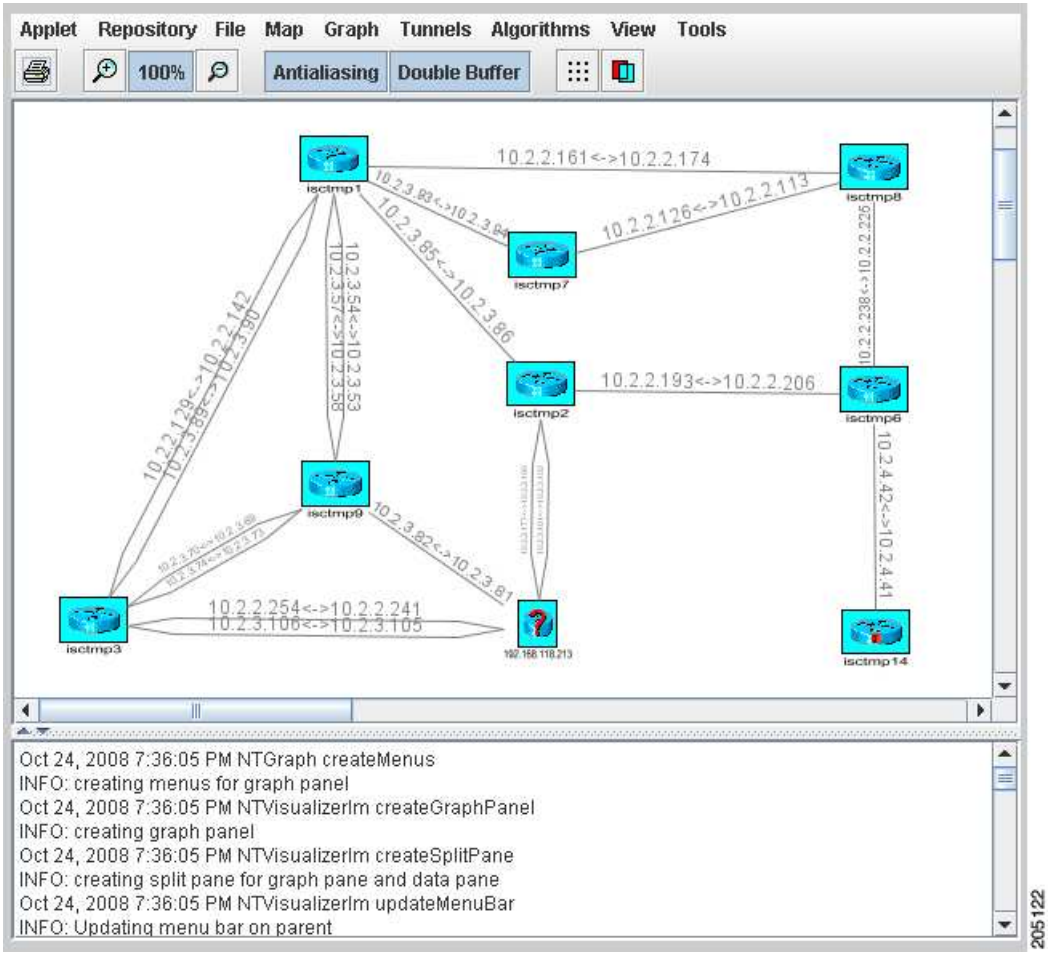
The Topology Display applet window in Figure 7-28 appears.

Figure 7-28 Topology Display Applet in Unordered State



After the nodes have been arranged to your liking, you might end up with a topology display similar to the one in Figure 7-29.

Figure 7-29 Topology Display Applet with User-Arranged Topology



Displaying and Saving Layouts

Use the two operations in the **Repository** menu, **Layout Graph** and **Save Graph Layout**, to display or save the current layout of the network graph.

Prior to generating the graph layout, the coordinates must be set on each of the network devices. Otherwise, the graph will have a random layout.

- **Layout Graph**—The graph is laid out from the repository. If a graph layout is already present, that layout is cleared once you click **Yes** in the **Clear Graph Layout** confirmation box. If the layout has not previously been saved, a random layout of the repository contents is drawn. If it has been saved previously, the saved layout is redrawn.
- **Save Graph Layout**—Save the current graph layout. Doing so will ensure that whenever the graph layout is cleared with **Layout Graph** or the topology applet is closed, the same layout will be created when the applet is restarted. If a map was used, the map is also redrawn.

Using Maps

You can associate a map with each view. Currently, the topology viewer only supports maps in the Environmental Systems Research Institute, Inc. (ESRI) shape format. The following sections describe how to load maps and selectively view map layers and data associated with each map.

The map features are accessed from the **Map** menu in the Topology window.

To access the **Map** menu, use the following steps:

Step 1 Choose **Traffic Engineering > TE Topology.**

Step 2 Start the **TM Topology Interface Applet**.

If link and node data for your network is already in the repository, a Progress Report lists the various network elements as the corresponding data is loaded.

Step 3 Select the **Map** menu.

The menu appears.

From the **Map** menu, you can either load or clear (remove) maps as described in the following.

Loading a Map

You might want to set a background map showing the physical locations of the displayed devices. To load a map, use the following steps:

Step 1 In the menu bar, select **Map > Load**.

Providing the web map server is running, the Map Chooser window appears.

Step 2 Make your selections in the Map Chooser window.

The right-hand side of the window contains a small control panel, which allows you to select the projection in which a map is shown. A map projection is a projection which maps a sphere onto a plane. Typical projections are Mercator, Lambert, and Stereographic.

For more information on projections, consult the Map Projections section of Eric Weisstein's World of Mathematics at:

http://mathworld.wolfram.com/topics/MapProjections.html

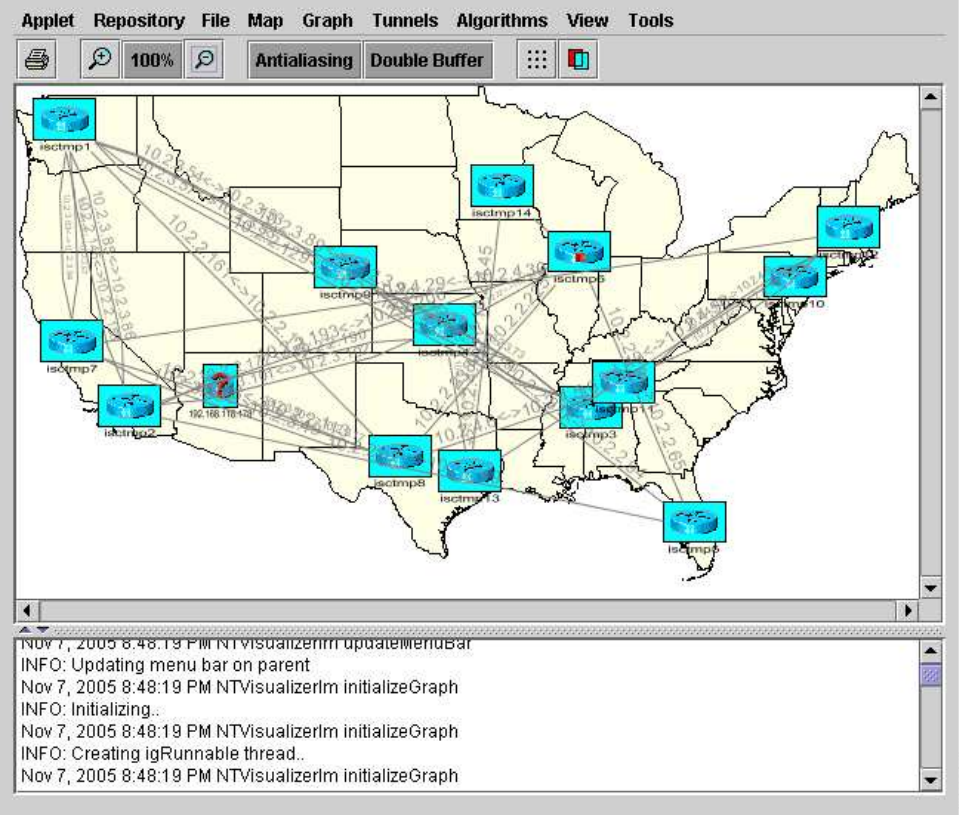
If desired, make changes to the settings in the **Longitude Range** and **Latitude Range** fields.

Step 3 Select a map file and click **Open** to load the map.

Selecting the map file and clicking the **Open** button starts loading it. Maps can consist of several components and thus a progress dialog is shown informing you which part of the map file is loaded.

A map similar to the one in Figure 7-30 appears.

Figure 7-30 Loaded Map



Step 4 Use the various functions in the menus of the Topology Display window to manipulate the display contents in the Topology view. Some of these are described in subsequent sections.

Adding New Maps

You might need to add your own maps to the selection of maps available to the Topology Tool. This is done by placing a map file in the **$ISC_HOME/resources/webserver/tomcat/webapps/ipsc-maps/data** directory or a subdirectory thereof within the Prime Fulfillment installation. To make this example more accessible, assume that you wish to add a map of Toowong, a suburb of Brisbane, the capital of Queensland. The first step to do so is to obtain maps from a map vendor. All maps must be in the ESRI shape file format (see **ESRI shapefile technical description**). In addition, a data file can accompany each shape file. Data files contain information about objects and the corresponding shapes are contained within the shape file. Let us assume that the vendor provided four files:

- toowong_city.shp
- toowong_city.dbf
- toowong_street.shp
- toowong_street.dbf

We have to create a .map file that informs the TE Topology tool about layers of the map. In this case we have two layers: a city and a street layer. The map file, say, Toowong.map, would thus have the following contents:

```
toowong_city
toowong_street
```

It lists all layers that create a map of Toowong. The order is important, as the first file forms the background layer, with other layers placed on top of the preceding layers.

Having obtained shape and data files and having written the map file, place all five files in the **$ISC_HOME/resources/webserver/tomcat/webapps/ipsc-maps/data** directory. All map files must be located in this folder. After this is done, the map is automatically accessible to the topology viewer.

Clearing Maps

To clear the active map, select **Map > Clear**.

Use this feature to clear (remove) the active map to leave only nodes and links in the corresponding network.

Using Highlighting and Attributes

The **Graph** menu provides access to a range of tools to manage and manipulate graphs.

Use the JavaServer Pages to look at the list of nodes, links, and tunnels. From the JSP pages, select the display button at the bottom of the window to highlight elements.

The tools in the **Graph** menu serve to modify the appearance of the topology.

These are described in the following sections.

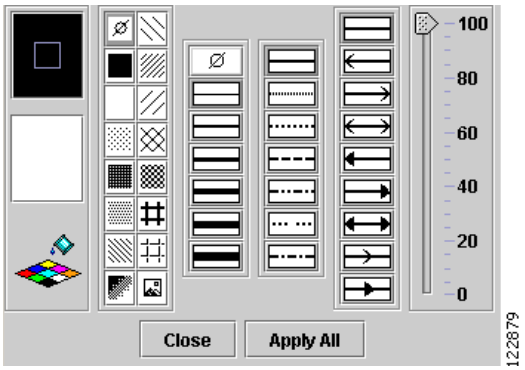
Clear Highlighting

Clear Highlighting serves to remove highlighting from specific elements as listed in its submenus.

Add/Modify Attributes

When you select **Attributes** from the **Graph** menu, the Graphic Attributes window in Figure 7-31 appears.

Figure 7-31 Graphic Attributes



The **Add/Modify Attributes** tool is used as follows:

Step 1 Select graph elements (nodes/links) in the topology display.

Use Ctrl/Shift to select multiple elements.

Step 2 Choose **Graph > Attributes** to open the Graphic Attributes window.

Step 3 Change the desired attributes and click **Apply All**.

Note Only selected links (Step 1) are affected.

Clear Current Graph Layout

Use the **Clear** function in the **Graph** menu to remove the topology graph from the current view.

Although this is also achieved with **Layout Graph** in the **Repository** menu, **Layout Graph** re-creates the graph last saved in the repository in addition to clearing the graph.

Using AntiAlias, BackingStore, DoubleBuffer

AntiAlias, found in the **Graph** menu, is used to create smoother lines and a more pleasant appearance at the expense of performance.

BackingStore allows graphics content to be automatically saved when moved to the background and regenerated when returned to the foreground. This helps avoid superfluous refreshing.

DoubleBuffer enables double buffering for dragging elements on the graph.

Using Algorithms

In the **Algorithms** menu various algorithms can be used to enhance and otherwise alter the graph layout.

Note The algorithms only work when the nodes are interconnected with links.

Spring is a graph layout algorithm that optimizes the graph layout based on weights.

Randomize rearranges the nodes in the current topology layout at random.

If there are overlapping links, the layout can be optimized by selecting **Optimize Links**.

The spring settings are used to enhance the appearance of the topology display according to user preferences. When selecting **Spring Settings**, the Spring Settings window appears.

Sample Configlets

The configlets included in this section show the CLIs generated by Prime Fulfillment for particular services and features. Each configlet example provides the following information:

- Service

- Feature
- Devices configuration (network role, hardware platform, relationship of the devices and other relevant information)
- Sample configlets for each device in the configuration
- Comments.

All examples in this section assume the presence of an MPLS-TE core.

Note The configlets generated by Prime Fulfillment are only the delta between what needs to be provisioned and what currently exists on the device. This means that if a relevant CLI is already on the device, it does not show up in the associated configlet.

This section provides sample configlets for traffic engineering service provisioning in Cisco Prime Fulfillment.

It includes the following sections:

Primary Tunnel Configlet (IOS)

Configuration

- Service: MPLS-TE primary tunnel
- Feature: MPLS TE configlet (IOS) for deploying a primary tunnel
- Device configuration: CISCO12410 with IOS 12.0(32)S.

Configlets

IOS Device Configuration	Comments
! **Explicit path:** ip explicit-path name isctmp2-isctmp8-1 enable next-address 10.2.2.145 next-address 10.2.2.174 !	Create an explicit path with the specified next addresses, which indicate the strict path that the tunnel traverses. This explicit path is used by the primary tunnel detailed above.
! **Primary tunnel:** interface Tunnel1000 description CISCO ISC-P24 ip unnumbered Loopback0 no ip directed-broadcast tag-switching ip tunnel destination 192.168.118.183 tunnel mode mpls traffic-eng tunnel mpls traffic-eng priority 0 0 tunnel mpls traffic-eng bandwidth 10 tunnel mpls traffic-eng affinity 0x0 mask 0x0 tunnel mpls traffic-eng path-option 1 explicit name isctmp2-isctmp8-1 tunnel mpls traffic-eng path-option 2 dynamic tunnel mpls traffic-eng record-route !	Create a TE primary tunnel with the following attributes: - tag switching: This command is generated because the policy has the 'mpls ip' flag enabled. This allows the TE tunnels to be used for MPLS VPN traffic. - Destination 192.168.118.183 - TE encapsulation - Setup and hold priorities both 0 - Bandwidth global pool 10 kbps - Tunnel affinity 0x0 - Explicit first path option - Dynamic second path option

Bandwidth Protection Backup Tunnel Configlet (IOS)

Configuration

- Service: MPLS-TE with FRR (Fast Re-Route)
- Feature: This tunnel protects primary tunnel traffic in the event of either a link or node failure
- Device configuration: CISCO12410 with IOS 12.0(32)S.

Configlets

IOS Device Configuration	Comments
! Explicit path: ip explicit-path name isctmp5-isctmp4-1 enable next-address 10.2.2.145 next-address 10.2.2.174 !	Create an explicit path with the specified next addresses, which indicate the strict path that the tunnel traverses. This explicit path is used by the backup tunnel detailed above.
! Backup tunnel: interface Tunnel1001 description CISCO ISC-B30 ip unnumbered Loopback0 tunnel destination 192.168.118.213 tunnel mode mpls traffic-eng tunnel mpls traffic-eng backup-bw sub-pool 30000 tunnel mpls traffic-eng priority 0 0 tunnel mpls traffic-eng affinity 0x0 mask 0x0 tunnel mpls traffic-eng path-option 1 explicit name isctmp5-isctmp4-1 tunnel mpls traffic-eng record-route !	Create a TE backup tunnel with the following attributes: - Destination 192.168.118.213 - TE encapsulation - Protect subpool bandwidth of 30000 kbps - Setup and hold priorities both 0 - Tunnel affinity 0x0 - Explicit first path option
interface POS0/1 mpls traffic-eng backup-path tunnel 1001 !	Backup tunnel 1001 protects interface POS0/1

Connectivity Protection Backup Tunnel Configlet (IOS)

Configuration

- Service: MPLS-TE with FRR (Fast Re-Route)
- Feature: MPLS TE configlet (IOS) for deploying a connectivity protection backup tunnel and its associated exclude address path
- Device configuration: CISCO12410 with IOS 12.0(32)S.

Configlets

IOS Device Configuration	Comments
! **Explicit path:** ip explicit-path name L47-excl enable exclude-address 192.168.1.18 ! !	Create an explicit path with an exclude address, which indicates the IP address the path should avoid. This explicit path is used by the backup tunnel detailed above.

! **Backup tunnel:**
interface Tunnel1000
  description CISCO ISC-B1
  ip unnumbered Loopback0
  tunnel mode mpls traffic-eng
  tunnel destination 10.52.96.38
  tunnel mpls traffic-eng priority 0 0
  no tunnel mpls traffic-eng bandwidth
  tunnel mpls traffic-eng path-option 1 explicit name L47-excl
  tunnel mpls traffic-eng affinity 0x0 mask 0x0
  tunnel mpls traffic-eng backup-bw sub-pool unlimited
  tunnel mpls traffic-eng record-route
!
	Create a TE backup tunnel with the following attributes:
	- Destination 10.52.96.38
	- TE encapsulation
	- Setup and hold priorities both 0
	- Backup tunnel does not reserve any bandwidth
	- Explicit first path option
	- Tunnel affinity 0x0
	- Unlimited backup bandwidth for protecting sub pool

interface ATM4/0.1 point-to-point mpls traffic-eng backup-path Tunnel1000	Set up backup path on ATM interface.

TE Traffic Admission Configlet Using CBTS (IOS)

Configuration

- Service: TE Traffic Admission
- Feature: MPLS TE configlet (IOS) for admitting traffic using Class-Based Tunnel Selection (CBTS)
- Device configuration: CISCO12410 with IOS 12.0(32)S.

Configlets

IOS Device Configuration	Comments
! **TE Traffic Admission using CBTS:** interface Tunnel1000 tunnel mpls traffic-eng exp 1 2 3 ! ! **Static route:** ip route 192.168.118.189 255.255.255.255 Tunnel1000	Class-based tunnel selection where traffic with EXP bit 1, 2, or 3 are selected Create a static route, which admits all traffic destined for 192.168.118.189 into the above-configured Tunnel 1000.

The above is then deployed to an already existing primary tunnel such as the Primary Tunnel Configlet (IOS), page 7-89.

TE Traffic Admission Configlet (IOS)

Configuration

- Service: TE Traffic Admission
- Feature: MPLS TE configlet (IOS) for TE Traffic Admission
- Device configuration: OSR-7609 with IOS 12.2(33)SRA.

Configlets

IOS Device Configuration	Comments

! **TE Traffic Admission:**
interface Tunnel1000
  tunnel mpls traffic-eng autoroute announce
  tunnel mpls traffic-eng autoroute metric relative 0
	Autoroute announce with relative metric, 0 (default)

The above is then deployed to an already existing primary tunnel such as the Primary Tunnel Configlet (IOS), page 7-89.

Primary Tunnel Configlet (IOS XR)

Configuration

- Service: MPLS-TE Primary Tunnel
- Feature: MPLS TE configlet (IOS XR) for deploying a primary tunnel
- Device configuration: CISCO12406 with IOS XR 3.7.0.

Configlets

IOS Device Configuration	Comments
! **Explicit path:** explicit-path name isctmp12-isctmp7-1 index 1 next-address ipv4 unicast 10.163.25.109 index 2 next-address ipv4 unicast 10.163.25.106 !	Create an explicit path with the specified next addresses, which indicate the strict path that the tunnel traverses. This explicit path is used by the primary tunnel detailed above.
! **Primary tunnel:** interface tunnel-te133 description CISCO ISC-P2 ipv4 unnumbered Loopback0 priority 0 0 signalled-bandwidth 13 destination 192.168.118.214 fast-reroute path-option 1 explicit name isctmp12-isctmp7-1 path-option 2 dynamic record-route !	Create a TE primary tunnel with the following attributes: - Destination 192.168.118.214 - TE encapsulation - Setup priority 0 - Hold priority 0 - Reserve 13 kbps from global pool - Tunnel affinity 0x0 - Explicit first path option - Dynamic second path option - Enable FRR for the tunnel
mpls ldp interface tunnel-te 133 !	Enable ldp (Label Distribution Protocol) on the tunnel interface. This command is generated because the policy has the 'mpls ip' flag enabled. This allows the TE tunnels to be used for MPLS VPN traffic

Bandwidth Protection Backup Tunnel Configlet (IOS XR)

Configuration

- Service: MPLS-TE with FRR (Fast Re-Route)
- Feature: MPLS TE configlet (IOS XR) for deploying a backup tunnel
- Device configuration: CISCO12406 with IOS XR 3.7.0.

Configlets

IOS Device Configuration	Comments
! **Explicit path:** explicit-path name isctmp8-isctmp9-1 index 1 next-address ipv4 unicast 10.163.25.109 index 2 next-address ipv4 unicast 10.163.25.106 ! ! **Backup tunnel:** interface tunnel-te1009 description CISCO ISC-B1411 ipv4 unnumbered Loopback0 priority 0 0 backup-bw 9600000 destination 10.163.24.131 path-option 1 explicit name isctmp8-isctmp9-1 record-route affinity 0 mask 0 ! mpls traffic-eng interface POS0/1/0/1 backup-path tunnel-te 1009	Create an explicit path with the specified next addresses, which indicate the strict path that the tunnel traverses. This explicit path is used by the backup tunnel detailed above. Create a TE backup tunnel with the following attributes: - Destination 10.163.24.131 - TE encapsulation - Protect any pool bw of 9600000 kbps - Setup and hold priority of 0 - Tunnel affinity 0x0 - Explicit first path option

Connectivity Protection Backup Tunnel Configlet (IOS XR)

Configuration

- Service: MPLS-TE with FRR (Fast Re-Route)
- Feature: MPLS TE configlet (IOS XR) for deploying a connectivity protection backup tunnel and its associated exclude address path
- Device configuration: CISCO12406 with IOS XR 3.7.0.

Configlets

IOS Device Configuration	Comments
! **Explicit path:** explicit-path name L96-excl index 1 exclude-address ipv4 unicast 192.168.1.42 ! !	Create an explicit path with an exclude address, which indicates the IP address the path should avoid. This explicit path is used by the backup tunnel detailed above.

! **Backup tunnel:**
interface tunnel-te1000
  description CISCO ISC-B2
  ipv4 unnumbered Loopback0
  destination 10.52.96.37
  priority 0 0
  no signalled-bandwidth 0
  path-option 1 explicit name L96-excl
  affinity 0 mask 0
  backup-bw sub-pool unlimited
  record-route
!
mpls traffic-eng
  interface POS0/1/0/2
  backup-path tunnel-te 1000
!
	Create a TE backup tunnel with the following attributes:
	- Destination 10.52.96.37
	- TE encapsulation
	- Setup priority 0
	- Hold priority 0
	- Explicit first path option
	- Tunnel affinity 0x0
	- An unlimited sub pool acts as backup bandwidth
	
	Tunnel 1000 protects interface POS0/1/0/2

TE Traffic Admission Configlet Using PBTS (IOS XR)

Configuration

- Service: TE Traffic Admission
- Feature: MPLS TE configlet (IOS XR) for admitting traffic using Policy-Based Tunnel Selection (PBTS)
- Device configuration: CISCO12406 with IOS XR 3.7.0.

Configlets

IOS Device Configuration	Comments

`! TE Traffic Admission using PBTS:`
`interface tunnel-te133`
`  autoroute announce`
`  autoroute metric absolute 100`
`  policy-class 2`
`!`
	`Autoroute announce with absolute metric 100`

The above is then deployed to an already existing primary tunnel such as the Primary Tunnel Configlet (IOS XR), page 7-94.

TE Traffic Admission Configlet (IOS XR)

Configuration

- Service: TE Traffic Admission
- Feature: MPLS TE configlet (IOS XR) for TE Traffic Admission
- Device configuration: CISCO12406 with IOS XR 3.7.0

Configlets

IOS XR Device Configuration	Comments

`! TE Traffic Admission Using Static Route:`
`router static`
`  address-family ipv4 unicast`
`  1.2.3.4/32 tunnel-te 1000 123`
`  !`
`!`
	`Configuration of TE Traffic Admission`
	`on tunnel 1000 with static route`

The above is then deployed to an already existing primary tunnel such as the Primary Tunnel Configlet (IOS XR), page 7-94.

Warnings and Violations

This section lists warnings and violations that might be invoked when using the planning tools in Prime Fulfillment (computation engine).

Warnings and violations are tied in with the planning tools (see the Planning Tools section in the *Cisco Prime Fulfillment Theory of Operations Guide 6.2*). They are issued under the following circumstances:

- During an attempt to audit, place, repair, or groom a primary managed tunnel.
- During an attempt to protect selected network elements (links, routers, or SRLGs). Here, they help determine the cause of the failed protection (see Protection Planning, page 7-59).

  When the off-line backup route generation is called to determine if certain elements can be protected, the backup route generator responds for each element with either a set of tunnels that protect the element or a set of violations and warnings that help determine why the element could not be protected.

Note In the following, the term DirectedLink refers to a router interface.

This section contains the following:

Warnings

This class is characterized by all reports that are warnings. They are considered less severe than violations in the sense that they don't prevent the computation of a protection path.

Protection Computation Warnings

WarningFixVetoed

A fix of this element would have caused a neighbouring element to become unprotected. This fix is vetoed and no changes are proposed.

WarningRouterNotConformant

This element or any adjacent routers is/are not Protocol Conformant. It cannot therefore be protected.

Fields:

- Report Type—Name of report type.
- Description—Description of the problem signaled by the violation.
- Non-conformant router—Router that does not support traffic engineering.

WarningTunnelBandwidthQuotaTooSmall

The bandwidth of a backup tunnel that protects this element is below the minimum allowed bandwidth capacity.

Fields:

- Minimum allowed bandwidth quota—Minimum bandwidth allowed to protect the element in question.
- Actual tunnel bandwidth quota—Actual bandwidth of the backup tunnel.

WarningTunnelNumberTooLarge

There are too many backup tunnels for a flow through this element.

Fields:

- Maximum tunnel number allowed—Maximum number of tunnels allowed for a given network element.
- Actual Tunnel Count—Actual number of tunnels imposed on this network element.
- Flow:
  - Maximum Bandwidth—Maximum bandwidth for the traffic flow that needs to be protected.
  - Head Links—Protected interface for this flow.
  - Through Router —Protected device through which the regular traffic flow passes. If the protected element is a link, the Through Router field will not appear.
  - Tail Router—Hostname of destination (tail) router.
  - Type (NHop, NNHop)—Next hop type: NHOP for link (no through router) and NNHOP for node.

WarningZeroProtectedFlow

A flow through this element is protected by a backup tunnel, but has a maximum flow of zero.

Fields:

- Flow:
  - Maximum Bandwidth—Maximum available bandwidth on the element.
  - Head Links—Protected interface for this flow.
  - Through Router —Protected device through which the regular traffic flow passes. If the protected element is a link, the Through Router field will not appear.
  - Tail Router—Hostname of destination (tail) router.
  - Type (NHop, NNHop)—Next hop type: NHOP for link (no through router) and NNHOP for node.

Violations

This class is specialized by all reports that are violations. They are considered more "severe" than warnings because unlike warnigns, they will prevent the computation of a protection path.

Primary Placement Computation Violations

ViolationFrrProtectionInadequate

The FRR protection for a tunnel does not meet the specified protection level.

Fields:

- Report Type—Name of report type.
- Description—Description of the problem signaled by the violation.
- Required FRR Protection Level—Used to enable an MPLS traffic engineering tunnel to use a backup tunnel in the event of a link failure if a backup tunnel exists. Possible levels are **None**, **Best Effort**, **Link and SRLG**, and **Link, SRLG and Node.**
- Primary Tunnel:
  - Name—Tunnel identifier composed of a name and a tunnel number.
  - Head—Hostname of head router.
  - Tail—Hostname of destination (tail) router.
- Path—Tunnel Path
  - Node—Device hostname. Is only displayed if the protection level is "Link, SRLG & Node".
  - Protected (Node)—Indicates whether each node is protected (Yes) or not (No). Is only displayed if the protection level is ""“Link, SRLG & Node".
  - Link Label—IP addresses of the interfaces on the link.
  - Protected (Link)—Indicates whether each link is protected (Yes) or not (No).

ViolationInconsistentResourceAttributeChanges

A Topology-change attempts to modify one or more attributes on a resource causing a pair of its attributes to become inconsistent.

Fields:

- Report Type—Quality report, warning report, or violation report.

- Description—Description of the problem signaled by the violation.
- Resource—
  - Id—Id for head device or head interface representing the network resource.
  - Type—Resource device or interface.
- Attributes:
  - Attribute—Names of inconsistent attributes.
  - New Value—New attribute value proposed by user.

ViolationInconsistentTunnelAttributeChanges

A Tunnel-change attempts to modify one or more attributes on a tunnel causing a pair of its attributes to become inconsistent.

Fields:

- Report Type—Quality report, warning report, or violation report.
- Description—Description of the problem signaled by the violation.
- Tunnel:
  - Name—Tunnel identifier composed of a name and a tunnel number.
  - Head—Hostname of head router.
  - Tail—Hostname of destination (tail) router.
- Attributes:
  - Attribute—Names of inconsistent attributes.
  - New Value—New attribute value proposed by user.

ViolationLinkAffinityMismatch

A least one directed link in the path of a Primary Tunnel does not have attribute flags that match the affinity bits and mask of the Tunnel.

Fields:

- Report Type—Quality report, warning report, or violation report.
- Description—Description of the problem signaled by the violation.
- Primary Tunnel:
  - Name—Tunnel identifier composed of a name and a tunnel number.
  - Head—Hostname of head router.
  - Tail—Hostname of destination (tail) router.
  - Affinity Bits/Mask—Affinity bits and mask of the tunnel.
- Path—Name of tunnel path.
  - Outgoing Interface—Hostname/IP address of outgoing interface.
  - Attribute Flags—Links attributes to be compared to the tunnel's affinity bits. All have to be identical to have a valid path. The violation is triggered when at least one is different.

ViolationLinkPoolOversubscribed

The specified bandwidth pool for a directed link is over-subscribed by Primary Tunnels that pass through it.

Fields:

- Report Type—Quality report, warning report, or violation report.
- Description—Description of the problem signaled by the violation.
- Directed Link:
  - Head Device/Interface—Hostname for the head device and IP address of interface.
  - Tail Device/Interface—Hostname for the destination (tail) device or interface.
  - Pool—Global pool or sub pool.
  - Pool Bandwidth—The allocated global pool or sub pool bandwidth on the link.
- Primary Tunnel (table)—Specifies how many tunnels are using the link resource.
  - Name—Tunnel identifier composed of a name and a tunnel number.
  - Head—Hostname of head router.
  - Tail—Hostname of destination (tail) router.
  - Bandwidth—Total bandwidth of the tunnel.
  - Pool—Global pool or sub pool.
  - Path—Name of tunnel path.

ViolationMaxReRoutesExceeded

This number of Primary Tunnel re-routes in this solution exceeds the specified maximum.

Fields:

- Report Type—Quality report, warning report, or violation report.
- Description—Description of the problem signaled by the violation.
- Number of re-routes in solution—Number of re-routes proposed by the computation engine.
- Specified maximum number of re-routes—Maximum number of re-routes allowed.

ViolationNoPathInLayout

In the presence of other Primary Tunnels that have already been placed on the topology, no legitimate path is possible for a requested Primary Tunnel. Note: If a user requested path was specified this only means that the Primary Tunnel could not be placed on that requested path in the presence of other Primary Tunnels.

Fields:

- Report Type—Quality report, warning report, or violation report.
- Description—Description of the problem signaled by the violation.
- Requested Primary Tunnel:
  - Name—Tunnel identifier composed of a name and a tunnel number.
  - Head—Hostname of head router.
  - Tail—Hostname of destination (tail) router.
  - Bandwidth—Total bandwidth of the tunnel.
  - Requested Path—User-specified path for the tunnel.
  - Pool—Global pool or sub pool.

- FrrProtection—Possible protection levels are **None**, **Best Effort**, **Link and SRLG**, and **Link, SRLG and Node**.
- Propagation Delay—The time it takes for traffic to travel along a link from the head interface to the tail interface.
- AffinityBits/Mask—Affinity bits and mask of the tunnel.

ViolationNoPathInTopology

Irrespective of other Primary Tunnels placed upon the topology, no valid path is possible for a requested Primary Tunnel. Note: If a user requested path was specified this only means that the Primary Tunnel could not be placed on that requested path irrespective of other tunnels.

Fields:

- Report Type—Quality report, warning report, or violation report.
- Description—Description of the problem signaled by the violation.
- Requested Primary Tunnel:
  - Name—Tunnel identifier composed of a name and a tunnel number.
  - Head—Hostname of head router.
  - Tail—Hostname of (destination) tail router.
  - Bandwidth—Total bandwidth of the tunnel.
  - Requested Path—User-specified path for the tunnel.
  - Pool—Global pool or sub pool.
  - FrrProtection—Possible protection levels are **None**, **Best Effort**, **Link and SRLG**, and **Link, SRLG and Node**.
  - Propagation Delay (optional)—The maximum time allowed for traffic to travel along the requested path.
  - AffinityBits/Mask—Affinity bits and mask of the tunnel.

ViolationNoTunnelForDemand

No path implements a requested PrimaryTunnel, even though there exists a valid path in the network that this tunnel could take.

Fields:

- Report Type—Quality report, warning report, or violation report.
- Description—Description of the problem signaled by the violation.
- Requested Primary Tunnel:
  - Name—Tunnel identifier composed of a name and a tunnel number.
  - Head—Hostname of head router.
  - Tail—Hostname of destination (tail) router.
  - Bandwidth—Total bandwidth of the tunnel.
  - Requested Path—User-specified path for the tunnel.
  - Pool—Global pool or sub pool.
  - FrrProtection—Possible protection levels are **None**, **Best Effort**, **Link and SRLG**, and **Link, SRLG and Node**.

- Propagation Delay (optional)—The maximum time allowed for traffic to travel along the requested path.
- AffinityBits/Mask—Affinity bits and mask of the tunnel.

ViolationPathMismatch

A Primary Tunnel has a different path to that specified for it in the User Specified Path.

Fields:

- Report Type—Quality report, warning report, or violation report.
- Description—Description of the problem signaled by the violation.
- Primary Tunnel:
  - Name—Tunnel identifier composed of a name and a tunnel number.
  - Head—Hostname of head router.
  - Tail—Hostname of destination (tail) router.
  - Actual Path—Actual path of the tunnel associated with the violation.
  - Requested Path—User-specified path for the tunnel.

ViolationPathNotConnected

The path of a Primary Tunnel is not “connected”, that is. it does not form a connected sequence of admin-up links between the tunnel head and tail, or it contains loops.

Fields:

- Report Type—Quality report, warning report, or violation report.
- Description—Description of the problem signaled by the violation.
- Primary Tunnel:
  - Name—Tunnel identifier composed of a name and a tunnel number.
  - Head—Hostname of head router.
  - Tail—Hostname of destination (tail) router.
  - Path—Name of tunnel path.

ViolationPathUsesMissingLinks

A Tunnel-change attempts to create or modify a Tunnel so that its path or “User Requested Path” uses one or more directed links that do not exist in this topology.

Fields:

- Report Type—Quality report, warning report, or violation report.
- Description—Description of the problem signaled by the violation.
- Primary Tunnel:
  - Name—Tunnel identifier composed of a name and a tunnel number.
  - Head—Hostname of head router.
  - Tail—Hostname of destination (tail) router.
  - Change Type—Add Tunnel/Modify Tunnel.
  - Path Type—Requested/Actual.

  - Path—Name of tunnel path.
  - Outgoing Interface—Yes or No depending on whether a link is missing.
  - Incoming Interface—Yes or No depending on whether a link is missing.

ViolationPrimaryTunnelDelayTooLong

A Primary Tunnel has a propagation delay that is larger than the Maximum Propagation Delay specified for it.

Fields:

- Report Type—Quality report, warning report, or violation report.
- Description—Description of the problem signaled by the violation.
- Required Max Propagation Delay—The maximum time allowed for traffic to travel along the requested path.
- Primary Tunnel:
  - Name—Tunnel identifier composed of a name and a tunnel number.
  - Head—Hostname of head router.
  - Tail—Hostname of destination (tail) router.
  - Path—Name of tunnel path.
  - Actual Propagation Delay (table)—The time it takes for traffic to travel along each link in the entire path.

    - Link—Link segments in path.

    - Propagation Delay—Travel time for the traffic for each link segment.

ViolationResourceIdUnknown

A change attempts to remove or modify a resource (link, router or SRLG) with an Id, when no resource with that Id exists.

Fields:

- Report Type—Quality report, warning report, or violation report.
- Description—Description of the problem signaled by the violation.
- Resource to be removed:
  - Id—Id for head device or head interface representing the network resource.
  - Type—Resource device or interface.

ViolationTunnelIdInUse

A change attempts to add a Primary Tunnel with an Id that already exists.

Fields:

- Report Type—Quality report, warning report, or violation report.
- Description—Description of the problem signaled by the violation.
- Tunnel to Add:
  - Name—Tunnel identifier composed of a name and a tunnel number.
  - Head—Hostname of head router.
  - Tail—Hostname of destination (tail) router.

- Existing Tunnel:
  - Name—Tunnel identifier composed of a name and a tunnel number.
  - Head—Hostname of head router.
  - Tail—Hostname of destination (tail) router.

ViolationTunnelIdUnknown

A change attempts to remove or modify a Primary Tunnel with an Id when no tunnel with that Id exists.

Fields:

- Report Type—Quality report, warning report, or violation report.
- Description—Description of the problem signaled by the violation.
- Tunnel to Remove:
  - Id—Unique tunnel identifier used within Prime Fulfillment.

Protection Computation Violations

ViolationAggregateBandwidthOnLink

The bandwidth of backup tunnels for this element, which pass through the link, have a maximum bandwidth quota that exceeds the backup bandwidth of the link.

Fields:

- Required Bandwidth (due to tunnels)—Required bandwidth for the tunnels on the link.
- Link:
  - Backup Bandwidth—Total available bandwidth of the link.
  - Head Router—Hostname of the head router.
  - Head Interface—IP address of the head interface.
  - Tail Router—Hostname of destination (tail) router.
  - Tail Interface—IP address of the destination (tail) interface.
  - Label—IP addresses of the interfaces on the link.
  - Admin Status—Indicates whether the link is **Up** or **Down**.

ViolationBadBackupTunnel

The tunnel does not protect a flow over this element.

ViolationBandwidthProtectionMismatch

The tunnel backup bandwidth quotas of all the tunnels protecting a flow do not add up exactly to the maximum bandwidth of that flow.

Fields:

- Protected bandwidth—The protectable bandwidth of the protection path.
- Flow:
  - Maximum Bandwidth—Maximum available bandwidth on the element.
  - Head Links—Protected interface for this flow.

  - Through Router —Protected device through which the regular traffic flow passes. If the protected element is a link, the Through Router field will not appear.
  - Tail Router—Hostname of destination (tail) router.
  - Type (NHop, NNHop)—Next hop type: NHOP for link (no through router) and NNHOP for node.

ViolationLinkLevelTunnelDelayTooLarge

The delay of the backup tunnel is greater than that allowed.

Fields:

- Maximum allowed delay—Maximum delay allowed on the backup tunnel.
- Actual delay of tunnel—Actual delay of the backup tunnel.

ViolationNoBackupTunnels

There are no backup tunnels protecting this flow through the element.

Fields:

- Flow:
  - Maximum Bandwidth—Maximum available bandwidth on the element.
  - Head Links—Protected interface for this flow.
  - Through Router —Protected device through which the regular traffic flow passes. If the protected element is a link, the Through Router field will not appear.
  - Tail Router—Hostname of destination (tail) router.
  - Type (NHop, NNHop)—Next hop type: NHOP for link (no through router) and NNHOP for node.

ViolationPassesThroughSRLG

A backup tunnel is protecting a flow over this element that starts at a link within an Shared risk link group(SRLG). However that tunnel also passes through another link in the same SRLG.

Fields:

- Link:
  - Backup Bandwidth—Total available bandwidth of the link.
  - Head Router—Hostname of the head router.
  - Head Interface—IP address of the head interface.
  - Tail Router—Hostname of destination (tail) router.
  - Tail Interface—IP address of the destination (tail) interface.
  - Label—IP addresses of the interfaces on the link.
  - Admin Status—Indicates whether the link is **Up** or **Down**.
- SRLG—User-defined SRLG name.
- Flow:
  - Maximum Bandwidth—Maximum available bandwidth on the element.
  - Head Links—Protected interface for this flow.

- Through Router —Protected device through which the regular traffic flow passes. If the protected element is a link, the Through Router field will not appear.
- Tail Router—Hostname of destination (tail) router.
- Type (NHop, NNHop)—Next hop type: NHOP for link (no through router) and NNHOP for node.

ViolationUsesFailedElement

A backup tunnel that protects this element also uses it.

Document Type Definition (DTD) File

The Document Type Definition (DTD) file provides the rules required by the XML import file for importing bulk data into Prime Fulfillment.

For instructions on how to import tunnels into Prime Fulfillment, see Import Primary Tunnel, page 7-49.

This section includes the following:

DTD File

This is the DTD file provided with Prime Fulfillment.

```
<?xml version="1.0" encoding="UTF-8"?>

<!-- Data Definition for file based tunnel import -->

<!-- Import File Structure -->
<!ELEMENT IMPORT_DATA (TUN_ADD|TUN_CHANGE|TUN_DELETE|TUN_MIGRATE)+ >

<!-- Notes on attributes:
     importId:must be unique within the file,
          it is alphanumeric, must begin with alpha character,
          and no special character
     head, tail:hostname of valid TE enabled device
     policy:name of existing managed tunnel policy
     bw: must be numeric and values between 0-2147483647
     tnum:is the number portion of a tunnel interface
          E.g. for "interface tunnel3", use tnum="3"
          must be numeric and values between 0-65535
-->


<!-- Tunnel Add

   - #IMPLIED attributes are optional, if not specified, defaults to null
   - If tnum is not specified, system will generate tunnel number
   - To enable auto bandwidth, specify AUTOBW element
   - bw is required if autobw is not enabled
   - By default, tunnel will be created with a system path and a dynamic path
```

```
-->

<!ELEMENT TUN_ADD (AUTOBW?)>
<!ATTLIST TUN_ADD
         importId ID #REQUIRED
    head CDATA #REQUIRED
    tail CDATA #REQUIRED
    policy CDATA #REQUIRED
    bw CDATA #IMPLIED
    tnum CDATA #IMPLIED>


<!-- Tunnel Change

  - #IMPLIED attributes are optional, if not specified, value on existing
    tunnel is kept
  - To enable auto-bw, or to change auto-bw parameters, specify AUTOBW element
  - To disable auto-bw, set disableAutoBw="yes" and do not specify AUTOBW element
  - Existing tunnel path cannot be changed directly, setting reroutable="true"
    will enable system to reroute the tunnel if necessary

-->

<!ELEMENT TUN_CHANGE (AUTOBW?)>
<!ATTLIST TUN_CHANGE
         importId ID #REQUIRED
    head CDATA #REQUIRED
    tnum CDATA #REQUIRED
    policy CDATA #IMPLIED
    bw CDATA #IMPLIED
    disableAutoBw (yes) #IMPLIED
    reroutable (true|false) #IMPLIED>


<!-- Tunnel Delete

  - all attributes are required to identify tunnel to be deleted

-->

<!ELEMENT TUN_DELETE EMPTY>
<!ATTLIST TUN_DELETE
         importId ID #REQUIRED
    head CDATA #REQUIRED
    tnum CDATA #REQUIRED>


<!-- Tunnel Migrate

  - #IMPLIED attributes are optional, if not specified, value on existing
    tunnel is kept
  - All comments under Tunnel Change (above) applies to Tunnel Migrate
  - only unmanaged primary tunnel can be migrated
  - for tunnels with unmanaged tunnel policy, must specify a managed policy
  - for tunnels that was non-conformant:
         . if bw was zero, specify a new bw or enable auto-bw
         . if path was dynamic or non-conformant, the path options will be
           replaced with a system path and a dynamic path, and reroutable will
           be set to true.
  - reroutable attribute applicable only for tunnel that had a conformant first
         explicit path (i.e. explicit path with no loopback)

-->
```

```
<!ELEMENT TUN_MIGRATE (AUTOBW?)>
<!ATTLIST TUN_MIGRATE
         importId ID #REQUIRED
    head CDATA #REQUIRED
    tnum CDATA #REQUIRED
         policy CDATA #IMPLIED
         bw CDATA #IMPLIED
    disableAutoBw (yes) #IMPLIED
         reroutable (true|false) #IMPLIED>


<!-- Auto Bandwdith

  - #IMPLIED attributes are optional, if not specified, value is set to null
    for TUN_ADD and existing value is kept TUN_CHANGE
  - maxBw is required when used in TUN_ADD or if existing tunnel is not auto-bw
    enabled
  - minBw and maxBw must be numeric and values between 0-2147483647
  - maxBw must be greater than minBw if specified
  - freq must be numeric and values between 300-604800

-->

<!ELEMENT AUTOBW EMPTY>
<!ATTLIST AUTOBW
    freq CDATA #IMPLIED
    minBw CDATA #IMPLIED
    maxBw CDATA #IMPLIED>
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE IMPORT_DATA SYSTEM "TeImport.dtd">

<IMPORT_DATA>

<!--  Add New Managed Tunnel -->
<TUN_ADD importId="a1" head="isctmp3" tail="isctmp1" policy="mgdPolicy" bw="400" />
<TUN_ADD importId="a2" head="isctmp2" tail="isctmp9" policy="mgdPolicy" >
  <AUTOBW freq="300" minBw="100" maxBw="200"/>
</TUN_ADD>

<!--  Modify Existing Tunnel -->
<TUN_CHANGE importId="c1" head="isctmp2" tnum="200" bw="30" />
<TUN_CHANGE importId="c2" head="isctmp4" tnum="2" policy="mgdPolicy" reroutable="true"/>
<TUN_CHANGE importId="c3" head="isctmp5" tnum="46">
  <AUTOBW freq="300" minBw="100" maxBw="200"/>
</TUN_CHANGE>
<TUN_CHANGE importId="c4" head="isctmp2" tnum="200" bw="30" disableAutoBw="yes"/>

<!--  Delete Existing Tunnel -->
<TUN_DELETE importId="d1" head="isctmp3" tnum="45"/>

<!--  Migrate Tunnel -->
<TUN_MIGRATE importId="m1" head="isctmp2" tnum="3" policy="mgdPolicy"/>
<TUN_MIGRATE importId="m2" head="isctmp5" tnum="1" policy="mgdPolicy"/>

</IMPORT_DATA>
```

Example

The following is an example of a tunnel import XML file conforming to the DTD file specified in DTD File, page 7-108. It consists of a sample block for each of the Add, Change, Delete, and Migrate operations.

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE IMPORT_DATA SYSTEM "TeImport.dtd">

<IMPORT_DATA>

<!--  Add New Managed Tunnel -->
<TUN_ADD importId="a1" head="isctmp3" tail="isctmp1" policy="mgdPolicy" bw="400" />
<TUN_ADD importId="a2" head="isctmp2" tail="isctmp9" policy="mgdPolicy" >
  <AUTOBW freq="300" minBw="100" maxBw="200"/>
</TUN_ADD>

<!--  Modify Existing Tunnel -->
<TUN_CHANGE importId="c1" head="isctmp2" tnum="200" bw="30" />
<TUN_CHANGE importId="c2" head="isctmp4" tnum="2" policy="mgdPolicy" reroutable="true"/>
<TUN_CHANGE importId="c3" head="isctmp5" tnum="46">
  <AUTOBW freq="300" minBw="100" maxBw="200"/>
</TUN_CHANGE>
<TUN_CHANGE importId="c4" head="isctmp2" tnum="200" bw="30" disableAutoBw="yes"/>

<!--  Delete Existing Tunnel -->
<TUN_DELETE importId="d1" head="isctmp3" tnum="45"/>

<!--  Migrate Tunnel -->
<TUN_MIGRATE importId="m1" head="isctmp2" tnum="3" policy="mgdPolicy"/>
<TUN_MIGRATE importId="m2" head="isctmp5" tnum="1" policy="mgdPolicy"/>

</IMPORT_DATA>
```

CHAPTER 8

Managing Service Requests

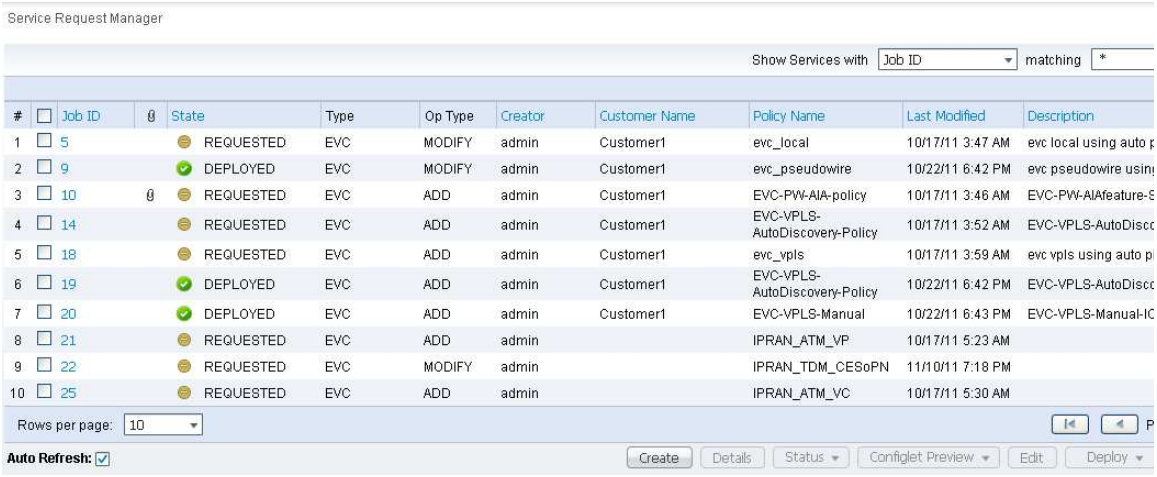
This chapter describes how to manage Prime Fulfillment service requests through the Service Request Manager window. It contains the following sections:

Accessing the Service Request Manager Window

To manage service requests, choose **Operate > Service Requests > Service Request Manager**.

Figure 8-1 shows the Service Request Manager window.

Figure 8-1 Service Request Manager Window



The Service Request Manager window shows the current list of service requests for this username. The window provides the following information about each service request:

- **JobID**—The job number assigned to the service request by Prime Fulfillment.
- **Data Files**—Shows if a data file is associated with the service request. A paper clip icon appears in the Data Files column if a service request has one or more templates associated with it. For more information about how templates and data files are used with service requests, see Chapter 9, "Managing Templates and Data Files."
- **State**—The transition state for the service request. See Service Request States, page 8-13 for more information.
- **Type**—The type of service request. For example, MPLS VPN, L2VPN, VPLS, VRF, TE, or EVC.
- **Operation Type**—The operation type for the service request. For example, ADD means that you are adding this service request, MODIFY that a service request has been changed from an earlier state, and DELETE that you are decommissioning this service request.
- **Creator**—Username identity of person who created or last modified the service request.
- **Customer Name**—Customer name for the service request.
- **Policy Name**—Name of policy assigned to this service request.
- **Last Modified**—Date and time the service request was created or last modified.
- **Description**—Optional text description of the service request.

You can use the buttons at the bottom of the Service Request Manager window to perform the following operations for service requests:

- **Create**—Create a Prime Fulfillment service request. See other chapters in this guide for more information on creating services requests for particular services.
- **Details**—View the service request history report, audit the service request, and view configlets. For more details, see Viewing Service Request Details, page 8-3.
- **Status**—View links and access any available logs for a selected service request. For more details, see Viewing the Status of Service Requests, page 8-8.
- **Configlet Preview**—Preview configlets that will be sent to a device for a specific service request. For more details, see Previewing Configlets, page 8-9.

- **Edit**—Edit a service request. For more details, see Editing Service Requests, page 8-10.
- **Deploy**—Deploy a service request. For more details, see Deploying Service Requests, page 8-10.
- **Decommission**—Decommission a service request. For more details, see Decommissioning Service Requests, page 8-12.
- **Delete**—Delete a service request. For more details, see Deleting Service Requests, page 8-13.

Viewing Service Request Details

The service request details include the link endpoints for the service request, the history, and the configlet generated during the service request deployment operation. Use the service request details to troubleshoot a problem or error with the service request or to check the commands in the configlet.

This section describes how to view the details of a service request, including the history, link details, and configlets.

To view service request detail, perform the following steps.

Step 1 Choose **Operate > Service Requests > Service Request Manager**.

Step 2 Select the service request and click **Details**.

The Service Request Details window appears.

From the Service Request Details page, you can view more information about:

- **Details > Links**—Service request links report.
- **Details > History**—Service request history report.
- **Details > Audit**—Not supported by Prime Fulfillment.
- **Details > Configlets**—View the Prime Fulfillment generated configlet for the service request.

The following sections describe the links, history, audit, and configlet details for a service request.

Viewing Service Request Link Details

The service request link details include the link endpoints, PE secured interface, VLAN ID, and whether a CE is present.

To see this information, perform the following steps.

Step 1 Click **Links** on the Service Request Details window.

The Service Request Link window appears.

Step 2 Choose a link and click **Details**.

The Service Request Link Details window appears.

Step 3 Click **OK** to return to the Service Request Link window.

Step 4 Choose another link to view, or click **OK** to return to the Service Request Details window.

Viewing Service Request History Information

To view history information about the service request, perform the following steps.

Step 1 Click **History** on the Service Request Details window.

The Service Request State Change Report window appears.

The history report shows the following information about the service request:

- **Element name**—The device, interface, and subinterfaces participating in this service request.
- **State**—The transition states the element has gone through.
- **Create Time**—The time the element was created for this service request.
- **Report**—The action taken by Prime Fulfillment for the element in this service request.

Step 2 Click **OK** to return to the Service Request Details window.

Viewing Audit Reports Service Requests

This section describes how to view configuration and functional audit reports for Prime Fulfillment service requests.

Viewing Configuration Audit Reports

A configuration audit verifies if all the commands for a service (service intent) are present on the network elements that participate in the service. Each time a service request is deployed in Prime Fulfillment, a configuration audit occurs. When a configuration audit occurs, Prime Fulfillment verifies that all Cisco IOS commands are present and that they have the correct syntax. An audit also verifies that there were no errors during deployment. If the device configuration does not match what is defined in the service request, the audit flags a warning and sets the service request to a Failed Audit or Lost state.

A configuration audit can fail if some of the commands are removed after provisioning from the network elements. This could happen if the commands are manually removed or they are removed as part of provisioning some other service. Another reason a configuration audit can fail is if Prime Fulfillment does not recognize commands in the configuration file. The default behavior in Prime Fulfillment is to skip unrecognized commands in the configuration file during the configuration audit. Such unrecognized commands might have be been present in an existing configuration or manually inserted in the configuration file. If an unrecognized command is at the start of a block of commands, Prime Fulfillment will skip the initial command and continue to parse the subcommands in the block. This might lead Prime Fulfillment to assume there is an error in the logic flow within the configuration file and cause the audit to fail.

Configuration audits can be performed manually through the Prime Fulfillment Task Manager. For information on how to create a task to manually schedule a configuration audit, see Task Manager, page 10-23.

To display the Configuration Audit report for a service request, perform the following steps.

Step 1 Choose **Operate > Service Requests > Service Request Manager**.

The Service Request Manager window appears.

Step 2 Choose a service request for the configuration audit.

Step 3 Click **Details**.

The Service Request Details window appears.

Step 4 Click the **Audit** button and choose **Config** from the drop-down list.

The Service Request Audit Report window appears.

This window lists the device name and role, and a message regarding the status of your configuration audit. If the audit is unsuccessful, the message field lists details on the failed audit. The audit failure message indicates missing commands and configuration issues. Carefully review the information in the message field. If the audit fails, you must correct all errors and redeploy the service request.

Step 5 Click **OK** to return to the Service Request Details window.

Viewing a Functional Audit Report

A functional audit verifies that the links in a service request or VPN are working correctly. The audit checks the routes to remote CEs in the VRF route tables on the PE devices. Prime Fulfillment automatically provides a functional audit whenever a service request is deployed or force-redeployed. A functional audit could fail if BGP peering is incorrect, MPLS setup in the core is faulty, or remote links are down.

Functional audits can be performed manually through the Prime Fulfillment Task Manager. For information on how to create a task to manually schedule a functional audit, see Task Manager, page 10-23.

To display the functional audit report for a service request, perform the following steps.

Step 1 Choose **Operate > Service Requests > Service Request Manager**.

The Service Request Manager window appears.

Step 2 Choose a service request for the functional audit.

Step 3 Click **Details**.

The Service Request Details window appears.

Step 4 Click the **Audit** button and choose **Functional** from the drop-down list.

The Service Request Audit Report window appears.

This window lists the device name and role, and a message regarding the status of your configuration audit. If the audit is unsuccessful, the message field lists details on the failed audit. The audit failure message indicates missing commands and configuration issues. Carefully review the information in the message field. If the audit fails, you must correct all errors and redeploy the service request.

Step 5 Click **OK** to return to the Service Request Details window.

Viewing Service Request Configlets

After you deploy the service request, Prime Fulfillment generates Cisco IOS or IOS XR commands to turn on appropriate services on all the network devices that participate in the service request.

Note For IOS devices, the configlets will appear as CLI commands. For IOS XR devices, the configlets can be viewed in XML or CLI format. For information about viewing configlets for IOS XR devices, see Viewing Configlets on IOS XR Devices, page 8-6.

To view the configlets that are generated, perform the following steps.

Step 1 Choose **Operate > Service Requests > Service Request Manager** to view the available service requests.

Step 2 Check the appropriate check box to select the service request for which you want to view the associated configlets.

Step 3 Click the **Details** button.

The Service Request Details window appears.

Step 4 Click the **Configlets** button.

The Service Request Configlets window appears. This window displays a list of devices for which configlets have been generated.

Step 5 To view configlets that were generated for a device, select a device and click the **View Configlet** button.

The Service Request Configlet window updates showing the configlet. By default, the latest generated configlet is displayed.

Step 6 If applicable, you can display configlets for a device based on the time of creation. Choose the desired time of creation in the Create Time list to display a specific configlet based on the time the configlet was generated for the service request.

Step 7 Click **OK** when you are finished viewing the configlet.

Viewing Configlets on IOS XR Devices

By default, service requests for IOS XR devices log the configuration sent to the device in XML format. Therefore, when configlets are viewed for IOS XR devices, they are displayed in raw XML format. Prime Fulfillment also allows the configlet to be viewed in CLI format. This feature is enabled by setting the DCPL property **DCS/getCommitCLIConfigAfterDownload** to true (the default setting).

Note The DCPL property **DCS/getCommitCLIConfigAfterDownload** must be set to true to display the configlet(s) in CLI format. On setting the DCPL property to true, CLI configlets will only be available for subsequent service request deployments. They will not be available for configlets that were deployed before the DCPL property was set to true.

To view the configlets for IOS XR devices in XML or CLI formats, or both, perform the following steps.

Step 1 Choose **Operate > Service Requests > Service Request Manager** to view the available service requests.

Step 2 Check the appropriate check box to select the service request for which you want to view the associated configlets.

Step 3 Click the **Details** button.

The Service Request Details window appears.

Step 4 Click the **Configlets** button.

The Service Request Configlets window appears. This window displays a list of devices for which configlets have been generated.

Step 5 To view configlets that were generated for an IOS XR device, select an IOS XR device and click the **View Configlet** button.

The Service Request Configlet window appears showing the configlet in CLI format. By default, the latest generated configlet is displayed.

Step 6 If applicable, you can display configlets for a device based on the time of creation. Choose the desired time of creation in the Create Time list to display a specific configlet based on the time the configlet was generated for the service request.

Step 7 To view the configlet in XML format, click the **XML Configlet** radio button.

The window refreshes and displays the configlet in XML format.

Step 8 To toggle among different formats, use the following radio buttons:

- **XML Configlet**—Displays the configlet in XML format.
- **CLI Configlet**—Displays the configlet in CLI format. This the default selection.
- **Both**—Displays the configlet side by side in both XML and CLI formats.

Step 9 Click **OK** when you are finished viewing the configlet.

Editing Configuration Files

To view or edit an existing router configuration file, perform the following steps.

Note Exercise caution when editing a configuration file, particularly if you then choose to make the edited file the running configuration file.

Step 1 Click the **Inventory > Physical Inventory > Devices**.

The Devices Inventory window appears.

Step 2 Check the check box next to the device name to choose the configuration file versions you want to view.

Step 3 Click **Config**.

The Device Configurations window appears.

The Device Configurations window displays the list of the current versions of the configuration files for the selected device. The configurations are listed by date and time. The configuration file listed first is the latest version.

Step 4 Choose the version of the configuration file you want to view, then click **Edit**.

The contents of the selected configuration file are displayed.

You can view or edit the displayed device configuration file.

Step 5 If necessary, edit the configuration file.

Step 6 When finished editing the file, click **Save**.

Viewing the Status of Service Requests

From the Service Request Manager window, you can obtain status information on a service request as detailed in the following sections.

Viewing Links

To view information about links associated with a service request, perform the following steps.

Step 1 Choose **Operate > Service Requests > Service Request Manager** to view the available service requests.

Step 2 Check the appropriate check box to select the service request for which you want to view the associated links.

Step 3 Click the **Status** button and choose **Links**.

The SR Link window appears.

This window displays a list of links associated with this service request.

Step 4 When you are finished reviewing the information, click the **Return to SRs** button.

Viewing Logs

To view logs associated with a service request, perform the following steps.

Step 1 Choose **Operate > Service Requests > Service Request Manager** to view the available service requests.

Step 2 Check the appropriate check box to select the service request for which you want to view the associated links.

Step 3 Click the **Status** button and choose **Logs**.

The Task Logs window appears.

This window displays the task by **Runtime Task Name**, and the **Action**, **Start Time**, **End Time**, and the **Status** of the task. You can use this window to view or delete the logs.

Step 4 To view the log, check the check box for the row that represents the task and click the **View Log** button.

The Task Log page appears.

It is possible to set the types of log level you want to view. Specify the Log Level and click **Filter** button to view that information you want to view.

Step 5 Click **Return to Logs** to specify another log to view.

Step 6 When you are finished reviewing the log information, click the **Close** button.

Previewing Configlets

The preview configlet operation allows you to preview the configlet(s) that will be sent to a device (or devices) for a selected service request before the device is actually provisioned. This allows you to check that the service request is generating the expected configlet(s), including any templates that may be applied.

Note the following caveats:

- The preview configlet preview feature is available to service requests in all states except In Progress and Closed.
- The preview configlet feature is not supported for TEM service requests.

To preview configlets for a service request, perform the following steps.

Step 1 Choose **Operate > Service Requests > Service Request Manager**.

Step 2 In the Service Request Manager window, select a service request and click **Configlet Preview**.

Step 3 Choose one of the following from the drop-down list:

- **For Deploy**—Generates the configlet(s) that would be generated by a Deploy operation.
- **For Force Deploy**—Generates the configlet(s) that would be generated by a Force Deploy operation.

These choices mimic the two different deployment options available, as the type of deployment may affect the configlet generated.

The Configlet Preview window appears. This window contains the generated configlets for each device in the service request.

Note This operation may take some time, as the configlet(s) must be uploaded from the device.

Step 4 After you review the configlets, click **OK** to return to the Service Request Manager window.

This feature is also available through the Deploy Service Request window, when a task is created to deploy the selected service request(s). Navigate to the Deploy Service Request window by choosing one or more service requests in the Service Request Manager window. Then choose **Deploy > Deploy** or **Deploy > Force Deploy**. The Deploy Service Request window that appears contains a table representing the selected service requests. Click on the Configlet Preview link in this table to show preview configlets.

Editing Service Requests

To edit a service request, perform the following steps.

Step 1 Choose **Service Operate > Service Requests > Service Request Manager**.

Step 2 Select the service request you want to modify and click **Edit**.

The Service Request Editor window appears.

Note The exact name and contents of this window will vary based on the type of service request being edited.

Step 3 Make the desired changes in the editor and click **Save**.

The Service Requests window reappears with the corresponding state of the service request set to In Progress and the Operation Type changed to Modify.

In order for the changes to be provisioned on the network, you must deploy the service request. For information about how to deploy a service request, see Deploying Service Requests, page 8-10. After deployment, look for the service request state to go to Deployed to indicate a successful deployment.

Deploying Service Requests

To apply policies to network devices, you must deploy the service request. When you deploy a service request, Prime Fulfillment compares the device information in the Repository (the Prime Fulfillment database) with the current device configuration and generates a configlet.

To apply device changes to network devices, you must deploy the service request. When you deploy a service request, Prime Fulfillment compares the device information in the Repository (the Prime Fulfillment database) with the current device configuration and generates a configlet.

Service Deployment

To deploy the service requests immediately or schedule their deployment, perform the following steps.

Step 1 Choose **Operate > Service Requests > Service Request Manager**.

The Service Requests Manager window appears.

Step 2 Check the check box next to the Job ID for the service request you want to deploy.

Step 3 Click the **Deploy** drop-down list.

You have two deployment options:

- Deploy: Use **Deploy** when the service request state is Requested or Invalid.
- Force Deploy: Use **Force Deploy** when the service request state is Deployed or Failed Audit.

Step 4 Choose **Deploy**.

The Deploy Service Request dialog box appears, which allows you to schedule when you want to deploy the selected service request.

Step 5 Complete the fields in this dialog box to schedule the service requested as needed.

Step 6 When satisfied with the schedule settings, click **Save**.

You return to the Service Request Manager window.

Check the Status display in the pop-up window at the lower corner of the window. If the service request has been deployed successfully, the Status display appears and shows a check in the Succeeded check box.

Step 7 To update the State from Requested to Deployed, check the **Auto Refresh** check box.

Note You can view logs to check on the task status and whether or not it completed successfully. For information on viewing logs, see Viewing Logs, page 8-8.

Monitoring Service Requests

To monitor a service request that is being deployed, you must use the task logs to help you troubleshoot why a service request has failed or to find more details about a service request.

To monitor a service request, perform the following steps.

Step 1 Choose **Operate > Tasks > Task Manager**.

The Task Logs window appears.

Step 2 Click **Find** to refresh the window.

The task that is executing will be the first in the list of tasks that are being performed in Prime Fulfillment.

Step 3 Choose the task you want to monitor and click **Logs**.

Step 4 Choose the run-time task that you want to monitor and click **View Log**.

Step 5 Choose the log level from the **Log Level** drop-down list and click **Filter**.

The log levels are All, Severe, Warning, Info, Config, Fine, Finer, and Finest.

Step 6 Click **Return to Logs**.

Step 7 Click **Close** in the Task Logs window.

Simulated Deployment of Service Requests

Simulate deploy is an additional option when deploying a service request. To use this feature, you must first set the DCPL property **Services\Common\allowSimulateDeploy** to true. When enabled, any service request that you can deploy by a standard deploy operation (for example, moving a service request from the Requested to Deployed state) can also be deployed in simulation mode. In a simulated deployment the provisioning flow proceeds as normal up to the point at which the configlet is to be downloaded to the device. For example, a live configuration will still be uploaded from the device. However, when downloading a configlet, Prime Fulfillment will act as if in echo mode (that is, the

configuration will not be downloaded to the actual device). In effect, this is echo mode on a per service request basis. Multiple deployment operations, both standard and simulated, can run concurrently using a mixture of echo-based transport and live device interactions.

To simulate deploy a service request, perform the following steps.

Step 1 Choose **Operate > Service Requests > Service Request Manager**.

The Service Requests Manager window appears.

Step 2 Check the check box next to the Job ID for the service request you want to deploy.

Step 3 Click the **Deploy** drop-down list.

Assuming the DCPL property **Services\Common\allowSimulateDeploy** has been set to true, you have three deployment options:

- Deploy
- Force Deploy
- Simulate Deploy

Step 4 Choose **Simulate Deploy**.

The Deploy Service Request dialog box appears, which allows you to schedule when you want to simulate deploy the selected service request.

Step 5 Complete the fields in this dialog box to schedule the service requested as needed.

Step 6 When satisfied with the schedule settings, click **Save**.

You return to the Service Request Manager window.

Check the Status display in the pop-up window at the lower corner of the window. If the service request has been deployed successfully, the Status display appears and shows a check in the Succeeded check box.

Step 7 To update the State from Requested to Deployed, check the **Auto Refresh** check box.

Note You can view logs to check on the task status and whether or not it completed successfully. For information on viewing logs, see Viewing Logs, page 8-8.

Decommissioning Service Requests

To decommission a service request, perform the following steps.

Step 1 Choose **Operate > Service Requests > Service Request Manager**.

Step 2 In the Service Request Manager window, select the service request you want to decommission and click **Decommission**.

The Confirm Decommission Service Request(s) window appears.

Step 3 Click **OK** to confirm the decommissioning of the service request.

The Service Request Manager window reappears with the corresponding Operation Type changed to Delete.

Step 4 Deploy the service request by selecting it and clicking **Deploy > Deploy**.

This is necessary for the changes to be provisioned to the network.

Step 5 In the Deploy Service Request window, select the time at which the deployment should take place (default is immediately), and click **Save**.

Step 6 After deployment, look for the service request state to go to Closed to indicate that the service request has been decommissioned successfully.

Deleting Service Requests

The Delete operation is designed to remove a service request from the repository without affecting the network.

To delete a service request, perform the following steps.

Step 1 Choose **Operate > Service Requests > Service Request Manager**.

Step 2 In the Service Request Manager window, select the service request you want to decommission and click **Delete**.

From the drop-down list choose one of the following:

- **Delete**—The regular delete can only be used on the service request in Closed state.

  Note The regular delete cannot be used on TE Resource, TE Tunnel, or TE Protection service requests because these cannot be decommissioned. These three types of service requests can only be force deleted.

- **Force Delete**—During force delete, the repository checks the necessary dependency on the service request before it can be deleted, so if a service request cannot be deleted, there will be an error message.

The Delete Service Request(s) window appears.

Step 3 Click **OK** to confirm the delete or force delete operation.

Service Request States

A service request transition state describes the different stages a service request enters during the provisioning process. For example, when you deploy a service request, Prime Fulfillment compares the device information in the Repository (the Prime Fulfillment database) with the current device configuration and generates a configlet. When the configlet is generated and downloaded to the device, the service request enters the Pending state. When the device is audited, the service request enters the Deployed state.

Prime Fulfillment service requests are processed in parallel, except when multiple service requests attempt to configure the same device. In this case, the service requests are processed sequentially (that is, only one write to the device can happen at a time).

Figure 8-2, "Service Requests States Transition Diagram," shows a high-level diagram of the relationships and movement among Prime Fulfillment service request states.

Figure 8-2 Service Requests States Transition Diagram

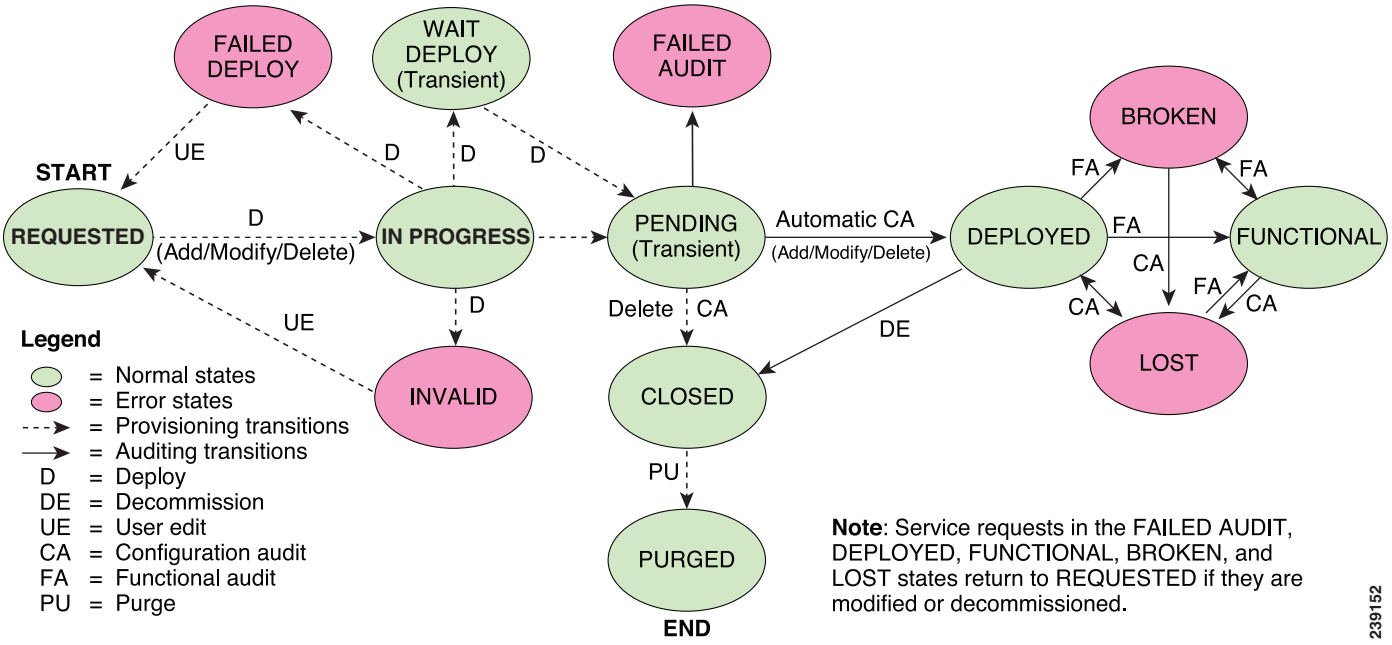


Table 8-1, "Summary of Prime Fulfillment Service Request States," describes the functions of each Prime Fulfillment service request state. They are listed in alphabetical order.

Table 8-1 Summary of Prime Fulfillment Service Request States

Service Request Type	Description
Broken (valid only for MPLS services)	The router is correctly configured but the service is unavailable (due to a broken cable or Layer 2 problem, for example). An MPLS service request moves to Broken if the auditor finds the routing and forwarding tables for this service, but they do not match the service intent.
Closed	A service request moves to Closed if the service request should no longer be used during the provisioning or auditing process. A service request moves to the Closed state only upon successful audit of a decommission service request. Prime Fulfillment does not remove a service request from the database to allow for extended auditing. Only a specific administrator purge action results in service requests being removed.
Deployed	A service request moves to Deployed if the intention of the service request is found in the router configuration file. Deployed indicates that the configuration file has been downloaded to the router, and the intent of the request has been verified at the configuration level. That is, Prime Fulfillment downloaded the configlets to the routers and the service request passed the audit process.

Table 8-1 Summary of Prime Fulfillment Service Request States (continued)

Service Request Type	Description
Failed Audit	This state indicates that Prime Fulfillment downloaded the configlet to the router successfully, but the service request did not pass the audit. Therefore, the service did not move to the Deployed state. The Failed Audit state is initiated from the Pending state. After a service request is deployed successfully, it cannot re-enter the Failed Audit state (except if the service request is redeployed).
Failed Deploy	The cause for a Failed Deploy status is that DCS reports that either the upload of the initial configuration file from the routers failed or the download of the configuration update to the routers failed (due to lost connection, faulty password, and so on).
Functional (valid only for MPLS services)	An MPLS service request moves to Functional when the auditor finds the VPN routing and forwarding tables (VRF) for this service and they match with the service intent. This state requires that both the configuration file audit and the routing audit are successful.
Invalid	Invalid indicates that the service request information is incorrect in some way. A service request moves to Invalid if the request was either internally inconsistent or not consistent with the rest of the existing network/router configurations (for example, no more interfaces were available on the router). The Provisioning Driver cannot generate configuration updates to service this request.
Lost	A service request moves to Lost when the Auditor cannot find a configuration-level verification of intent in the router configuration files. The service request was in the Deployed state, but now some or all router configuration information is missing. A service request can move to the Lost state only when the service request had been Deployed.
Pending	A service request moves to Pending when the Provisioning Driver determines that the request looks consistent and was able to generate the required configuration updates for this request. Pending indicates that the service request has generated the configuration updates and the configuration updates are successfully downloaded to the routers. The Auditor regards pending service requests as new requests and begins the audit. If the service has been freshly provisioned and not yet audited, it is not an error (pending audit). However, if an audit is performed and the service is still pending, it is in an error state.
Requested	If the service is newly entered and not yet deployed, it is not an error. However, if a Deploy is done and it remains Requested, the service is in an error state.

Table 8-1 Summary of Prime Fulfillment Service Request States (continued)

Service Request Type	Description
In Progress	Whenever a service request is requested for deployment, irrespective of its current state, it displays the In Progress state. The In Progress state is an intermediate state between Requested and Deployed. Whenever multiple service requests are concurrently requested for deployment, they all display the state of In Progress.
Wait Deploy	This service request state pertains only when downloading configlets using Cisco Configuration Engine. Wait Deploy indicates that the configlet has been generated, but it has not been downloaded because the device is not currently online. The configlet is staged in the repository until such time as the Cisco Configuration Engine notifies Prime Fulfillment that the device is up. Configlets in the Wait Deploy state are then downloaded to the device.

Table 8-2, "User Operations on Prime Fulfillment Service Requests," describes user operations and their impact on Prime Fulfillment service requests.

Table 8-2 User Operations on Prime Fulfillment Service Requests

User Operations	Description
Decommission	This user operation removes the service from all devices in the service request.
Force Deploy	This user operation allows you to Deploy a service request from any state except Closed. This is equivalent to restarting the state diagram. The service request can move from its current state to any other possible state. However, it does not move to the Requested state.
Force Delete	This user operation removes a service request from the database irrespective of its state. If you Force Delete a service request from the Prime Fulfillment repository before first decommissioning the service request, the service remains running on the network (specifically, the configuration remains on the devices on which the service was provisioned), but all record of the service request that created the service is removed from Prime Fulfillment.
Delete	When a service request is deleted, it is removed from the Prime Fulfillment database.

CHAPTER 9

Managing Templates and Data Files

This chapter explains the use of templates and data files in Prime Fulfillment. It contains the following sections:

Overview

Templates provide a means to deploy commands and configurations not normally supported by Cisco Prime Fulfillment to a device. Templates are written in the Velocity Template Language (VTL) and are generally comprised of IOS and IOS XR device CLI configurations.

Templates support the browsing, creation, and deletion of Template Folders, Templates, and Data Files and it supports the viewing of Template-generated configurations. This is applicable to both IOS and IOS XR. For IOS XR devices the configlet generated from template data files are CLI commands, not XML commands.

The configuration created from the template and data file can be downloaded to devices. When creating a Service Request, you can select from the list of templates and data files and associate them with the Service Request. At Deploy time, the template and data file are instantiated and the configuration is appended or prepended to the configlet generated by Prime Fulfillment. Another method is to use the Device Console feature to download templates independent of Service Requests, as explained in the "Download Template" section on page 13-3.

Prime Fulfillment provides a way to integrate a template with Prime Fulfillment configlets.

For a given customer edge router and/or provider edge router, you specify the following:

- template name
- template data file name

- whether the template configuration file should be appended or prepended to the Prime Fulfillment configlet
- whether the template configuration file is active or inactive for downloading to the edge device

The template data files are tightly linked with the corresponding template (a data file cannot be linked to more than one template). You can use a data file and its associated template to create a template configuration file. The template configuration file is merged with (either appended or prepended to) the Prime Fulfillment configlet. Prime Fulfillment downloads the combined Prime Fulfillment configlet and template configuration file to the edge device router.

- You can download a template configuration file to a router.
- You can apply the same template to multiple edge routers, assigning the appropriate template data file for each device. Each template data file includes the specific data for a particular device (for example, the management IP address or hostname of each device).

Template commands are treated independently from those associated with a service creation (Multi Protocol Label Switching (MPLS), Layer 2 Virtual Private Network (L2VPN), Virtual Private LAN Service (VPLS), Traffic Engineering (TE), and so on). Consequently, template commands must be removed separately from the device(s) during a service decommission. To remove prior template commands, a separate template is needed during a decommission process. Decommissioning a service request does not automatically remove the original template commands. A separate negate template needs to be added to the decommission process and the original templates must be removed. The negate template must contain the necessary NO commands to successfully remove any unwanted IOS commands added by the original template.

Summary of Template Manager Features

This section highlights key features of template and data file support in Prime Fulfillment, especially those that have an impact on working with policies and service requests.

Template Attributes

The Prime Fulfillment template mechanism allows you to differentiate templates by specifying (optional) attributes on a template, including:

- Device type
- Line card type
- Port type
- Software version (IOS or IOS XR)

These attributes are set through a drop-down list when setting up the template in Template Manager. Prime Fulfillment uses these attributes to automatically select the template/data file that most closely matches the device defined within the service request.

Associating Templates at the Policy Level

Prime Fulfillment supports the association of templates and data files in policies.

Selective Determination of Templates for U-PE and PE-AGG Device Roles

For added flexibility, Prime Fulfillment allows you to selectively apply templates to U-PE and PE-AGG devices (for example, in a ring environment) based on whether the devices have a UNI interface.

Enhanced Subtemplate Support

A new attribute in the Template Editor allows subtemplates to be associated with a template. Prime Fulfillment supports dynamic instantiation of subtemplates based on device attributes. While creating the subtemplates, values for these identifiers must be provided by the operator.

Dynamic Data File Creation

The user can create a data file during service request creation and associate it to the template copied from the associated policy. This functionality extends data file creation from the Template wizard to doing so directly from the service request wizard Template Association screen. In addition, you can modify any or all variables that are part of the template/date file attached to a service request and apply the updated template/data file without removing the entire service.

Automatic Application of Negate Templates

To remove a configuration created from a template/data file, a negate template must be applied to the existing service. This is no longer a manual process in Prime Fulfillment. You create both the positive and negate template. You can assign a positive template/data file to a policy. Prime Fulfillment calls the appropriate negate template at the appropriate time, as the negate template has a direct relationship with the deploy template. Prime Fulfillment determines which negate template to use, based on the service request action requested (for example, deploying or decommissioning a service). The negate template has the same name as the template, with the addition of the suffix .Negate. The negate template does not share the data file of the deploy template. The negate template must have its own data file defined.

Compatibility of the Template Mechanism with Previous Prime Fulfillment Releases

Prime Fulfillment maintains compatibility with the template mechanism in previous Prime Fulfillment releases. Templates created in earlier versions of Prime Fulfillment work “as is,” without any modifications to the templates or the workflow. In the case of a policy in the system that was created in an earlier Prime Fulfillment release, the GUI workflow for associating templates/data files is not visible. In such a case, the operator adds the template and data files during service deployment, as in previous releases of Prime Fulfillment.

Template Support for IOS and IOS XR

The template mechanism is supported for both IOS and IOS XR devices. For IOS XR devices, the configlet generated from templates/data files contains CLI commands and not XML statements. For IOS XR devices, template support is provided as CLI commands. For IOS devices, the operator can download a template configlet using the device console.

Note Note the following known issue in the case of IOS XR devices. When a service request is deployed with templates that contain improper or unsupported configurations, the service request still goes to the DEPLOYED state. This because the IOS XR device does not issue an error report on the improper configuration(s) deployed.

RBAC Support for Template Usage

Templates and data files are only accessible to users with the proper RBAC role. A permission type for data files has been added. The permissions allowed for the data files are view, create, modify, and delete. Operators cannot view templates/data files assigned to other roles, and are not permitted to deploy templates/data files to which they do not have access.

Template Variables

Template variables support most Prime Fulfillment repository variables for MPLS, L2VPN, VPLS, and FlexUNI/EVC. For a list of supported repository variables, see Summary of Repository Variables, page 9-33.

DCPL Properties

There are a few Dynamic Component Properties Library (DCPL) properties governing templates. These DCPL properties affect when a template is applied, whether negate templates are appended or prepended, whether templates are applied in the case when an service has multiple lines, only one of which have been edited, etc. For documentation on DPLC properties related to templates, see Appendix B, "Property Settings."

Importing and Exporting Templates

Prime Fulfillment provides a mechanism to import and export templates and data files. See Template Examples, page 9-32, for more information.

Template and Data File Workflow

This section summarizes the basic operations involved in setting up and using templates, data files, and negate templates in Prime Fulfillment.

Basic Template Manager Functions

- Create templates and negate templates for different configurations.
- Specify device attributes for the templates.
- Associate subtemplates to templates, if applicable
- Create data files for the subtemplates.
- Create a negate template for each subtemplate.
- Create data files for the negate templates.
- Create a super template and attach subtemplates to it.

These basic Template Manager functions are documented in other sections of this chapter.

Policy-Level Template Functions

- Create a policy and enable template support for the policy.
- Associate templates and (optionally) data files to the policy, if desired.

For information on how to associate templates and data files at the policy level, see the section Using Templates with Policies, page 9-21, in this chapter.

Service Request-Level Template Functions

Note When a policy is only associated with a template and no data file, then during creation of a service request using that policy, automatic selection of a data file for that template takes place, if the template has only one data file. If the template does not have a data file, then one must be created for that template and associated to the service request before saving is permitted.

- Create a service request and associate template(s) to a link.

- Deploy the service request on a device (for example, a 7600).
- The subtemplate and corresponding data file for the 7600 are autoselected for deployment.
- A configlet is generated from the subtemplate.
- Decommission the service request.
- The negate template for the subtemplate is autoselected and deployed.

For information on how to use templates and data files is service requests, see the section Using Templates and Data Files in the Service Request Workflow, page 9-28.

Basic Template and Data File Tasks

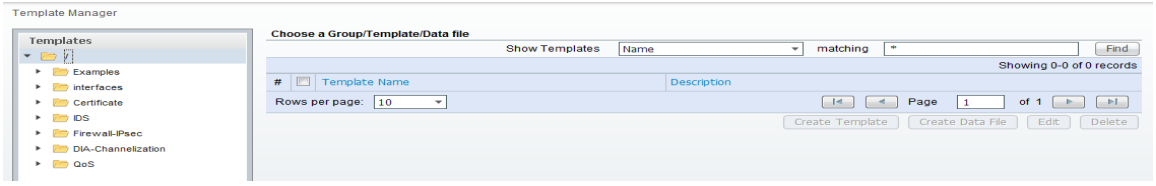
This section describes basic tasks you can perform with templates and data files. These include:

Viewing the Templates Tree and Data Pane

To use Templates, follow these steps:

Step 1 Choose **Service Design > Templates > Template Manager** and you receive a window as shown in Figure 9-1,

Figure 9-1 Templates Manager



The Templates tree is in the left column. You can continue clicking the **arrow** sign next to each created folder and subfolder until you get to the last level of information. The last possible level is the template name. Data file information is not kept in the tree.

The right section of the window is the data pane. The name of the folder or template is in the upper-left corner. When you check the check box next to the template or data file information, the **Create Template**, **Create Data File**, **Edit**, or **Delete** buttons are enabled as described in the following sections.

When there are many templates in a folder or many data files in a template, the **Show Templates matching** or **Show Data Files matching** filter in the upper right-hand corner of the data pane can be very useful. For example, you can click the drop-down list for **Show Templates** or **Show Data Files** and choose to match (matches are case-sensitive) the **Name** or **Description** and then in the **matching** box you can choose to work with templates or data files, respectively, that start with **abc**. In this case, enter **abc*** in the field and then click the **Show** button. Only the templates or data files, respectively, that start with **abc** appear. For more information about filters, see Filters, page 1-5.

Note The template search facility applies to the folder currently selected and not across all folders.

Note The data file search applies to the template currently selected and not across all folders and templates.

You can also **View** configurations when the table displays data files.

Step 2 Then you can do begin performing basic tasks with templates and data files, as described in the following sections.

Creating Folders and Subfolders

To create a new folder or subfolder, follow these steps:

Step 1 Choose **Service Design > Templates > Template Manager**.

Step 2 In the **Template Manager** tree, right-click in the white area and choose **New > Folder** to create a new folder or right-click on an existing folder or subfolder and choose **New > Folder** to create a subfolder.

Note There is no limit to the number of levels of folders and subfolders you can create.

Step 3 In the new text field that appears in the **Template Manager** tree, enter the new folder or subfolder name.

Copying Folders or Subfolders

To copy a folder or subfolder and paste it into another folder or subfolder, follow these steps:

Step 1 Choose a folder or subfolder and then right-click and you receive the opportunity to copy. Click **Copy**.

Step 2 Right-click on the folder or subfolder into which you want to paste the copied folder or subfolder and all its content and click **Paste**.

You will see the new folder or subfolder and all its content in the selected location. You can edit from there.

Creating Templates

You can either create a new template in an existing folder or you can create a new folder first and then create the template. To create a new folder, see the section "Creating Folders and Subfolders".

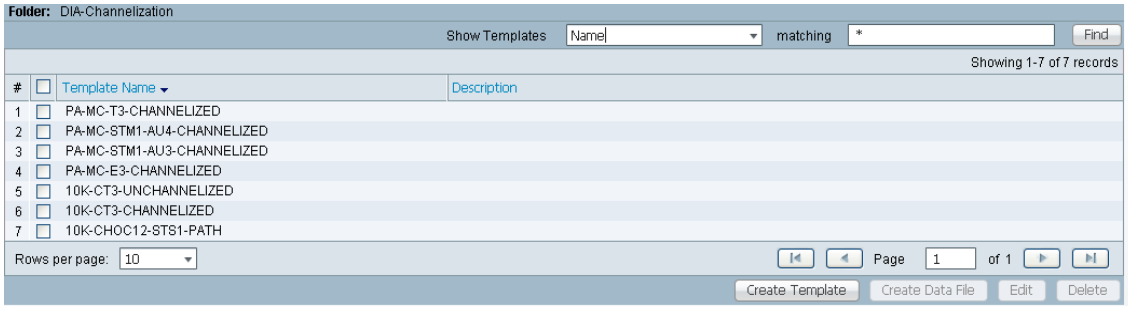
To create a new template, follow these steps:

Step 1 Choose **Service Design > Templates > Template Manager**.

Step 2 In the **Template Manager** tree, click on the folder in which you want to create a new template.

A window appears as shown in Figure 9-2.

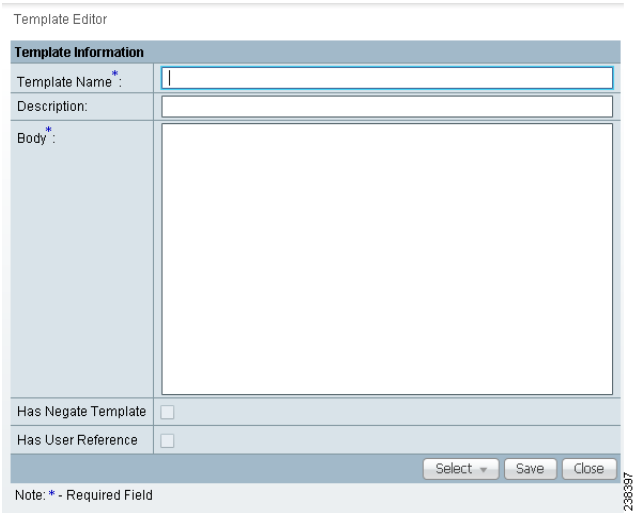
Figure 9-2 Folder with Existing Templates



Step 3 You can use the **Show Templates** drop-down list to choose whether to view the templates alphabetically by **Name** or by **Description**. Then click the **Show** button to activate how you view the templates. If you enter characters in the **matching** field before you click the **Show** button, you minimize the list of templates that appear either by **Name** or by **Description**. For more details, see Viewing the Templates Tree and Data Pane, page 9-5.

Step 4 Click the **Create Template** button and you receive a window as shown in Figure 9-3.

Figure 9-3 Template Editor



Step 5 Enter the following:

- **Template Name** (required)—This must be a unique name within a folder. This name must begin with an alphabetic character and can only contain alphanumeric characters, underscores, and hyphens.
- **Description** (optional) —You can enter any description here.
- **Body** (required)—Enter the configuration text, Velocity Template Language (VTL) directives, and variables that you want included.

Note The VTL is the mark-up language used to describe the template. The VTL is explained at http://velocity.apache.org. For more specific information, you might like to navigate to http://velocity.apache.org/engine/devel/user-guide.html or http://velocity.apache.org/engine/devel/vtl-reference-guide.html.

Step 6 Click the **Select** drop-down list, and choose from the following:

- Negate Template, page 9-8
- User Reference, page 9-9
- Optional Attributes, page 9-10
- Sub-Template, page 9-12
- Variables, page 9-13
- Validate, page 9-16

These tasks are described in the following subsections.

Negate Template

To remove a configuration created from a template or data file, you must apply **Negate** to the existing service. The negate template is saved as *<TemplateName>***.Negate** in the same folder as the original template. When a template is removed, the negate template is also deleted. You can also delete the negate template separately. Data files can be associated for the negate template.

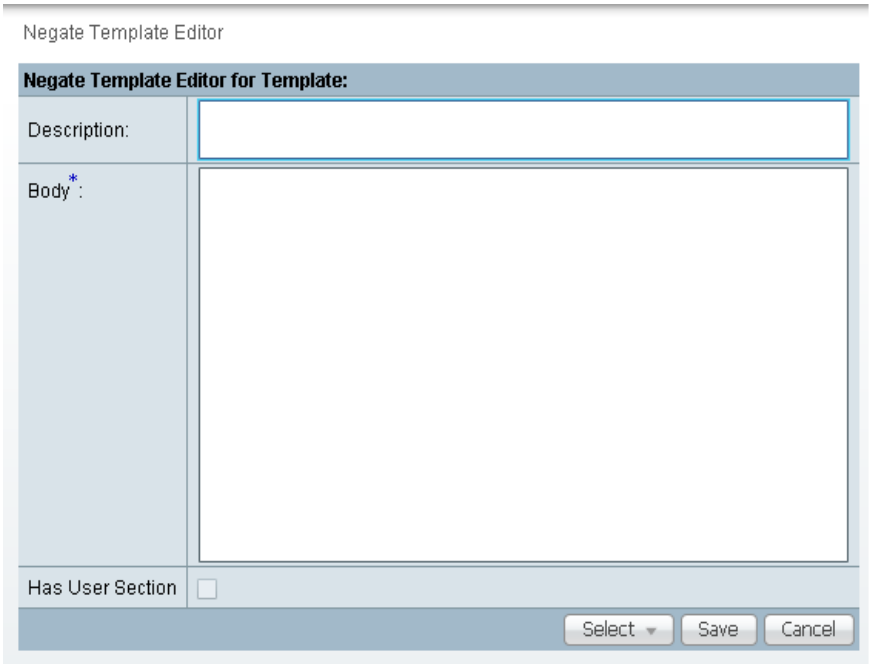
When a template is associated in a service Policy and Service Request, the negate template is automatically associated (see the *Cisco Prime Fulfillment User Guide 6.2*).

During decommissioning, a negate template is used for deployment. If you change a template, the negate template automatically changes to the negate template of the newly selected template.

Do the following after clicking the **Select** drop-down list in Step 6 of the "Creating Templates" section:

Step 1 Choose **Negate** and then click the **Go** button and you receive a window as in Figure 9-4.

Figure 9-4 Negate Template Editor



Step 2 Optionally add the name of the negate template in **Description**.

Step 3 Enter the template information in the required Body block. Enter **no** to indicate negate before each line of information, corresponding to the lines in the template.

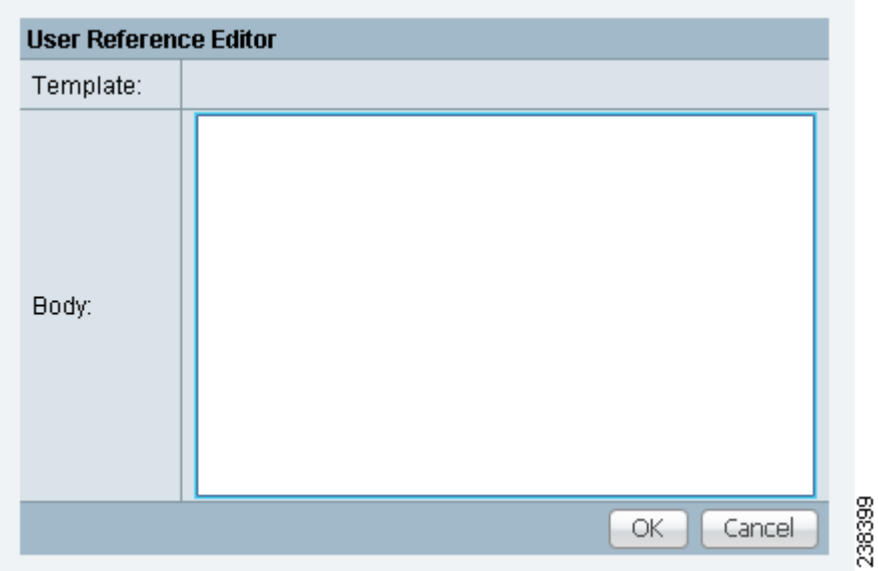
User Reference

You can keep information about this template by using **User Reference**.

Do the following after clicking the **Select** drop-down list in Step 6 of the "Creating Templates" section:

Step 1 Choose **User Reference** and then click the **Go** button and you receive a window as in Figure 9-5.

Figure 9-5 User Reference Editor



Step 2 In Figure 9-5, you can add information in the available fields, **Template** and **Body**.

Step 3 When you click the **OK** button, the information updates in Figure 9-3. When you click **Cancel**, you return to Figure 9-3 without updates.

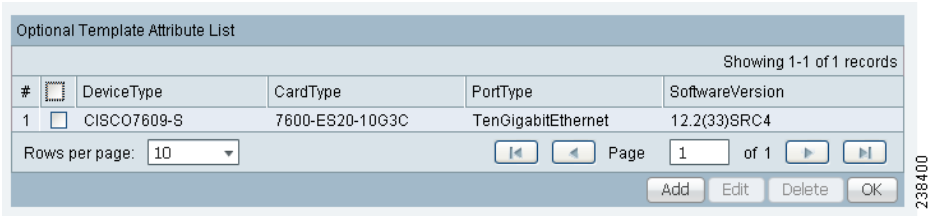
Optional Attributes

When you choose **Optional Attributes**, you can view the predefined **Device Type**, **Card Type**, **Port Type**, and **Software Version** (IOS and IOS XR) populated from the Prime Fulfillment repository. When no attribute value is provided for any of the four categories, the attribute is applicable for all in that type. For example, if the drop-down list for **Port Type** has no choices, the attribute value is applicable for all Port Types. Each combination of attributes should match. Each combination of attributes is called an attribute set, and templates can have multiple attributes, for example, a template can be applicable for the 7600 series and the 3500 series.

Do the following after clicking the **Select** drop-down list in Step 6 of the "Creating Templates" section:

Step 1 Choose **Optional Attributes** and then click the **Go** button and you receive a window as in Figure 9-6.

Figure 9-6 Optional Template Attribute List



Step 2 You can view the predefined **Device Type**, **Card Type**, **Port Type**, and **Software Version** (IOS and IOS XR) populated from the Prime Fulfillment repository. When no attribute value is provided for any of the four categories, the attribute is applicable for all in that type. Templates can have multiple attributes. You are required to create different templates based on roles and associate them to a Policy and Service Request (see the *Cisco Prime Fulfillment User Guide 6.2*).

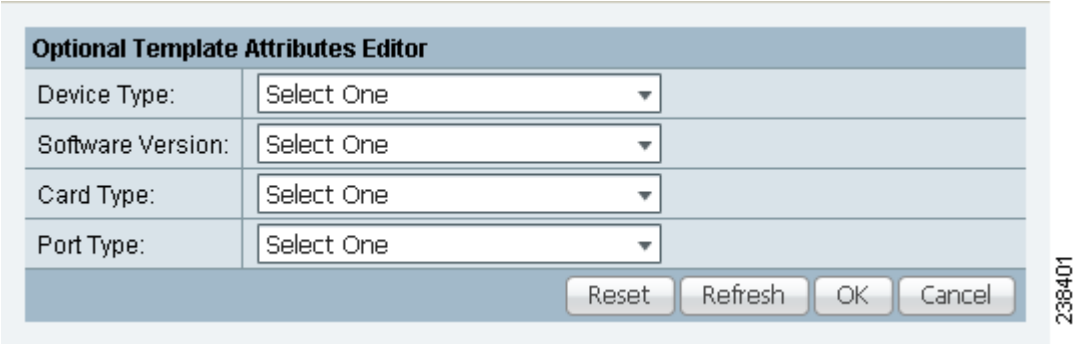
Step 3 Check the check box for the attribute set (row of information) for which you want to do the following (except for **Add**, when you should not check a check box):

- Click the **Add** button to open the optional attributes editor for adding attributes. The added attribute set is then reflected in the attribute list page.
- Click the **Edit** button to open the optional template attributes editor for modifying attributes. Multiple editing in one process is not allowed.
- Click the **Delete** button and the selected attributes are deleted. You can delete multiple selected attributes at the same time.
- Click the **OK** button and the window closes and you return to the previous page.

Step 4 When you click the **Add** or **Edit** button, a popup window appears in which you can enter the optional identifiers, as shown in Figure 9-7.

Note Before clicking the **Edit** button, you must check the check box for the one attribute set (row of information) in Figure 9-6 that you want to edit. You cannot edit multiple rows at the same time.

Figure 9-7 Optional Template Attributes Editor



Step 5 In Figure 9-7, click the drop-down list for each of **Device Type**, **Software Version**, **Card Type**, and **Port Type**.

Note The drop-down lists are intelligently filtered based on selection in the previous attribute. For example, if you have selected the 7600 for the **Device Type**, then the **Card Type** choices are related to the 7600.

Step 6 Click one of the following buttons:

- **Reset**—Allows you to start over in this selection process.
- **Refresh**—Refreshes the option list from the database and from the user-defined file. The user-defined attributes are read from the **usertemplateattr.xml** file.

Note The user-defined attribute file name **usertemplateattr.xml** can be changed by using the DCPL property: **TemplateManger\userTemplateAttrFile**. (See Appendix B, "Property Settings" for more details.)

Note The **Refresh** process can take some time. Just be aware of this.

- **OK**—Accepts your selected template attributes, adds them as a set, and returns you to an updated Figure 9-6 with an added attribute set (row of information).
- **Cancel**—Returns you to the previous window without any changes.

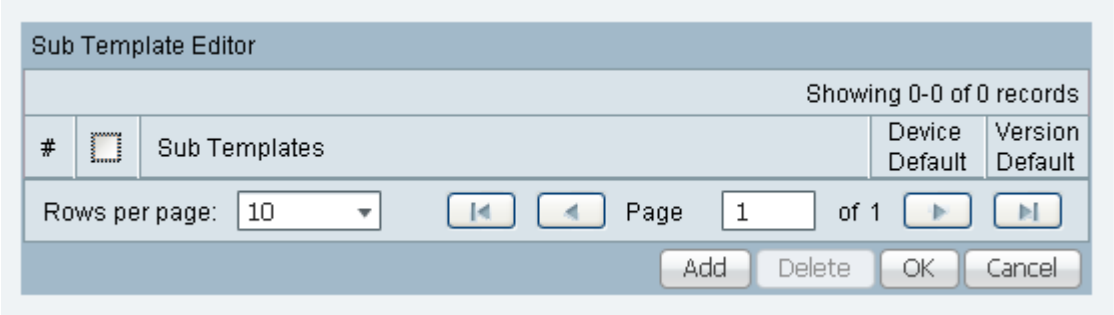
Sub-Template

A template using other templates is called a super-template. The template being used is called the sub-template. The super-template instantiates all required sub-templates by passing values for the variables in the sub-template. After instantiation, the super-template puts the sub-template generated configlet into the super-template.

Do the following after clicking the **Select** drop-down list in Step 6 of the "Creating Templates" section:

Step 1 Choose **Sub-Template** and then click the **Go** button and you receive a window as in Figure 9-8.

Figure 9-8 Sub-Template Editor



Step 2 Check the check box for the sub-template (row of information) for which you want to do the following (except for **Add**, when you should not check a check box):

- Click the **Add** button to add a new row. Then under the **Sub Templates** column, click **Add link** and a new pop-up appears from which you can choose the new subtemplates. Default check boxes are unselected. The changes are not persisted until saved by clicking the **Ok** button.
- Click the **Delete** button to delete selected rows. You can delete multiple selected rows at the same time. The changes are not persisted until saved by clicking the **Ok** button.
- Click the **OK** button and all changes will be saved on the form. The window closes and you return to the previous page.
- Click the **Cancel** button and all the changes are discarded. The window closes and you return to the previous page.

Step 3 You can associate a sub-template with a super-template. When the templates are instantiated during service provisioning (see the *Cisco Prime Fulfillment User Guide 6.2*), the appropriate sub-templates are used based on the run time information on the device, line card, role, port, and device software versions. Appropriate sub-template attributes provided by the user are instantiated during deployment based on the attributes. The following are some points to be aware of:

- Only one level of sub-template is supported, but there are no checks for depth of sub-templates.
- No validations occur to check if super-template and sub-template structure is cyclic.
- When you try to delete a sub-template that is referenced by a super-template, a warning message appears. You can modify a sub-template.
- Sub-templates can be attached to multiple super-templates.
- Data files are not supported for sub-templates. If multiple data files are found, the first available data file is chosen based on the alphabetic sorting during deployment.

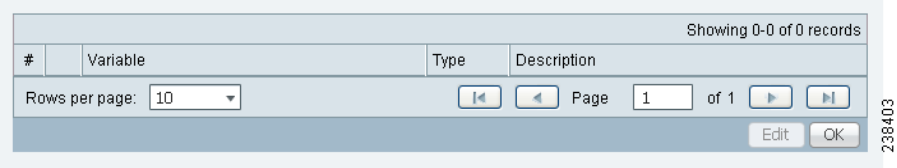
Step 4 You can mark a sub-template as default. There will be a default for the **Device** type and the **Software** version attribute types. When no attributes are marked for the templates, the template is treated as a default template. These templates have lower preference than default sub-templates for an attribute type. When multiple subtemplates have no attributes marked, no subtemplate is selected. For more infomation on using sub-templates, see Associating Subtemplates During Service Provisioning, page 9-25.

Variables

Do the following after clicking the **Select** drop-down list in Step 6 of the "Creating Templates" section:

Step 1 Choose **Variables** and then click the **Go** button and you receive a window as in Figure 9-9.

Figure 9-9 Template Variables



Step 2 Click the radio button for the Variable you want to edit and click **Edit**.

You receive a Variable Definition window.

Step 3 Click the drop-down list for **Type** to receive the following choices:

- **String**—Proceed to Step 4.
- **Integer**—Proceed to Step 5.
- **Float**—Proceed to Step 6.
- **IPv4 Address—**Proceed to Step 7.
- **Sub-Template**—Proceed to Step 8.

Step 4 The default Type to appear is **String**, a combination of ASCII characters considered as a group. The resulting Variable Definition window for Type String is shown and its attributes are as follows:

- **Description** (optional)—You can enter any descriptive statement about this variable here.

- **Required**—Leave the default of the checked check box if this variable is required. Otherwise, uncheck it.
- **Dimension**—Choose **0** (default), which indicates a scalar or enum variable; choose **1**, in which case the variable becomes a one-dimensional array; or choose **2**, in which case the variable becomes a two-dimensional array.
- **Pattern** (optional)—Specify a regular expression pattern of the string. For example, a pattern of **isc[0-9]+** defines a string that starts with **isc** followed by one or more digits from **0** to **9**.
- **Minimum Length** (optional) —f you specify a minimum length, the string cannot be less than the length specified here.
- **Maximum Length** (optional)—If you specify a maximum length, the string cannot exceed the length specified here.
- **Default** radio button (optional)—If there is a default value for the specified variable, specify it here.
- **Available Values** radio button (optional)—Enter string values for this variable. Separate the values by commas.

After you enter all the data, click **OK** to accept this information for the specified variable; continue editing all variables you want to change in this same way, then click **OK** in a window such as Figure 9-9, which now includes these updated variables; click **Save** and then **Close** or click **Close** and when asked, agree to **Save** for a window such as Figure 9-3. Create a Data File is shown in the "Creating Data Files" section on page 9-16, **Edit** is shown in the "Editing Templates and Data Files" section on page 9-19, and **Delete** is shown in the "Deleting Templates and Data Files" section on page 9-19.

Step 5 When you choose the Type **Integer**, a whole number, the resulting Variable Definition window for Type Integer is shown and its attributes are as follows:

- **Description** (optional)—You can enter any descriptive statement about this variable here.
- **Required**—Leave the default of the checked check box if this variable is required. Otherwise, uncheck it.
- **Dimension**—Choose **0** (default), which indicates a scalar or enum variable; choose **1**, in which case the variable becomes a one-dimensional array; or choose **2**, in which case the variable becomes a two-dimensional array.
- **Minimum Value** (optional)—If you specify a minimum value, the integer cannot be less than the value specified here.
- **Maximum Value** (optional)—If you specify a maximum value, the integer cannot exceed the value specified here.
- **Default** radio button (optional)—If there is a default value for the specified variable, specify it in the field after the radio button.
- **Available Values** radio button (optional)—Enter string values for this variable in the field after the radio button. Separate the values by commas.

After you enter all the data, click **OK** to accept this information for the specified variable; continue editing all variables you want to change in this same way, then click **OK** in a window such as Figure 9-9, which now includes these updated variables; click **Save** and then **Close** or click **Close** and when asked, agree to **Save** for a window such as Figure 9-3. Create a Data File is shown in the "Creating Data Files" section on page 9-16, **Edit** is shown in the "Editing Templates and Data Files" section on page 9-19, and **Delete** is shown in the "Deleting Templates and Data Files" section on page 9-19.

Step 6 When you choose the Type **Float**, a number that has no fixed number of digits before or after the decimal point, the resulting Variable Definition window for Type Float is shown and its attributes are as follows:

- **Description** (optional)—You can enter any descriptive statement about this variable here.

- **Required**—Leave the default of the checked check box if this variable is required. Otherwise, uncheck it.
- **Dimension**—Choose **0** (default), which indicates a scalar or enum variable; choose **1**, in which case the variable becomes a one-dimensional array; or choose **2**, in which case the variable becomes a two-dimensional array.
- **Minimum Value** (optional)—If you specify a minimum value, the floating point value cannot be less than the value specified here.
- **Maximum Value** (optional)—If you specify a maximum value, the floating point value cannot exceed the value specified here.
- **Default** radio button (optional)—If there is a default value for the specified variable, specify it here.
- **Available Values** radio button (optional)—Enter string values for this variable. Separate the values by commas.

After you enter all the data, click **OK** to accept this information for the specified variable; continue editing all variables you want to change in this same way, then click **OK** in a window such as Figure 9-9, which now includes these updated variables; click **Save** and then **Close** or click **Close** and when asked, agree to **Save** for a window such as Figure 9-3. Create a Data File is shown in the "Creating Data Files" section on page 9-16, **Edit** is shown in the "Editing Templates and Data Files" section on page 9-19, and **Delete** is shown in the "Deleting Templates and Data Files" section on page 9-19.

Step 7 When you choose the Type **IPv4 Address**, the resulting Variable Definition window for Type IPv4 Address is shown and its attributes are as follows:

- **Description** (optional)—You can enter any descriptive statement about this variable here.
- **Required**—Leave the default of the checked check box if this variable is required. Otherwise, uncheck it.
- **Dimension**—Choose **0** (default), which indicates a scalar or enum variable; choose **1**, in which case the variable becomes a one-dimensional array; or choose **2**, in which case the variable becomes a two-dimensional array.
- **Subnet Mask** (optional)—Enter a valid subnet mask.
- **Class** (optional)—Enter the class of the IP address. The options are: **Undefined**, **A**, **B**, or **C**.
- **Default** radio button (optional)—If there is a default value for the specified variable, specify it here.
- **Available Values** radio button (optional)—Enter string values for this variable. Separate the values by commas.

After you enter all the data, click **OK** to accept this information for the specified variable; continue editing all variables you want to change in this same way, then click **OK** in a window such as Figure 9-9, which now includes these updated variables; click **Save** and then **Close** or click **Close** and when asked, agree to **Save** for a window such as Figure 9-3. Create a Data File is shown in the "Creating Data Files" section on page 9-16, **Edit** is shown in the "Editing Templates and Data Files" section on page 9-19, and **Delete** is shown in the "Deleting Templates and Data Files" section on page 9-19.

Step 8 When you choose the Type **Sub-Template**, you instantiate one subtemplate into the Main template. The resulting Variable Definition window for Type Sub-Template is shown and its attributes are as follows:

- **Description** (optional)—You can enter any descriptive statement about this variable here.
- **Required**—Leave the default of the checked check box if this variable is required. Otherwise, uncheck it.
- **Location** (required)—Enter the full path name of the parent template. For example **/test2/testyy**.

The variable varName is defined as the subtemplate type (by selecting **Variables** and clicking **Go**). The Sub-Template defined earlier is called and you must provide the subtemplate path. The syntax is as follows:

$<*varName*>.callWithDatafile(<*DatafileName*>)

After you enter all the data, click **OK** to accept this information for the specified variable; continue editing all variables you want to change in this same way, then click **OK**, which now includes these updated variables; click **Save** and then **Close** or click **Close** and when asked, agree to **Save** for a window such as Figure 9-3. Create a Data File is shown in the "Creating Data Files" section on page 9-16, **Edit** is shown in the "Editing Templates and Data Files" section on page 9-19, and **Delete** is shown in the "Deleting Templates and Data Files" section on page 9-19.

Validate

To validate the information you entered in Figure 9-3 (see Step 5), do the following after clicking the **Select & Click Go** drop-down list in Step 6 of the "Creating Templates" section:

Step 1 Choose **Validate** and then click the **Go** button.

Step 2 For a successful validation, you will receive a information window appears.

Creating Data Files

You can create a new data file from an existing template. If the template you want is not available, go to the "Creating Templates" section on page 9-7.

To create a data file, follow these steps:

Step 1 Choose **Service Design > Templates > Template Manager**.

Step 2 In the **Template Manager** tree in the left part of your window, do one of the following

1. Left-click on the folder or subfolder in which the template for which you want to create a data file exists or
2. Click on the arrow next to the folder of choice and then click on the template for which you want to create a data file.

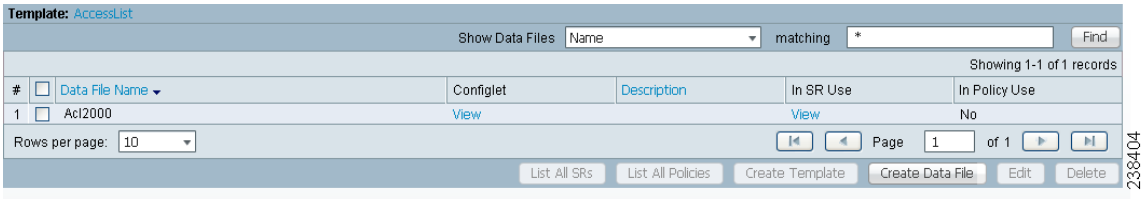
Step 3 If you chose 1. in Step 2, a window appears as shown in Figure 9-2.

Check the check box for the template for which you want to create a data file and click **Create Data File**. Then proceed to .

Otherwise, proceed to Step 4.

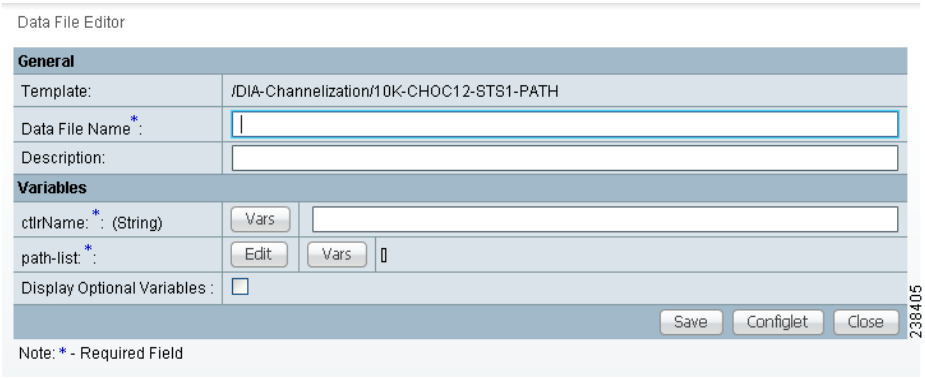
Step 4 If you chose 2. in Step 2, the buttons appear as shown in Figure 9-10.

Figure 9-10 Choose Existing Template, Another Way



Click **Create Data File**. An example of a window that appears is shown in Figure 9-11.

Figure 9-11 Template Data File Editor



Step 5 In the **General** area, fill in the following:

- **Data File Name** (required)—This must be a unique name. This name must begin with an alphabetic character and can only contain alphanumeric characters and the underscore.
- **Description** (optional)—Enter any description that helps you identify this data file.

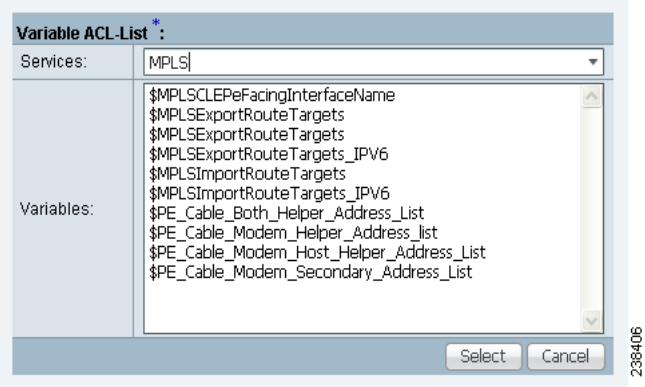
In the example in Figure 9-11, in the **Variables** area, **cntrlName** is a string variable (**Dimension** defined when the template was created was **0**); you can also create a one-dimensional array (**Dimension** defined when the template was created was **1**); and **t1-list** is a two-dimensional array (**Dimension** defined when the template was created was **2**).

If **t1-list** is a Dynamic Java Class variable, you *must* enter the entire Java Class package name. For example: com.cisco.isc.class_name.

Note **cntrlName** can *only* be a string variable.

Step 6 If you click **Vars** as shown in Figure 9-11, you receive a window as shown in Figure 9-12.

Figure 9-12 Template Data File Editor



Click the **Services** drop-down list to have access to variables for:

- **MPLS**
- **L2VPN**
- **VPLS**
- **VRF**
- **FlexUNI**

Then click the entry in **Variables** that you want to use and click **Select**.

If you have a **0** dimensional entry (set as **Dimension 0** when creating a template), you can only enter variables in the provided field.

Step 7 When you click **Edit**, as shown in Figure 9-11, the resulting window depends on whether you are editing a **1** or **2** dimensional array.

Proceed to Step 8 for information about a **1** dimensional array.

Proceed to Step 11 for information about a **2** dimensional array.

Step 8 For a one-dimensional array (set as **Dimension 1** when creating the template), when you click **Edit**, you receive a window.

Step 9 To add a variable, click **Add** and a window appears in which you can add the variable. Then click **OK.**

Step 10 To edit or delete a variable, highlight the variable and click **Edit** or **Delete**. For **Edit**, you receive a window appears. Then click **OK**. For **Delete**, *be sure* you want to delete. After you click **Delete**, it automatically occurs and the window is updated. Proceed to Step 16.

Step 11 For a two-dimensional array (set as **Dimension 2** when creating the template), when you click **Edit**, you receive a window appears.

Step 12 Click **Add Row** and a window appears. Enter a value and click **OK**.

Step 13 Click **Add Column** and a window appears.

Step 14 Enter a value and click **OK**. A resulting window appears.

Step 15 You can check any of the check boxes (toggles) and you can then **Edit** or **Delete** that row or column. You can also continue to **Add Row** and **Add Column** as shown in Step 13 and Step 14, respectively.

Step 16 When you complete setting up your two-dimensional array, click **OK**. A window as shown in Figure 9-11 is updated to reflect the new data file information.

Step 17 You can then click **Save** and then **Close** to save this information and close this file; click **Configure** to show the configuration file; or click **Close** and then be sure to click **OK**, if you want to save the information you have created. If you do not want to save this information, click **Close** and then click **Cancel**.

Editing Templates and Data Files

To edit a Template or Data File, follow these steps:

Step 1 Choose **Service Design > Templates > Template Manager**.

Step 2 In the **Template Manager** tree, left-click on the folder or subfolder in which the template you want to edit exists or the template in which the data file you want to edit exists. Alternatively, when the name in the upper left corner of the data pane is a template, you can click on the template name to edit the template.

To edit a template, a window appears as shown in Figure 9-2. To edit a data file, a window appears as shown in Figure 9-10.

Step 3 You can use the **Show Templates** or **Show Data Files** drop-down list to choose whether to view the templates or data files alphabetically by **Name** or by **Description**. Then click the **Show** button to activate how you view the templates or data files. If you enter characters in the **matching** field before you click the **Show** button, you minimize the list of templates or data files that appear either by **Name** or by **Description**. For more details, see the **Show Templates matching** or **Show Data Files matching** filter in the upper right-hand corner of the data pane can be very useful. For example, you can click the drop-down list for **Show Templates** or **Show Data Files** and choose to match (matches are case-sensitive) the **Name** or **Description** and then in the **matching** box you can choose to work with templates or data files, respectively, that start with **abc**. In this case, enter **abc*** in the field and then click the **Show** button. Only the templates or data files, respectively, that start with **abc** appear. For more information about filters, see Viewing the Templates Tree and Data Pane, page 9-5.

Step 4 Check the check box for the template or data file you want to edit.

Note For a data file, there is a **Configlet** column in which you can click **View** to view the configuration file.

Step 5 Click **Edit**.

Step 6 When editing a template, you receive a window as shown in Figure 9-3. Then proceed as in Step 5 in the Creating Templates section. When editing a data file, you receive a window as shown in Figure 9-10. Then proceed as in in the Creating Data Files section.

Deleting Templates and Data Files

To delete a Template or Data File, follow these steps:

Step 1 Choose **Service Design > Templates > Template Manager**.

Step 2 In the **Templates** tree, left-click on the folder or subfolder in which the template you want to delete exists or the template in which the data file you want to delete exists.

To delete a template, a window appears as shown in Figure 9-2. To delete a data file, a window appears as shown in Figure 9-10.

Step 3 You can use the **Show Templates** or **Show Data Files** drop-down list to choose whether to view the templates or data files alphabetically by **Name** or by **Description**. Then click the **Show** button to activate how you view the templates or data files. If you enter characters in the **matching** field before you click the **Show** button, you minimize the list of templates or data files that appear either by **Name** or by **Description**. For more details, see the **Show Templates matching** or **Show Data Files matching** filter in the upper right-hand corner of the data pane can be very useful. For example, you can click the drop-down list for **Show Templates** or **Show Data Files** and choose to match (matches are case-sensitive) the **Name** or **Description** and then in the **matching** box you can choose to work with templates or data files, respectively, that start with **abc**. In this case, enter **abc*** in the field and then click the **Show** button. Only the templates or data files, respectively, that start with **abc** appear. For more information about filters, see Viewing the Templates Tree and Data Pane, page 9-5.

Step 4 Check the check box for the template or data file you want to delete.

Note For a data file, there is a **Configlet** column in which you can click **View** to view the configuration file.

Step 5 Click the **Delete** button.

A confirmation window appears prompting you to confirm the deletion. Before deleting a data file, make sure it is not associated with a service request, by checking that the **In SR Use** column is set to **No**. When deleting a folder or a template, make sure that none of the data files they contain are associated with a service request. By clicking **OK**, you continue the deletion, and by clicking **Cancel**, you cancel the deletion.

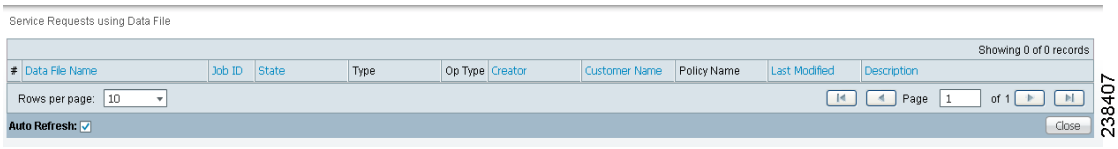
You receive an updated window as shown in Figure 9-2, or Figure 9-10, with the deleted template or data file no longer available.

Listing Service Requests Associated with a Data File

In the **In SR Use** column, as shown in Figure 9-10, **Yes** indicates that the data file is in use and **No** indicates that the data file is not in use. If **Yes** appears, you can click on it and you receive a list of all the associated service requests. If **Yes** appears, a **List All SRs** button is enabled in the bottom row. If you click the **List All SRs** button, all the service requests associated with the selected data file(s) appears, as shown in Figure 9-13. If **No** appears in the **In SR Use** column, the **List All SRs** button is disabled.

From Figure 9-13, if you click the **Close** button, the previous window appears.

Figure 9-13 List All SRs



Note The only data files listed in the **Data File Name** column are those selected previously by the user to get to this window. The service request might be associated with other data files that are not displayed.

Listing Policies Associated with a Data File

In the **In Policy Use** column, as shown in Figure 9-13, **Yes** indicates that the data file is in use and **No** indicates that the data file is not in use. If **Yes** appears, you can click on it and you receive a list of all the associated policies. If **Yes** appears, a **List All Policies** button is enabled in the bottom row. If you click the **List All Policies** button, all the policies associated with the selected data file(s) appears. If **No** appears in the I**n Policy Use** column, the **List All Policies** button is disabled.

If you click the **Close** button for the newly created window, the previous window appears.

Note The only data files listed in the **Data File Name** column are those selected previously by the user to get to this window. The policy might be associated with other data files that are not displayed.

Using Templates with Policies

This section provides information on how to enable template support and associate templates/data files with Prime Fulfillment policies. It contains the following sections:

Overview

Prime Fulfillment supports associating templates/data files to a service policy. This minimizes steps in the provisioning workflow and also reduces potential errors that can occur if an incorrect template/data file is selected during service creation. In the Policy Editor workflow, after the policy attributes are set, a new Templates Association window appears. The Enable Templates check box that appears in this window allows you to enable template association for the policy and to specify templates/data files to be available for service requests based on the policy. More than one template/data file can be associated to the policy. Each template/data file can be associated to a device role. The available device roles are determined by the policy type. In the case of U-PE and PE-AGG device roles, templates/data files can be selectively determined based on whether the device has a UNI interface. Later, at the time of service request creation, templates are only available if the device type matches the role type specified for the template within the policy or role type along with (or without) the presence of UNI interface in the policy.

Associating Templates and Data Files to a Policy

This section describes how to associate templates and data files to an Prime Fulfillment policy. These features also apply in the case of editing a policy.

After the policy attributes are set for a policy, the Template Association window appears in the workflow.

This window is where you associate the templates/data files as a final step before clicking the Finish button and saving the policy settings.

To associate template(s)/data file(s) with the policy, perform the following steps.

Step 1 Check the **Template Enable** check box to enable template use in service requests based on this policy. This check box is unchecked by default.

The GUI updates with fields allowing you to associate templates/data files to the policy.

Step 2 Click the **Add** button to add a row in which to specify associated templates/data files.

A new row appears in the GUI, providing fields to set the role type, specify templates/data files, and specify if the template/data file is editable within service requests based on the policy.

Step 3 In the Role Type column, choose a device role from the drop-down list.

The role selections might include:

- N-PE
- PE-AGG
- U-PE
- CE (MULTI_VRF)
- CE (MANAGED)
- MVRF

Note The available device roles in the drop-down list are determined by the policy type.

Step 4 To add a template/data file click the **Add** link in the Template/Data File column.

The Add/Remove Templates window appears.

Step 5 Click the **Add** button to select a template/data file to associate with the policy.

Note If the device role is specified as U-PE or PE-AGG, templates can be selectively added based on whether the device has a UNI interface. For details on this feature, see Selectively Determining Templates for U-PE and PE-AGG Device Roles, page 9-23. The actual steps for adding templates/data files are the same as in the following steps.

The Template Datafile Chooser window appears.

This is a standard Template Manager window used to navigate to and choose templates and (optionally) data files in Prime Fulfillment.

Note The following steps involving the Template Datafile Chooser window assume a familiarity with the functionality of the window. For additional information about Template Manager and how templates and data files are created and managed in Prime Fulfillment, see Overview, page 9-1 The steps shown here are for example purposes. You must modify the steps as required for your environment. For example, you might want to choose only a template file or both a template file and a data file to associate with the policy. Both scenarios are supported.

Step 6 Navigate to a template in the folder tree and click it to select it.

The template is listed in the right side of the GUI, along with any data files that are associated with it.

Step 7 Check the check box to the left of a data file name and click the **Accept** button.

Note You can select only the template or both template and data file at this stage, depending on your needs, and whether or not a data file exists for the template.

The Template Datafile Chooser window closes and the selected template/data file appears listed in the Add/Remove Templates window.

If you did not choose a data file, then the Datafile column is blank.

Step 8 Check the check box to the left of the template name to choose the template.

Step 9 Under Action, use the drop-down list and choose **APPEND** or **PREPEND.**

Append tells Prime Fulfillment to append the template-generated CLIs to the regular Prime Fulfillment (non-template) CLIs (configlet). Prepend is the reverse (adds the template to the beginning of the configlet).

Step 10 Choose **Active** to use this template for service requests based on this policy.

If you do not choose Active, the template is not used.

Step 11 To associate additional templates/data files with the policy click **Add** in the Add/Remove Templates window and repeat the appropriate steps to add other templates/data files.

Step 12 To remove a template row from the window, check a template and click the **Remove** button to remove the template from the list.

Step 13 When you are satisfied with the selections in the Add/Remove Templates window, click **OK**.

The Template Association window appears with the template(s)/data file(s) listed as active link(s). If you have added more than one template/data file, they appear in a comma-separated list of links.

You can click on any link to return to the Add/Remove Templates window, in order to edit/update the template/data file information.

Step 14 Check the **Edit** check box to make the template/data file attributes editable in service requests based on the policy.

Step 15 To add additional templates/data files for a given role to the policy, you can click the **Add** button in the Template Association window and repeat the steps outlined above.

Step 16 To delete templates/data files that have been associated to the policy, check a template/data file to choose it.

Then click the **Delete** button to delete it from the Template Association window.

Step 17 When you are finished associating the template(s)/data file(s) to the policy, click the **Finish** button in the Template Association window.

The attributes for the policy are saved and the policy creation or modification is complete.

Selectively Determining Templates for U-PE and PE-AGG Device Roles

Prime Fulfillment provides the capability to selectively determine which U-PE and PE-AGG devices (for example, in a ring environment) to apply templates/data files. During template association in the service policy workflow, the U-PE and PE-AGG device roles have two options to associate templates/data files. These options are:

- Devices with UNI. This option causes templates/data files to be configured on devices of the specified role with a UNI interface.

- All other devices. This option causes templates/data files to be configured on all devices of the specified role, including those with a UNI interface.

Usage notes:

- The templates/data files are selected by clicking on the Add link next to the desired option. The subsequent steps are the same as provided in Associating Templates and Data Files to a Policy, page 9-21.
- This features is not applicable for device roles other than U-PE and PE-AGG. The N-PE role only displays a single Add link in the Template/Data File column.
- For backward compatibility, when editing or viewing old and existing policies, for U-PE and PE-AGG devices, associated templates/data files will display under the All other Devices option.
- When you copy an existing policy, you can copy associated templates/data files (if any) from the All other Devices or Devices with UNI options of the existing policy into the new policy. This is similar to normal Prime Fulfillment behavior.
- You can associate templates (without data files) for either the All other Devices or Devices with UNI options or both.
- Selective determination of templates is supported in all L2VPN and FlexUNI/EVC policy types and service requests. For MPLS VPN, only MPLS PE-CE and MPLS PE-NoCE policies and service requests are supported. For the MPLS VPN PE-CE policy type, this feature is applicable if the PE is or is not associated with an NPC.This feature is not available for Multi-VRFCE policies and service requests.

The following notes describe how this feature is supported in the service request workflow:

- During service request creation, selective templates are differentiated based on the devices having a UNI interface or having both UNI and NNI interfaces for the U-PE and PE-AGG device roles. Templates in the policy are copied to the respective devices functioning in the specified roles. There is no behavioral change for devices of other roles.
- The selective determination of templates is not applicable for service request modification scenarios, as after the service request is created, it is the user's decision to make any changes for templates configured on devices.

Using Templates with Service Requests

This section provides information on templates and data files with a service request. It contains the following sections:

Overview

This section provides overview information about template usage in service requests. It covers the following topics:

- Using Negate Templates to Decommission Template Configurations, page 9-27
- Using Templates and Data Files in the Service Request Workflow, page 9-28

For details on how these features are implemented in the Prime Fulfillment GUI, see the section Using Templates and Data Files in the Service Request Workflow, page 9-28.

Associating Templates to a Service Request

The template mechanism in Prime Fulfillment provides a way to add additional configuration information to a device configuration generated by a service request. To use the template mechanism, the policy on which the service request is based must have been set to enable templates. Optionally, templates and data files to be used by the service request can be specified in the policy. During service request creation, templates/data files can be added to a device configuration if the operator has the appropriate RBAC permission to do so. See the section Choosing a Template in the Service Request Workflow, page 9-28, for how to choose templates/data files in the service request workflow.

Associating Subtemplates During Service Provisioning

All templates can be used by other templates as building blocks. The template using other templates is called a super template. The template being used is called a subtemplate. A new attribute in the Template Editor allows subtemplates to be associated with a super template. The super template instantiates all required subtemplates by passing values for the variables in the subtemplate. After instantiation, the super template puts the configlets generated for the subtemplate into the super template.
Prime Fulfillment branches templates into subtemplates based on device type, line card type, port type, role type, and software versions. These optional attributes are set while creating the subtemplates. The subtemplates are selected based on the following matching criteria:

- Only exact matches are recognized for the card type and port type attributes. No wild card match is allowed for these attributes.
- Only an exact match is recognized for the device type attribute.
- For the software version attribute, the match is done for a software version equal to the current version, if available. If not, the previous highest version is matched.
- If exact matching attributes are not found, then the match proceeds with the criteria described in Table 9-1. An information message listing the exactly matched subtemplates of the super-template is shown if and only if any of the matching criteria are met.
- If none of the attributes are matched, then the default subtemplate is applied.
- If no default subtemplate exists, a subtemplate with all null attribute values is matched.
- If none of the rows specified in the table match, then Prime Fulfillment looks for subtemplates that are marked as device default, or else version default. If no subtemplates are marked as such, then no matching subtemplates are picked. A warning message is displayed.

The matching criteria are summarized in Table 9-1.

Table 9-1 Default SubTemplate Matching Criteria

Matching Order	Role Type	Device Type	Line Card	Port Type	Software Version
1	Exact Match	Exact Match	Exact Match	Exact Match	Exact Match
2	Exact Match	Exact Match	Exact Match	Exact Match	Previous Highest

Table 9-1 Default SubTemplate Matching Criteria

Matching Order	Role Type	Device Type	Line Card	Port Type	Software Version
3	Exact Match	Exact Match	Exact Match	No Values	Exact Match
4	Exact Match	Exact Match	Exact Match	No Values	Previous Highest
5	Exact Match	Exact Match	No Values	No Values	Exact Match
6	Exact Match	Exact Match	No Values	No Values	Previous Highest
7	Exact Match	Exact Match	No Values	No Values	No Values
8	Exact Match	No Values	Exact Match	Exact Match	Exact Match
9	Exact Match	No Values	Exact Match	Exact Match	Previous Highest
10	Exact Match	No Values	Exact Match	No Values	Exact Match
11	Exact Match	No Values	Exact Match	No Values	Previous Highest
12	Exact Match	No Values	No Values	No Values	Exact Match
13	Exact Match	No Values	No Values	No Values	Previous Highest
14	Exact Match	Default	No Values	No Values	No Values
15	Exact Match	No Values	No Values	No Values	Default
16	Exact Match	No Values	No Values	No Values	No Values

Additional usage notes for subtemplates:

- Prime Fulfillment does not perform checks for the depth of subtemplates. Only one level of subtemplates is supported.
- No validations are done to check if the super template and subtemplate structures are cyclic.
- When the operator attempts to delete a subtemplate that is referenced by a super template, a warning message is generated.
- Subtemplates can be modified.
- Subtemplates can be attached to multiple super templates.
- In the current release, multiple data files are not supported for subtemplates. If multiple data files are found, the service request automatically chooses the first data file (from a list of available data files, sorted alphabetically).

Creating Data Files During Service Request Creation

The operator can create data files “on demand” during service request creation. If template(s) are attached to a service policy, and no data file(s) exist for the template(s), a wizard prompts the operator to enter values for variables. If data file(s) are created on demand during service request creation, it is possible to modify any or all of the variables during modification or redeployment of the service request.

The service request workflow supports dynamic creation of data files as follows:

- If a template is marked as non-editable in the policy on which the service request is based, the operator cannot edit it during service request creation. However, the name of template and data files are still visible, even though they cannot be modified.
- If a template is marked as editable in the policy, then (assuming appropriate RBAC permission) the operator can change the template/data files during service request creation.

The following points apply if the template is editable:

- If a template is associated with a service policy, and at least one data file exists for the template, the operator can select the appropriate data file during service request creation.
- If only one data file exists for the template, it is automatically selected.
- During service request creation, the operator can enter values for template variables.
- Optionally, if no data file exists for the template, the operator can create a new data file during service request creation. When the Datafile Chooser window is opened from Template Association window, a Create Datafile button is provided, which allows the new data file to be created.
- The Create Datafile button is only displayed if the operator has the appropriate RBAC permissions to create a data file.

See the section Creating a Data File in the Service Request Workflow, page 9-29, for how to set up a data file in the service request workflow.

Using Negate Templates to Decommission Template Configurations

To remove a configuration created from a template/data file, a negate template must be applied to the existing service. Prime Fulfillment automatically applies the appropriate negate template during the decommission of the service request. For instructions on how to use the Prime Fulfillment Template Manager to create negate templates, see Negate Template, page 9-8

When a template is associated in a policy or service request, the negate template automatically gets associated. During decommission of the service, the negate template is used for deployment. When decommissioning a service request associated with a template/data file, the negate template is automatically picked up dynamically, by searching for a template name having the name of the original template followed by a suffix .Negate. This takes place at deployment time. Negate templates are dynamically instantiated based on the device attributes of the template to which it is associated.

Note Optional attributes (such as device type, line card type, port type, and software version) applied to a template automatically apply to the corresponding negate template. The optional attributes cannot be applied directly to negate templates.

When a service is decommissioned, the appropriate negate template is deployed. The data file for a negate template is selected during deployment as follows:

- If the negate template has no valid data file, either because there is no data file under the negate template with the same name as that of the main template or there is no data file at all, an error is raised during service request deployment.
- If only one data file is associated with the negate template, the data file is automatically selected. If there is a single data file for the negate template with a name that does not match that of the data file, then deployment will fail with errors and the service request will be moved to the INVALID state.
- In case of multiple data files, only data files with names that match negate template names are chosen.

The following points cover the behavior of templates in various modification scenarios:

- If you change the template associated with a service request, the negate template automatically changes to the negate template of the newly selected template. In this case, Prime Fulfillment executes the negate template of the previously associated template, as well as the newly associated template.

- When a template or negate template is modified, the service request does not roll back the configuration changes made earlier through the template.
- When a service request is modified, the template command is always deployed. (See the remaining bullet items for some additional clarifications.)
- When a service request is modified without changing template/data file information, the template commands are not redeployed. The only a modification that triggers a change in template/data file results is the negation of the old template and the addition of new template commands in the device configlet.
- When the ForceTemplateDeploy DCPL property is turned ON then, irrespective of templates being modified, if a service request is modified, templates are re-deployed. However, negate templates are not necessarily re-deployed. Negate templates are deployed only when a link/attachment circuit in the service request is deleted, which implicitly means removing templates associated with the link being deleted as well. When the ForceTemplateDeploy DCPL property is turned OFF, negate templates are instantiated under the following conditions:
  - Deleting or decommissioning a link/attachment circuit in a service request.
  - Modifying templates (for example, delete existing templates and adding new ones to a link, or deleting only existing ones).
  - Rehoming links/devices in a service request that has associated templates.
- When a device is changed in a service request, the negate template is deployed for the old device, and the template is deployed for the new device.
- When a link in a service request is removed and a new link is added, a negate template is deployed for the deleted link and a template is deployed for the added link.

Using Templates and Data Files in the Service Request Workflow

This section describes tasks related to templates, data files, and negate templates that can be performed in the service request workflow. The following tasks are covered:

- Choosing a Template in the Service Request Workflow, page 9-28
- Creating a Data File in the Service Request Workflow, page 9-29
- Decommissioning Service Requests with Added Templates, page 9-30
- Viewing Templates from the Service Requests Window, page 9-31

Choosing a Template in the Service Request Workflow

When creating a service request, the workflow involves selecting a policy on which to base the service request, setting interface and other attributes, and so on. The specific windows and attributes presented in the workflow depend on the type of service request, such as L2VPN, VPLS, MPLS, or FlexUNI/EVC.

To associate templates and data files in a service request, you must select a link in the appropriate window of the Service Request Editor window, usually by clicking the **Add** link for the device.

Note There is no choice of options to selectively determine templates for U-PE and PE-AGG devices during the service request workflow. Templates are automatically copied from the policy, based on the presence of a UNI interface on the devices functioning in U-PE and PE-AGG roles. See the section Selectively Determining Templates for U-PE and PE-AGG Device Roles, page 9-23, for more information on this feature.

To choose the template(s)/data file(s) for the device(s), perform the following steps.

Step 1 Click the **Add** link in Template/Datafile column for a device.

The Add/Remove Templates window appears.

Step 2 Click the **Add** button.

The Add/Remove Templates window appears.

Step 3 Navigate to a template in the folder tree and select it.

The template is listed in the right side of the GUI, along with any data files that are associated with it.

At this point, you can either select an existing data file, or click the **Create Data File** button to create a data file dynamically in the workflow. The rest of the steps in this section cover the case of selecting an existing template and data file. For instructions on how to create a data file dynamically, see the section Creating a Data File in the Service Request Workflow, page 9-29.

Step 4 Check the check box of a data file to choose it.

Step 5 Click the **Accept** button to confirm the choice.

The template/data file combination appears in the Add/Remove Templates window.

Step 6 To add additional templates/data files to the list, click the **Add** button and repeat the appropriate steps, as covered above.

Step 7 When you are satisfied with selection of templates/data files, click the **OK** button in the Add/Remove Templates window.

The templates/data files appear in the Template/Datafile column of the Template Association window.

If multiple templates/data files are selected for a device, they appear as a comma-separated list, as shown in the figure.

Step 8 Click the **Finish** button to create the service request with the template/data file selections you chose.

If the template associated to the service request is a super template comprising of one or more subtemplates, Prime Fulfillment displays a message confirming this.

For information about how templates/data files are instantiated when the service is deployed, see the information provided in the section Associating Templates to a Service Request, page 9-25.

Creating a Data File in the Service Request Workflow

During the final stage of setting the link attributes for a service request, the Template Association window appears. The Template Association window lists the devices comprising the link, the device roles, and the template(s)/data file(s) associated with the devices. You can choose the template(s)/data file(s) to be associated with the devices, as described in the section Choosing a Template in the Service

Request Workflow, page 9-28. If one of the templates selected in the Template Datafile Chooser window does not have an associated data file or if you would like create a new data file for it, you can do this dynamically in the workflow while setting up the service request.

To dynamically set up a new data file for a template, perform the following steps.

Step 1 In the Template Association window, click the **Add** link in the Template/Datafile column for a device.

(If a template was previously selected for a device, click the link for the template name.)

The Add/Remove Templates window appears.

Step 2 Click the **Add** button.

The Template Datafile Chooser window appears.

Step 3 Navigate to a template in the folder tree and select it.

The template is listed in the right side of the GUI, along with any data files that are associated with it. This example uses the AccessList1 template in the Examples directory.

Step 4 Click the **Create Data File** button to create a data file dynamically in the workflow.

The Data File Editor window appears.

Step 5 At this point, you are in the standard workflow for creating a data file in Prime Fulfillment.

In the Date File Editor window, you can specify a name and description for the data file, set variable values, view the configlet, and so on. For details on how to perform these steps, see Overview, page 9-1

Step 6 When you have completed setting the attributes for the new data file, click **Save** and then **Close** to save this information and close the file; click **Configure** to show the configuration file; or click **Close** and then be sure to click **OK**, if you want to save the information you have created.

If you do not want to save this information, click **Close** and then click **Cancel**.

When the data file is saved, the Template Datafile Chooser window appears with the newly created data file listed.

Decommissioning Service Requests with Added Templates

This section describes how to decommission Prime Fulfillment service requests that have added templates.

Note For general information on how templates are used in Prime Fulfillment, see Overview, page 9-1

Template commands are treated independently from those associated with a service creation. Consequently, template commands must be removed separately from the device(s) during a service decommission. To remove prior template commands, a separate template is needed during a decommission process. Decommissioning a service request does not automatically remove the original template commands. A separate negate template needs to be added to the decommission process and the original templates must be removed. The negate template must contain the necessary NO commands to successfully remove any unwanted IOS commands added by the original template.

The standard way to create a service request with a template added is as follows:

1. Define the service policy.
2. Build a template with a data file (and also a negate template and data file).

3. Create the service request with the template added. The steps to do this are covered in relevant chapters of this guide.
4. Deploy the service request to which the template was added.

To decommission a deployed service request, including associated templates, you must perform the following steps.

1. Create a negate template with data file (if one does not exist). This is used to remove the commands imposed by the original template. For an explanation of negate templates, see Chapter 4, "Using Templates" in the *Cisco Prime Fulfillment API Programmer Guide 6.2.*
2. Decommission the service request. The negate template will be picked up dynamically.

   The service request remains in the **Requested** state, but changed to an Operation Type of Delete.
3. Deploy the service request. This decommissions the service request and downloads the negate template, which removes the original template commands.

Viewing Templates from the Service Requests Window

In the Service Request Manager window, a paper clip icon appears in the Data Files column if a service request has one or more templates associated with it.

Note You can use the **Show Services with** field to search for service requests that have a specific data or template file. Choose **Data File Name** or **Template Name** from the drop-down list and enter a search string in the **matching** field. The matching field is not case-sensitive and supports wildcards (*). You can further limit the search by using the **of Type** field to confine the search to a particular service type. When listing service requests using Template Name, provide the entire path of the template file location (for example: examples\template, where examples is the folder name and template implies the template name).

To view the configlet(s) for the template(s) associated with a service request, perform the following steps.

Step 1 In the Service Request Manager window, check the check box for a service request with an associated template, as indicated by a paper clip icon in the Data Files column.

Step 2 Click the **Details** button.

The Service Request Details window appears.

The Associated data file(s) row displays a link for each data file associated with the service request, as shown in the figure.

Step 3 Click a data file link to display the configlet for the template.

Step 4 After viewing the configlet, click **OK** to close the configlet display window.

Step 5 Click **OK** to close the Service Request Details window.

Step 6 As an alternative, you can access the data files associated with a service request by clicking on the paper clip icon in the Service Requests window.

The Data file Details for Service Request window appears.

The window displays only a list of the data files associated with the service request.

Step 7 Click a data file link to display the configlet for the template.

Step 8 After viewing the configlet, click **OK** to close the configlet display window.

Step 9 Click **Close** to close the Service Request Datafile Details window and return to the Service Requests window.

Template Examples

To access template examples, choose **Service Design > Templates > Template Manager** and navigate through the folders in the Template pane. You can continue clicking the **arrow** sign next to each created folder and subfolder until you get to the last level of information. The last possible level is the template name.

Table 9-2 documents some of the available template examples. Refer to the Prime Fulfillment GUI for a complete listing of available examples.

Table 9-2 Template Examples and Their Descriptions

Folder	Template	Description
DIA-Channelization	10K-CHOC12-STS1-PATH	Sample template to break down channelized OC12 to STS-1 paths.
	10K-CT3-CHANNELIZED	Sample template creates T1 out of channelized T3 line card.
	10K-CT3-UNCHANNELIZED	Sample template Creates either a fullrate T3 or a subrate T3 interface out of a channelized T3.
	PA-MC-E3-CHANNELIZED	Sample template Creates E1 (channel groups) out of E3.
	PA-MC-STM1-AU3-CHANNELIZE	Sample template Creates E1 (channel groups) out of TUG-2. This template uses AU-3 AUG mapping that further creates TUG-2s.
	PA-MC-STM1-AU4-CHANNELIZE	Sample template Creates E1 (channel groups) out of TUG-2. This template uses AU-4 AUG mapping that creates TUG-3s and TUG-2s.
	PA-MC-T3-CHANNELIZED	Sample template Creates T1 (channel groups) out of T3.
Examples	AccessList	Demonstrates templates with nested repeat loop and multi-dimension variable.
	AccessList1	Demonstrates the simplest template variable substitution.
	CEWanCOS	Demonstrates if-else statements, repeat statements, mathematical expressions, and one-dimensional variables.

Table 9-2 Template Examples and Their Descriptions (continued)

Folder	Template	Description
QoS/L2/ATM	CLP_Egress	Sample template to demonstrate the setting of qos_group and ATM Cell Loss Priority at the output of an interface.
	CLP_Ingress	Sample template sets MPLS experimental bit of the ATM Cell marked with Cell Loss Priority, at the input of an interface.
QoS/L2/Ethernet	3400_Egress	
QoS/L2/FrameRelay	classification	Sample template to demonstrate the bandwidth reservation based on FrameRelay DLCI value.

Summary of Repository Variables

This section contains the following tables:

- Table 9-4 on page 9-43, "MPLS Repository Variables"
- Table 9-3 on page 9-33, "L2VPN Repository Variables"
- Table 9-7 on page 9-52, "VRF Repository Variables"
- Table 9-5 on page 9-46, "FlexUNI/EVC Repository Variables"
- Table 9-6 on page 9-47, "VPLS Repository Variables"

Table 9-3 provides a summary of the MPLS Repository variables available from Prime Fulfillment Templates.

Table 9-3 MPLS Repository Variables

Repository Variable	Dimension	Description
Advertised_Routes_To_CE	2	List of one or more IP addresses of the advertised static route to be placed on the PE to define the CE's address space.
CARD_TYPE	0	Refers to NPE or UNI interface depending on whether the service is implemented with ethernet access.
CE_BGP_AS_ID	0	BGP AS ID on a CE when the routing protocol between a CE and a PE is BGP.
CE_BGP_AS_ID_IPV6	0	If the Address family is IPv6, this specifies the Border Gateway Protocol (BGP) routing protocol Autonomous System (AS) number.
CE_DLCI	0	DLCI value on CE for Frame Relay encapsulation.
CE_EIGRP_AS_ID	0	EIGRP AS ID on a CE when the routing protocol between a CE and a PE is EIGRP.

Table 9-3 MPLS Repository Variables (continued)

Repository Variable	Dimension	Description
CE_Facing_MVRFCE_BGP_AS_ID	0	BGP AS ID on an MVRFCE when the routing protocol between a CE and an MVRFCE is BGP, when an MPLS link includes an MVRFCE.
CE_Facing_MVRFCE_DLCI	0	DLCI value on CE facing MVRFCE interface for Frame Relay encapsulation, when an MPLS link includes an MVRFCE.
CE_Facing_MVRFCE_EIGRP_AS_ID	0	EIGRP AS ID on an MVRFCE when the routing protocol between a CE and an MVRFCE is EIGRP, when an MPLS link includes an MVRFCE.
CE_Facing_MVRFCE_Intf	0	Name of the CE facing interface on an MVRFCE, when an MPLS link includes an MVRFCE.
CE_Facing_MVRFCE_Intf_Address	0	IP address assigned to the CE facing MVRFCE interface, when an MPLS link includes an MVRFCE.
CE_Facing_MVRFCE_Intf_Encap	0	Encapsulation for CE facing of an MVRFCE interface, when an MPLS link includes an MVRFCE.
CE_Facing_MVRFCE_Intf_Name	0	Name of the CE facing MVRFCE interface, when an MPLS link includes an MVRFCE.
CE_Facing_MVRFCE_Intf_Type	0	Interface type for CE facing of an MVRFCE interface, when an MPLS link includes an MVRFCE.
CE_Facing_MVRFCE_Ospf_Process_ID	0	OSPF process ID on MVRFCE when the routing protocol between a CE and an MVRCE is OSPF, when an MPLS link includes an MVRFCE.
CE_Facing_MVRFCE_Tunnel_Src_ Addr	0	Tunnel source address on CE facing MVRFCE interface for GRE encapsulation when an MPLS link includes an MVRFCE.
CE_Facing_MVRFCE_VCD	0	VCD value on CE facing MVRFCE interface for ATM encapsulation, when an MPLS link includes an MVRFCE.
CE_Facing_MVRFCE_VCI	0	VCI value on CE facing MVRFCE interface for ATM encapsulation, when an MPLS link includes an MVRFCE.
CE_Facing_MVRFCE_VLAN_ID	0	VLAN ID on CE facing MVRFCE interface for Ethernet encapsulation, when an MPLS link includes an MVRFCE.
CE_Facing_MVRFCE_VPI	0	VPI value on CE facing MVRFCE interface for ATM encapsulation, when an MPLS link includes an MVRFCE.

Table 9-3 MPLS Repository Variables (continued)

Repository Variable	Dimension	Description
CE_Intf_Address	0	IP address assigned to the CE interface.
CE_Intf_Encap	0	Encapsulation of the CE interface.
CE_Intf_Name	0	Name of the CE interface.
CE_MVRFCE_Bandwidth_Metric_For_ Redistribution	0	Bandwidth metric for redistribution of EIGRP when the routing protocol between a CE and an MVRFCE is EIGRP, when an MPLS link includes an MVRFC.
CE_MVRFCE_BGP_AS_ID	0	BGP AS ID on a CE when the routing protocol between a CE and an MVRFCE is BGP, when an MPLS link includes an MVRFCE.
CE_MVRFCE_Delay_Metric_For_ Redistribution	0	Delay metric for redistribution of EIGRP when the routing protocol between a CE and an MVRFCE is EIGRP, when an MPLS link includes an MVRFC.
CE_MVRFCE_EIGRP_AS_ID	0	EIGRP AS ID on a CE when the routing protocol between a CE and an MVRFCE is EIGRP, when an MPLS link includes an MVRFCE.
CE_MVRFCE_Loading_Metric_For_ Redistribution	0	Loading metric for redistribution of EIGRP when the routing protocol between a CE and an MVRFCE is EIGRP, when an MPLS link includes an MVRFC.
CE_MVRFCE_MTU_Metric_For_ Redistribution	0	MTU metric for redistribution of EIGRP when the routing protocol between a CE and an MVRFCE is EIGRP, when an MPLS link includes an MVRFC.
CE_MVRFCE_Ospf_Process_ID	0	OSPF process ID on CE when the routing protocol between a CE and an MVRCE is OSPF, when an MPLS link includes an MVRFCE.
CE_Ospf_Process_ID	0	OSPF process ID on CE when the routing protocol between a CE and a PE is OSPF.
CE_Tunnel_Src_Addr	0	Tunnel source address on CE for GRE encapsulation.
CE_VCD	0	VCD value on CE for ATM encapsulation.
CE_VCI	0	VCI value on CE for ATM encapsulation.
CE_Vlan_ID	0	VLAN ID on CE for Ethernet encapsulation.
CE_VPI	0	VPI value on CE for ATM encapsulation.
Export_Map	0	Name of the export map associated with the VRF.
Extra_CE_Loopback_Required	0	Flag to indicate whether an extra loopback request is required on the CE.

Table 9-3 MPLS Repository Variables (continued)

Repository Variable	Dimension	Description
Import_Map	0	Name of the import map associated with the VRF.
Is_Default_Info_Originate	0	Flag to indicate whether the **default-information originate** command for BGP on the PE when STATIC is a running protocol between a CE and a PE.
Is_Default_Info_Originate_IPV6	0	If the Address family is IPv6, Flag to indicate whether the default-information originate command for BGP on the PE when STATIC is a running protocol between a CE and a PE.
Is_Default_Routes_Sent_To_CE	0	Flag to indicate whether the default routes are sent to a remote CE.
Join_Grey_Mgmt_Vpn	0	Flag to indicate whether MPLS will join a Grey Management VPN.
Max_route_threshold	0	Percentage of the maximum number of routes that can be imported into the VRF.
Max_Routes	0	Maximum number of routes than can be imported into the VRF.
MPLSCeInterfaceMask	0	The mask of the IP address assigned to the CE interface for a particular MPLS VPN link.
MPLSCeLoopbackAddress	0	The IP address of the extra CE loopback address for a particular MPLS VPN link.
MPLSCLECeFacingEncapsulation	0	The encapsulation of the interface on the device facing the CE for that particular MPLS VPN link.
MPLSCLECeFacingInterfaceName	0	The name of the interface on the device facing the CE for that particular MPLS VPN link.
MPLSCLEPeFacingEncapsulation	0	The encapsulation of the interface on the device facing the PE for that particular MPLS VPN link.
MPLSCLEPeFacingInterfaceName	0	The name of the interface on the device facing the PE for that particular MPLS VPN link.
MPLSExportRouteTargets	1	List of Route Targets that are exported for a particular VRF associated with the MPLS VPN link.
MPLSImportRouteTargets	1	List of Route Targets that are imported for a particular VRF associated with the MPLS VPN link.
MPLSPeInterfaceMask	0	The mask of the IP address assigned to the PE interface for a particular MPLS VPN link.
Multicast_Enabled_IPv6	0	Enabling and disabling a Multicast IPv6 VPN. If the check box is enabled, Multicast IPv6 VPN configlets are generated.

Table 9-3 MPLS Repository Variables (continued)

Repository Variable	Dimension	Description
Multicast_Route_Limit	0	Multicast route limit value for the VRF
MVRFCE_CE_Advertised_Routes_To_CE	2	List of one or more IP addresses of the advertised static route to be placed on the PE to define the CE's address space, when the MPLS link includes an MVRFCE.
MVRFCE_CE_IP_Unnumbered	0	Flag to indicate whether the MVRCE to CE link is unnumbered, when an MPLS link includes an MVRFCE.
MVRFCE_CE_Is_Default_routes_Sent_To_CE	0	Flag to indicate whether the default routes are sent to a remote CE, when an MPLS link includes an MVRFCE.
MVRFCE_CE_NBR_ALLOW_AS_IN	0	AllowASIn flag when the routing protocol between a CE and an MVRFCE is BGP, when an MPLS link includes an MVRFCE.
MVRFCE_CE_NBR_AS_OVERRIDE	0	ASOverride flag when the routing protocol between a CE and an MVRFCE is BGP, when an MPLS link includes an MVRFCE.
MVRFCE_CE_Ospf_Area_Number	0	OSPF area number when the routing protocol between a CE and an MVRCE is OSPF, when an MPLS link includes an MVRFCE.
MVRFCE_CE_Ospf_Route_Policy	0	Name of the Redistribute OSPF route policy to be configured when an MPLS link includes an MVRFCE_CE.
MVRFCE_CE_Routes_To_Reach_Other_Sites	2	List of one or more IP addresses to specify the static routes to put on the CE, when the MPLS link includes an MVRFCE.
MVRFCE_CE_Routing_Protocol	0	Routing protocol between MVRFCE and CE.
PE_BGP_AS_ID	0	BGP AS ID on a PE when the routing protocol between a CE and a PE is BGP.
PE_Cable_Both_Helper_Address_List	1	List of DHCP server IP addresses to which both cable modem and host UDP broadcasts are forwarded.
PE_Cable_Modem_Helper_Address_list	1	List of DHCP server IP addresses to which cable modem UDP broadcasts are forwarded.
PE_Cable_Modem_Host_Helper_Address_List	1	List of DHCP server IP addresses to which host UDP broadcasts are forwarded.
PE_Cable_Modem_Secondary_Address_List	1	List of cable modem secondary addresses for cable interfaces.
PE_CE_Bandwidth_Metric_For_Redistribution	0	Bandwidth metric for redistribution of EIGRP when the routing protocol between a CE and a PE is EIGRP.
PE_CE_BGP_ADVERTISE_INTERVAL_IPV6		Advertising interval value for BGP routing protocol if the Address family is IPv6.

Table 9-3 MPLS Repository Variables (continued)

Repository Variable	Dimension	Description
PE_CE_BGP_DEFAULT_ORIGINATE_ROUTE_POLICY_IPV4	0	Default orginate route policy name when the routing protocol between a CE and a PE is BGP.
PE_CE_BGP_DEFAULT_ORIGINATE_ROUTE_POLICY_IPV6	0	Default orginate route policy name when the routing protocol between a CE and a PE is BGP, if the address family is IPV6.
PE_CE_BGP_MAX_PREFIX_NUMBER	0	BGPNieghbor MaxPrefix value for BGP routing protocol.
PE_CE_BGP_MAX_PREFIX_NUMBER_IPV6	0	BGPNieghbor MaxPrefix value for BGP routing protocol, if the Address family is IPv6.
PE_CE_BGP_MAX_PREFIX_RESTART	0	BGPNieghborMaxprefix restart value for BGP routing protocol.
PE_CE_BGP_MAX_PREFIX_RESTART_IPV6	0	BGPNieghborMaxprefix restart value for BGP routing protocol, if the address family is IPv6.
PE_CE_BGP_MAX_PREFIX_THRESHOLD	0	BGPNieghborMaxprefix threshold value for BGP routing protocol.
PE_CE_BGP_MAX_PREFIX_THRESHOLD_IPV6	0	BGPNieghborMaxprefix threshold value for BGP routing protocol, if the address family is IPv6.
PE_CE_BGP_MAX_PREFIX_WARNING_ONLY	0	BGPNieghborMaxprefix warnily_only (enable/disable).
PE_CE_BGP_MAX_PREFIX_WARNING_ONLY_IPV6	0	BGPNieghborMaxprefix warnily_only (enable/disable), if the Address family is IPv6.
PE_CE_BGP_Neighbor_Route_Map_Or_Policy_In	0	Name of the BGP Neighbor Route Map/Policy In to be configured on the device.
PE_CE_BGP_Neighbor_Route_Map_Or_Policy_Out	0	Name of the BGP Neighbor Route Map/Policy Out to be configured on the device.
PE_CE_Delay_Metric_For_Redistribution	0	Delay metric for redistribution of EIGRP when the routing protocol between a CE and a PE is EIGRP.
PE_CE_EIGRP_AUTHENTICATION_KEY_CHAIN_NAME	0	Keychain name to authenticate EIGRP protocol traffic on one or more interfaces, if the Routing protocol between CE and PE is EIGRP.
PE_CE_EIGRP_AUTHENTICATION_KEY_CHAIN_NAME_IPV6	0	If the address family is IPV6, this specifies keychain name to authenticate EIGRP protocol traffic on one or more interfaces if the routing protocol between CE and PE is EIGRP

Table 9-3 MPLS Repository Variables (continued)

Repository Variable	Dimension	Description
PE_CE_IP_Unnumbered	0	Flag to indicate whether the PE to CE link is unnumbered.
PE_CE_IPV6_Routing_Protocol	0	Routing protocol between PE and CE if the address family is IPv6.
PE_CE_Loading_Metric_For_ Redistribution	0	Loading metric for redistribution of EIGRP when the routing protocol between a CE and a PE is EIGRP.
PE_CE_MTU_Metric_For_ Redistribution	0	MTU metric for redistribution of EIGRP when the routing protocol between a CE and a PE is EIGRP.
PE_CE_NBR_Allow_AS_In	0	AllowASIn flag when the routing protocol between a CE and a PE is BGP.
PE_CE_NBR_Allow_AS_In_IPV6	0	If the Address family is IPv6, AllowASIn flag when the routing protocol between a CE and a PE is BGP.
PE_CE_NBR_AS_Override	0	ASOverride flag when the routing protocol between a CE and a PE is BGP.
PE_CE_NBR_AS_Override_IPV6	0	If the Address family is IPv6, ASOverride flag when the routing protocol between a CE and a PE is BGP.
PE_CE_NBR_Send_Community_IPV6	0	If the Address family is IPv6, then these values specify the “Standard”, “extended”, “Both” of the Send_Community attribute.
PE_CE_Ospf_Area_Number	0	OSPF area number when the routing protocol between a CE and a PE is OSPF.
PE_CE_Ospf_Match_Internal_External	0	Name of the Redistribute OSPF match criteria to be configured on the device.
PE_CE_OSPF_METRIC_TYPE	0	Metric type when the routing protocol between a CE and a PE is OSPF.
PE_CE_OSPF_METRIC_VALUE	0	Metric value when the routing protocol between a CE and a PE is OSPF.
PE_CE_Ospf_Route_Policy	0	Name of the Redistribute OSPF route policy to be configured on the device.
PE_CE_OSPF_ROUTE_POLICY	0	Route policy name when the routing protocol between a CE and a PE is OSPF.
PE_CE_Reliability_Metric_For_ Redistribution	0	Reliability metric for redistribution of EIGRP when the routing protocol between a CE and a PE is EIGRP.
PE_CE_Routing_Protocol	0	Routing protocol between PE and CE.
PE_DLCI	0	DLCI value on PE for Frame Relay encapsulatio.n
PE_EIGRP_AS_ID	0	EIGRP AS ID on a PE when the routing protocol between a CE and a PE is EIGRP.

Table 9-3 MPLS Repository Variables (continued)

Repository Variable	Dimension	Description
PE_Facing_MVRFCE_BGP_AS_ID	0	BGP AS ID on an MVRFCE when the routing protocol between a PE and an MVRFCE is BGP, when an MPLS link includes an MVRFCE.
PE_Facing_MVRFCE_DLCI	0	DLCI value on PE facing MVRFCE interface for Frame Relay encapsulation, when an MPLS link includes an MVRFCE.
PE_Facing_MVRFCE_EIGRP_AS_ID	0	EIGRP AS ID on an MVRFCE when the routing protocol between a PE and an MVRFCE is EIGRP, when an MPLS link includes an MVRFCE.
PE_Facing_MVRFCE_Intf	0	Name of the PE facing interface on an MVRFCE, when an MPLS link includes an MVRFCE.
PE_Facing_MVRFCE_Intf_Address	0	IP address assigned to the PE facing MVRFCE interface, when an MPLS link includes an MVRFCE.
PE_Facing_MVRFCE_Intf_Encap	0	Encapsulation for PE facing of an MVRFCE interface, when an MPLS link includes an MVRFCE.
PE_Facing_MVRFCE_Intf_Name	0	Name of the PE facing MVRFCE interface, when an MPLS link includes an MVRFCE.
PE_Facing_MVRFCE_Intf_Type	0	Interface type for PE facing of an MVRFCE interface, when an MPLS link includes an MVRFCE.
PE_FACING_MVRFCE_OSPF_ Process_ID	0	OSPF process ID on an MVRFCE when the routing protocol between a PE and an MVRCE is OSPF, when an MPLS link includes an MVRFCE.
PE_Facing_MVRFCE_Tunnel_Src_Addr	0	Tunnel source address on PE facing MVRFCE interface for GRE encapsulation when an MPLS link includes an MVRFCE.
PE_Facing_MVRFCE_VCD	0	VCD value on PE facing MVRFCE interface for ATM encapsulation, when an MPLS link includes an MVRFCE.
PE_Facing_MVRFCE_VCI	0	VCI value on PE facing MVRFCE interface for ATM encapsulation, when an MPLS link includes an MVRFCE.
PE_Facing_MVRFCE_VLAN_ID	0	VLAN ID on PE facing MVRFCE interface for Ethernet encapsulation, when an MPLS link includes an MVRFCE.
PE_Facing_MVRFCE_VPI	0	VPI value on PE facing MVRFCE interface for ATM encapsulation, when an MPLS link includes an MVRFCE.

Table 9-3 MPLS Repository Variables (continued)

Repository Variable	Dimension	Description
PE_Intf_Address	0	IP address assigned to the PE interface.
PE_Intf_Address_IPV6	0	If the Address family is IPv6, this specifies the IP address of the interface.
PE_Intf_Desc	0	Interface description for the PE interface.
PE_Intf_Encap	0	Encapsulation of the PE interface.
PE_Intf_Name	0	Name of the PE interface.
PE_Intf_Shutdown	0	Shutdown flag for the PE interface.
PE_IS_Cable_Modem_Maintenance_ Interface	0	Flag to indicate whether the interface is a maintenance interface.
PE_MVRFCE_Bandwidth_Metric_For_ Redistribution	0	Bandwidth metric for redistribution of EIGRP when the routing protocol between a PE and an MVRFCE is EIGRP, when an MPLS link includes an MVRFCE.
PE_MVRFCE_BGP_AS_ID	0	BGP AS ID on a PE when the routing protocol between a PE and an MVRFCE is BGP, when an MPLS link includes an MVRFCE.
PE_MVRFCE_Delay_Metric_For_ Redistribution	0	Delay metric for redistribution of EIGRP when the routing protocol between a PE and an MVRFCE is EIGRP, when an MPLS link includes an MVRFCE.
PE_MVRFCE_EIGRP_AS_ID	0	EIGRP AS ID on a PE when the routing protocol between a PE and an MVRFCE is EIGRP, when an MPLS link includes an MVRFCE.
PE_MVRFCE_IP_Unnumbered	1	Flag to indicate whether the PE to MVRFCE link is unnumbered, when an MPLS link includes an MVRFCE.
PE_MVRFCE_Loading_Metric_For_ Redistribution	0	Loading metric for redistribution of EIGRP when the routing protocol between a PE and an MVRFCE is EIGRP, when an MPLS link includes an MVRFCE.
PE_MVRFCE_MTU_Metric_for_ redistribution	0	MTU metric for redistribution of EIGRP when the routing protocol between a PE and an MVRFCE is EIGRP, when an MPLS link includes an MVRFCE.
PE_MVRFCE_NBR_ALLOW_AS_IN	0	AllowASIn flag when the routing protocol between a PE and an MVRFCE is BGP, when an MPLS link includes an MVRFCE.
PE_MVRFCE_NBR_AS_OVERRIDE	0	ASOverride flag when the routing protocol between a PE and an MVRFCE is BGP, when an MPLS link includes an MVRFCE.

Table 9-3 MPLS Repository Variables (continued)

Repository Variable	Dimension	Description
PE_MVRFCE_Ospf_Area_Number	0	OSPF area number when the routing protocol between a PE and an MVRCE is OSPF, when an MPLS link includes an MVRFCE.
PE_MVRFCE_OSPF_Process_ID	0	OSPF process ID on PE when the routing protocol between a PE and an MVRCE is OSPF, when an MPLS link includes an MVRFCE.
PE_MVRFCE_Ospf_Route_Policy	0	Name of the Redistribute OSPF route policy to be configured when an MPLS link includes a PE_MVRFCE.
PE_MVRFCE_Reliability_Metric_For_ Redistribution	0	Reliability metric for redistribution of EIGRP when the routing protocol between a PE and an MVRFCE is EIGRP, when an MPLS link includes an MVRFCE.
PE_MVRFCE_Routing_Protocol	0	Routing protocol between PE and MVRFCE, when an MPLS link includes an MVRFCE.
PE_OSPF_PROCESS_ID	0	OSPF process ID on PE when the routing protocol between a CE and a PE is OSPF.
PE_Tunnel_Src_Addr	0	Tunnel source address on PE for GRE encapsulation.
PE_VCD	0	VCD value on PE for ATM encapsulation.
PE_VCI	0	VCI value on PE for ATM encapsulation.
PE_Vlan_ID	0	VLAN ID on PE for Ethernet encapsulation.
PE_VPI	0	VPI value on PE for ATM encapsulation.
rd	0	Route Distinguisher value for the VRF.
RD_FORMAT	0	Defines the RD Format to be used in the MPLS Link, such as RD_AS or RD_IPADDR.
RD_IPADDRESS	0	Defines the RD_IPADDRESS Value to be used in the MPLS Link, if the RD Format is RD_IPADDRESS.
Redistribute_Connected	0	Flag to indicate whether the connected routes are redistributed into BGP on the PE.
Redistribute_Connected_IPV6	0	Flag to indicate whether the connected routes are redistributed into BGP on the PE, if the address family is IPv6.
Redistribute_Static	0	Flag to indicate whether the static routes are redistributed into BGP on the PE.
Redistribute_Static_IPV6	0	Flag to indicate whether the static routes are redistributed into BGP on the PE, if the Address family is IPv6
Redistributed_Protocol	1	List of routing protocols to be redistributed.
Rip_Metrics	0	Metric for redistribution associated with RIP.

Table 9-3 MPLS Repository Variables (continued)

Repository Variable	Dimension	Description
Routes_To_Reach_Other_Sites	2	List of one or more IP addresses to specify the static routes to put on the CE.
vrfName	0	Name of the VRF.

Table 9-4 provides a summary of the L2VPN Repository variables available from Prime Fulfillment Templates.

Table 9-4 L2VPN Repository Variables

Repository Variable	Dimension	Description
AC_Loopback_Address	0	PE loopback address also known as the router ID.
CARD_TYPE	0	Refers to NPE or UNI interface depending on whether the service is implemented with ethernet access.
CE_DLCI	0	DLCI value on CE for Frame Relay encapsulation.
CE_Encap	0	Encapsulation of the CE interface.
CE_Intf_Desc	0	Interface description for the CE interface.
CE_Intf_Main_Name	0	Major interface name for the CE interface.
CE_Intf_Shutdown	0	Shutdown flag for the CE interface.
CE_VCD	0	VCD value on CE for ATM encapsulation.
CE_VCI	0	VCI value on CE for ATM encapsulation.
CE_Vlan_ID	0	VLAN ID on CE for Ethernet encapsulation.
CE_VPI	0	VPI value on CE for ATM encapsulation.
L2VPNCLECeFacingEncapsulation	0	Encapsulation of the UNI.
L2VPNCLECeFacingInterfaceName	0	Name of the UNI.
L2VPNCLEPeFacingEncapsulation	0	Encapsulation of the NNI (should always be dot1q).
L2VPNCLEPeFacingInterfaceName	1	Name of the NNI (uplinks) (the number can be more than 1 in case of a ring topology, hence any array).
L2VPNDFBIT_SET	0	Indicates not to fragment the bit set (for L2TPv3 only).
L2VPNDynamicModeUseDefaults	0	Dynamic session setup using Prime Fulfillment default values (for L2TPv3 only).
L2VPN_intf_main_name	1	The main interface name for a CE or PE port.
L2VPNIP_PMTU	0	Enable the discovery of the path MTU for tunneled traffic (for L2TPv3 only).

Table 9-4 L2VPN Repository Variables (continued)

Repository Variable	Dimension	Description
L2VPNIP_TOS	0	Configure the value of the TOS byte in IP headers of tunneled packets or reflects the TOS byte value from the inner IP header (for L2TPv3 only).
L2VPNIP_TTL	0	Configure the value of the time to live byte in the IP headers (for L2TPv3 only).
L2VPNL2TP_CLASS_NAME	0	The L2TP class name to overwrite the default L2TP class name (for L2TPv3 only).
L2VPNL2TPv3Sequence	0	Specifies the direction in which sequencing of data packets in a pseudo wire is enabled (for L2TPv3 only).
L2VPNLocalCookieHighValue	0	Specifies the last 4 bytes of the value that the peer PE must include in the cookie field of incoming L2TP packets (for L2TPv3 only).
L2VPNLocalCookieLowValue	0	Specifies the first 4 bytes of the value that the peer PE must include in the cookie field of incoming L2TP packets (for L2TPv3 only).
L2VPNLocalCookieSize	0	Specifies the size (0, 4, or 8) of the cookie field of incoming L2TP packets (for L2TPv3 only).
L2VPNLocalHostName	0	Hostname of the N-PE that peers with a remote N-PE in the L2VPN end-to-end wire.
L2VPNLocalLoopback	0	Loopback address of the N-PE that peers with a remote N-PE in the L2VPN end-to-end wire.
L2VPNLocalSessionId	0	Specifies the ID for the local L2TPv3 session (for L2TPv3 only).
L2VPNLocalSwitchLoopBack1	1	The loopback1 for the local switch (for L2TPv3 only).
L2VPNLocalSwitchLoopBack2	1	The loopback2 for the local switch (for L2TPv3 only).
L2VPNRemoteCookieHighValue	1	Specifies the last 4 bytes of the value that this PE must include in the cookie field of incoming L2RP packets (for L2TPv3 only).
L2VPNRemoteCookieLowValue	1	Specifies the first 4 bytes of the value that this PE must include in the cookie field of incoming L2RP packets (for L2TPv3 only).
L2VPNRemoteCookieSize	1	Specifies the size (0, 4, or 8) of the cookie field of outgoing L2TP packets (for L2TPv3 only).
L2VPNRemoteHostName	0	Hostname of the remote N-PE that peers with the N-PE in context in the L2VPN end-to-end wire.

Table 9-4 L2VPN Repository Variables (continued)

Repository Variable	Dimension	Description
L2VPNRemoteLoopback	0	Loopback address of the remote N-PE that peers with the N-PE in context in the L2VPN end-to-end wire.
L2VPNRemoteSessionID	1	Specifies the ID for the remote L2TPv3 session (for L2TPv3 only).
L2VPNSessionSetupMode	0	Defines how the L2TPv3 session is set up (static or dynamic) (for L2TPv3 only).
L2VPNTransportMode	0	Defines how the L2TPv3 data is transferred (for Frame Relay: DLCI or Port; for ATM: VP or VC) (for L2TPv3 only).
L2VPNUniMajorInterfaceName	0	The main interface name of the UNI.
L2VPNVcId	0	The virtual circuit ID of the L2TPv3 or AToM tunnel.
PE_DLCI	0	DLCI value on PE for Frame Relay encapsulation.
PE_Encap	0	Encapsulation of the PE interface.
PE_Intf_Desc	0	Interface description for the PE interface.
PE_Intf_Main_Name	0	Major interface name for the PE interface.
PE_VCD	0	VCD value on PE for ATM encapsulation.
PE_VCI	0	VCI value on PE for ATM encapsulation.
PE_Vlan_ID	0	VLAN ID on PE for Ethernet encapsulation.
PE_VPI	0	VPI value on PE for ATM encapsulation.
PseudoWire_Class_Type_Of_Core	0	Core type of the Service Provider over which L2VPN is provisioned.
Uni_Aging	0	Length of time the MAC address can stay on the port security table.
Uni_Cdp_Enable	0	Flag to enable or disable layer 2 tunnelling on a Cisco Discover Protocol (CDP).
Uni_Cdp_Threshold	0	Number of packets per second to be received before the interface is shut down for the CDP protocol.
Uni_Mac_Address	0	Number of MAC addresses allowed for port security.
Uni_Port_Security	0	Flag to enable or disable security on a UNI interface.
Uni_Protocol_Tunnelling	0	Flag to enable or disable Layer 2 Bridge Protocol Data Unit (BPDU) protocol tunnelling on a UNI interface.
Uni_Recovery_Interval	0	Amount of time to wait before recovering a UNI port.

Table 9-4 L2VPN Repository Variables (continued)

Repository Variable	Dimension	Description
Uni_Shutdown	0	Flag indicating whether the User Network Interface (UNI) is shutdown.
Uni_Speed	0	Value of the UNI link speed.
Uni_Stp_Enable	0	Flag to enable or disable layer 2 tunnelling on a Spanning Tree Protocol (STP).
Uni_Stp_Threshold	0	Flag to enable or disable layer 2 tunnelling on an STP.
Uni_Violation_Access	0	Action taken when a port security violation is detected.
Uni_Vtp_Enable	0	Flag to enable or disable layer 2 tunnelling on a VLAN Trunk Protocol (VTP).
Uni_Vtp_Threshold	0	Flag to enable or disable layer 2 tunnelling on a VTP.

Table 9-5 provides a summary of the VRF Repository variables available from Prime Fulfillment Templates.

Table 9-5 VRF Repository Variables

Repository Variable	Dimension	Description
Address_Family	0	Addressing scheme from Service Request.
Cerc_Hub_RT	0	Customer Edge Routing Community (CERC) for Hub Route Target.
Cerc_Spoke_RT	0	CERC for Spoke Route Target.
Export_Map	0	Name of the export map associated with the VRF.
Export_RT_List	0	One or more Route Targets (RTs) to be exported from the VRF.
Import_Map	0	Name of the import map associated with the VRF.
Import_RT_List	0	One or more RTs to be imported in the VRF.
Max_Routes	0	Maximum number of routes that can be imported into the VRF.
Max_Threshold	0	Percentage of the maximum number of routes that can be imported into the VRF.
PE	0	Name of the Provider Edge (PE) device.
PE_BGP_AS	0	BGP Autonomous ID for PE device.
RD	0	Route Distinguisher value for the VRF.
Vrf_Name	0	Name of the VRF.

Table 9-6 provides a summary of the FlexUNI/EVC Repository variables available from Prime Fulfillment Templates.

Table 9-6 FlexUNI/EVC Repository Variables

Repository Variable	Dimension	Description
ATMIMA_VCI	0	Virtual circuit identifier for ATM/IMA service. A number between 1 and 65535.
ATMIMA_VPI	0	Virtual path identifier for ATM/IMA service. A number between 0 and 255.
ATM_Encapsulation	0	ATM encapsulation type. Possible values are AAL5 and AAL0.
AUG_MAPPING	0	A true value configures the administrative unit group mapping when SDH framing is used.
AU_THREE_NUMBER	0	Used to configure a particular admistrative unit type 3 (au-3) of an E1 line. A number from 1 to 3.
BACKUP_VC_ID	0	Backup virtual circuit ID for the AToM, where backup is configured for the primary pseudowire. This is applicable only for pseudowire core type connectivity between only two N-PEs.
CARD_TYPE	0	Refers to NPE or UNI interface depending on whether the service is implemented with ethernet access.
CEM_CLASS_NAME	0	A CEM class name.
CEM_GROUP_ID	0	CEM Group ID under the controller creates a CEM interface that has the same slot/subslot/port information as the controller. The number it can take depends on E1 or T1 line.
CEM_INTERFACE	0	The CEM interface is an interface that has been created as a result of configuring a CEM group under a controller. A CEM interface has the same slot/subslot/port information as that of its parent controller.
CHANNELISATION_MODE	0	Specifies the Channelization mode for a RAN service.
CLOCK_SOURCE_TYPE	0	The type of clock source. May be INTERNAL or LINE.
CONFIG_BRIDGE_DOMAIN	0	Value is true if USE_SVI is enabled.
CONTROLLER_NAME	0	Specifies the name of the controller.
CONTROLLER_TYPE	0	Type of controller used by device in a CEM TDM service. . May be E1 or T1 ????

Table 9-6 FlexUNI/EVC Repository Variables (continued)

Repository Variable	Dimension	Description
CORE_TYPE	0	Core type connectivity. Possible values for this are: a) pseudowire, b) VPLS, c) Local connect.
DEJITTERBUFFER	0	The size of the buffer used for network jitter in CEM configuration mode. The range is 1 to 500 milliseconds.
EVC_LINK_ID	0	Returns top EVC link ID of EVC SR.
EVC_NPE_HOSTNAME	0	NPE device hostname in EVC SR.
EVC_SR_DESCRIPTION	0	EVC SR description.
EVC_SR_JOB_ID	0	SR JOB ID of EVC SR
EVC_UNI_DEVICE_ID	0	UNI device ID. Allows configuration of a unique MPID value on a per-link basis. This is used for CFM, IP SLA, and Ethernet OAM support.
FLEXUNI_ATM_VCD	0	Returns the ATM VCD/sub-interface value provided for ATM links.
FLEXUNI_ATM_VCI	0	Returns the ATM VCI value provided for ATM links.
FLEXUNI_ATM_VPI	0	Returns the ATM VPI value provided for ATM links.
FLEX_UNI_BD_NAME	0	Returns the Bridge Domain name used for IOS XR.
FLEX_UNI_BG_NAME	0	Returns the Bridge Group name used for IOS XR.
FLEXUNI_ELINE_NAME	0	Returns the p2p Eline name used for IOS XR.
FLEXUNI_L2_GROUP_NAME	0	Returns the L2VPN group name used for IOS XR.
FLEXUNI_PW_CLASS_NAME	0	Returns the PW class element name used for IOS XR.
FLEXUNI_REMOTE_HOSTNAME	0	Returns the remote peer's hostname.
FLEXUNI_REMOTE_LOOPBACK	0	Returns the remote peer's loopback IP address.
FLEXUNI_VLANTranslationCeVlan	0	Returns the CE VLAN provided for VLAN translation.
FLEXUNI_VLANTranslationNode	0	Returns the PE device role of the node where the VLAN translation takes place on this attachment link.
FLEXUNI_VLANTranslationOuterVlan	0	Returns the Outer VLAN provided for VLAN translation.
FLEXUNI_VLANTranslationType	0	Returns the type of VLAN translation chosen for this attachment link.

Table 9-6 FlexUNI/EVC Repository Variables (continued)

Repository Variable	Dimension	Description
IDLEPATTERN	0	The pattern of dates used to replace the of each lost CESoPSN data packet. The range is from 0x00 to 0xFF, in hexadecimal. ????
IS_FLEX_UNI_LINK	0	Value is true if EVC LINK is FLEXUNI link.
LOCAL_CONNECT_NAME	0	Name of the connection between two Ethernet flow points (EFPs) using the connect command. Applicable only when there are two links that are FlexUNI/EVC enabled.
MAC_ACL_NAME	0	MAC ACL name.
MAC_ACL_RANGE	0	Range value specified for MAC ACL.
MATCH_INNER_VLANS	0	Contains the VLAN IDs that need to be matched for the ingress frame's inner VLAN tag. Applicable only for FlexUNI/EVC enabled links.
MATCH_OUTER_VLANS	0	Contains the VLAN IDs that need to be matched for the ingress frame's outer VLAN tag. Applicable only for FlexUNI/EVC enabled links.
No_Cell_Packed	0	Used in ATM services. The maximum number of cells to be packed into a packet. A number from 2 to 28.
PAYLOADSIZE	0	The payload size used in CEM configuration mode. The range is 32 to 1312 bytes.
PE_DEVICE_PLATFORM	0	Returns the platform type information of the N-PE device used in this link.
PE_INTERFACE_NAME	0	N-PE interface of the link for a service. This is the same as the UNI_INTERFACE_NAME for direct connect links.
PE_OR_UNI_INTF_DESC	0	UNI interface description.
PUSH_INNER_VLAN_ID	0	Push a second Dot1q VLAN tag onto an ingress frame. Applicable only for links configured with FlexUNI/EVC.
PUSH_OUTER_VLAN_ID	0	Push a Dot1q VLAN (outer) tag onto an ingress frame. Applicable only for links configured with FlexUNI/EVC.
PW_CLASS_NAME	0	Returns the pseudowire class name used for any IOS XR devices on current link.
PW_TUNNEL_ID	0	Tunnel ID that is configured with a pseudowire class for the N-PE (applicable only for pseudowire core type selection).
RAN_SERVICE_TYPE	0	RAN ervice type can be either SAToP_UNFRAMED or CESoPN_TIMESLOT.

Table 9-6 FlexUNI/EVC Repository Variables (continued)

Repository Variable	Dimension	Description
SERVICE_INSTANCE_ID	0	Service instance ID (a number: 1 to 8000) corresponding to the EFP for a FlexUNI/EVC enabled link.
SERVICE_INSTANCE_NAME	0	Name of the EFP given to the Service instance being configured for a FlexUNI/EVC enabled link.
SONET_FRAME_TYPE	0	Configures the controller framing type. Framing type is either SDH or SONET.
SR_JOB_ID	0	Returns unique Job ID of the current service request.
STD_UNI	0	Standard UNI status of the UNI interface.
STORM_CTL_BROADCAST_TRAFFIC	0	Storm control broadcast traffic value.
STORM_CTL_MULTICAST_TRAFFIC	0	Storm control multicase traffic value.
STORM_CTL_UNICAST_TRAFFIC	0	Storm control unicast traffic value.
STS_MODE_TYPE		sts-1 mode type ??? vt-15
STS_ONE_NUMBER		The sts-1 number. A number from 1 to 3.
SYSTEM_MTU	0	System MTU size used.
Sub_Interface	0	Sub-Interface number for an ATM pseudowire VC service.
TIME_SLOT	0	Specifies the time slot value/range for configuring a RAN service. Range is 1-24 for T1 controllers and 1-31 for E1 controllers.
TRANSLATE_INNER_VLAN_ID	0	Target inner VLAN ID of a frame that is being translated (VLAN translation). Applicable only for FlexUNI/EVC enabled links. This is applicable for 1:2/2:2 types of translation.
TRANSLATE_OUTER_VLAN_ID	0	Target outer VLAN ID of a frame that is being translated (VLAN translation). Applicable only for FlexUNI/EVC enabled links. This is applicable for any kind of translations (1:1/1:2/2:1/2:2).
TUG_THREE_NUMBER	0	Specifies the tug-3 number.
TUG_TWO_NUMBER	0	Specifies the tug-2 number.
TUNNEL_CDP_DROP_THRESHOLD	0	CDP DROP threshold value used.
TUNNEL_STP_DROP_THRESHOLD	0	STP DROP threshold value used.
TUNNEL_VTP_DROP_THRESHOLD	0	VTP DROP threshold value used.
T_LINE_NUMBER	0	Specifies the T1 line number.
Timer1	0	First MCPT timer value in microseconds. A number between 500 and 10000.

Table 9-6 FlexUNI/EVC Repository Variables (continued)

Repository Variable	Dimension	Description
Timer2	0	Second MCPT timer value in microseconds. A number between 1000 and 10000.
Timer3	0	Third MCPT timer value in microseconds. A number between 1500 and 10000.
UNI_AGING	0	The aging value of the UNI.
UNI_DEVICE_PLATFORM	0	Returns the platform type information of the UNI device used in this link.
UNI_ENCAPSULATION_TYPE	0	Encapsulation on the UNI. Possible values are: a) Dot1Q Trunk, b) Dot1Q Tunnel, c) Access.
UNI_INTERFACE_NAME	0	UNI of the link for a service. This is the same as PE_INTERFACE_NAME for direct connect links.
UNI_PORT_SECURITY	0	The port security status of the UNI.
UNI_SHUTDOWN	0	The UNI shutdown status.
UNI_SPEED	0	The speed value of the UNI.
UNI_VIOLATION_ACTION	0	Type of violation action used.
USER_DEFINED_ACL_NAME	0	User defined ACL name used in the attachment circuit.
UPE_FACING_INTERFACE_NAME	1	Arrays of one or two elements, containing names of NNI interfaces on NPE towards the U-PE. Two interfaces exist if access is via a ring, otherwise just one is present.
USE_SPLIT_HORIZON	0	Value is true if split horizon is enabled.
Use_MCPT_Timer	0	Indicates which MCPT timer to use for ATM services.
VC_ID	0	The virtual circuit ID for the AToM where pseudowire is the core connectivity type between two N-PEs.
VLAN_ID	0	VLAN ID corresponding to the service on PE devices for the link. For links that are configured with FlexUNI/EVC, this is applicable only on N-PE, while MATCH_OUTER_VLANS represents the service for that link.
VLAN_NAME	0	VLAN name configured for the VLAN ID corresponding to the link for the service.
VPLS_VPN_ID	0	VPLS VPN ID for VPLS core type connectivity.
VPN_ID	0	VPN name associated to EVC SR.
VTG_NUMBER	0	Specifies the Virtual Tributary Group number.

Table 9-7 provides a summary of the VPLS Repository variables available from Prime Fulfillment Templates.

Table 9-7 VPLS Repository Variables

Repository Variables	Dimension	Description
CARD_TYPE	0	Refers to NPE or UNI interface depending on whether the service is implemented with ethernet access.
VPLSBridgeDomainId	0	Bridge domain ID value.
VPLSCeEncapsulation	0	The encapsulation of the CE interface for a particular VPLS link.
VPLSCeInterfaceName	0	The name of the CE interface for a particular VPLS link.
VPLSCeMajorInterfaceName	0	The name of a major interface on a CE for a particular VPLS link.
VPLSCLECeFacingEncapsulation	0	The encapsulation of interfaces for a particular device facing the CE.
VPLSCLECeFacingInterfaceName	0	The interface name for a particular device facing the CE (the number can be more than 1 in case of a ring topology, hence any array).
VPLSCLEPeFacingEncapsulation	0	The encapsulation of interfaces for a particular device facing the PE
VPLSCLEPeFacingInterfaceName	1	The list of interface names for a particular device facing the PE (the number can be more than 1 in case of a ring topology, hence any array).
VPLSDisableCDP	0	The flag to specify if the CDP has been disabled on a UNI for a particular VPLS link.
VPLSFilterBPDU	0	The flag to specify whether the BPDUs will be filtered on a UNI for a particular VPLS link.
VPLSPeEncapsulation	0	The encapsulation of the PE interface for a particular VPLS link.
VPLSPeInterfaceDescription	0	The description assigned to the PE interface for a particular VPLS link.
VPLSPeInterfaceName	0	The name of the PE interface for a particular VPLS link.
VPLSPeMajorInterfaceName	0	The name of a major interface on a PE for a particular VPLS link.
VPLSPeNeighbors	1	The list of PE POPs participating in a particular VPLS VPN.
VPLSPeVfiName	0	The VFI name assigned to a particular VPLS instance existing on the PE POP.
VPLSPeVlanId	0	The VLAN ID assigned to the PE for a particular VPLS link.

Table 9-7 VPLS Repository Variables (continued)

Repository Variables	Dimension	Description
VPLSPeVpnId	0	The VPN ID assigned to a particular VPLS VPN.
VPLSSystemMTU	0	The maximum MTU value for a packet arriving on a UNI for a particular VPLS link.
VPLSTunnelCDPEnable	0	The flag to specify if the CDP packets will be tunneled to the remote site for a particular VPLS link.
VPLSTunnelCDPThreshold	0	The threshold value assigned for a CDP protocol before a violation action is reported on a UNI for a particular VPLS link.
VPLSTunnelRecoveryInterval	0	Interval for the UNI to recover from a shutdown scenario.
VPLSTunnelSTPEnable	0	The flag to specify if the STP packets will be tunneled to the remote site for a particular VPLS link.
VPLSTunnelSTPThreshold	0	The threshold value assigned for a STP protocol before a violation action is reported on a UNI for a particular VPLS link.
VPLSTunnelVTPEnable	0	The flag to specify if the VTP packets will be tunneled to the remote site for a particular VPLS link.
VPLSTunnelVTPThreshold	0	The threshold value assigned for a VTP protocol before a violation action is reported on a UNI for a particular VPLS link.
VPLSUniAging	0	The aging timer set on a UNI for a particular VPLS link.
VPLSUniDuplex	0	The duplex assigned to the UNI for a particular VPLS link.
VPLSUniMajorInterfaceName	0	The name of a major interface on a UNI device for a particular VPLS link.
VPLSUniMaxMacAddress	0	The maximum number of Mac addresses that can be learned on a UNI for a particular VPLS link.
VPLSUniPortSecurity	0	The port security option on a UNI for a particular VPLS link.
VPLSUniProtocolTunneling	0	The flag to specify if the protocols will be tunneled to the remote site for a particular VPLS link.
VPLSUniSecureMacAddresses	1	The explicit list of Mac addresses that can be learned on a UNI for a particular VPLS link.
VPLSUniShutdown	0	The shutdown flag on a UNI for a particular VPLS link.

Table 9-7 VPLS Repository Variables (continued)

Repository Variables	Dimension	Description
VPLSUniSpeed	0	The speed assigned to the UNI for a particular VPLS link.
VPLSUniViolationAction	0	The violation action option on a UNI for a particular VPLS link.
VPLSUseNativeVlan	0	The flag to specify if the native VLAN will be used on a UNI for a particular VPLS link.

Importing and Exporting Templates

The **importExportTemplateDB** tool is available to import and export templates into and from a Prime Fulfillment database.

Note If a **Negate** template is present, it is automatically imported or exported for every import or export template.

You can import or export the complete or partial template database by specifying appropriate arguments. You can find this tool at: **$PRIMEF_HOME/bin/importExportTemplateDB.sh**.

Enter the following:

importExportTemplateDB.sh <*admin_user_id*> <*password*> [<*other_arguments*>]

where:

<*admin_user_id*> is user identifier for someone with the **admin** role.

<*password*> is the password for the one with the **admin** role.

<*other_arguments*> is any combination of the following arguments separated by a space:

-nooverwrite
If you choose to use this **nooverwrite** argument, to prevent the overwriting of existing templates in the database, it must precede all other arguments and must be in the third position after <*admin_user_id*> and <*password*>.

Note The default (when **nooverwrite** is not specified) is to overwrite the templates.

-exp_db <*dest-dir*>
Use this argument to export all templates and data files in the database, where <*dest-dir*> is the destination directory to which you want to export.

-imp_db <*src-dir*>
Use this argument to import all the files in <*src-dir*> into the database, where <*src-dir*> is the source directory from which you want to import. The files in <*src-dir*> are created by the **exp_db** process.

-exp_template_folder <*src-folder-path*> <*dest-dir*>
Use this argument to export a database template folder and its subfolders, where **<***src-folder-path*> is the full path of the template folder to export and <*dest-dir*> is the directory where to place the exported files.

-imp_template_folder <*src-dir*> <*dest-folder*>
Use this argument to import all files in <*src-dir*> into the database, where <*src-dir*> is the source directory to import, and <*dest-folder*> is the destination import template folder.

-imp_template <*srcfile*> <*dest-folder*> <*template-name*>
Use this argument to import a template into the database, where <*srcfile*> is the full path of the template to import, <*dest-folder*> is the full path of the parent folder, and <*template-name*> is the template name in the database.

-imp_datafile <*srcfile*> <*dest-template*> <*datafile-name*>
Use this argument to import a template data file into the database, where <*srcfile*> *is the full path of the datafile to import,* <*dest-template*> *is the full path of the parent template,* and <*datafile-name*> is the data file name in the database.

-exp_template <*template-pathname*> <*output-file*>
Use this argument to export the database template to a file, where <*template-pathname*> is the full path of the template to export, and <*output-file*> is the output filename.

-exp_datafile <*datafile-pathname*> <*output-file*>
Use this argument to export a template data file to a file, where <*datafile-pathname*> is the full path of the template data file to export, and <*output-file*> is the output filename.

Known Issue with Importing Template Data Using the importExportTemplateDB.sh Script

Template data imported by using the **importExportTemplateDB.sh** script only shows up in the Template Manager GUI after the HTTPD or Prime Fulfillment processes are restarted.

One workaround is to manually create a template. Then all the previously imported templates and data files show up. With this workaround, there is no need to restart the HTTPD or Prime Fulfillment processes.

The steps to do this are as follows:

Step 1 Import the templates and data files.

Step 2 Check in Template Manager and verify if they show up.

Refreshing the browser and logging out/in will not help.

Step 3 Manually create a simple template in Template Manager.

As soon as you save and click on **Close**, the Template Manager window gets all the data, and all the previously imported templates, data files now appear.

Frequently Asked Questions

The following sections provide questions and answers that can help you troubleshoot Template Manager issues:

How do I split a string?

Prime Fulfillment provides a function substring**ToDelim(),** which can split the given string and return the substring based on the given delimiter.

Syntax:

substringToDelim (srcString, delimChar, 0/1)
where:

0 returns the string before the delimiter.
1 returns the string after the delimiter.

Usage: **$b=$TMSystem.substringToDelim(“10.11.230.145”, ".230.145", "0")**

Result: The value of **$b** is **10.11**. If **1** is specified instead of **0**, the value of **$b** is **230.145**.

How do I obtain address information from the given IP address?

Prime Fulfillment provides the functions that can be used to get the address, mask, and reverse mask from the given IP address.

Usage:

$TMSystem.getAddr ("10.33.4.5/30") returns 10.33.4.5
$TMSystem.getMask ("10.33.4.5/30") returns 255.255.255.252
$TMSystem.getReverseMask ("10.33.4.5/30") returns 0.0.0.3
$TMSystem.getNetworkAddr ("10.33.4.5/30") returns 10.33.4.4
$TMSystem.GetClassfulNetworkAddr ("10.33.4.5/30") returns 10.0.0.0
$TMSystem.CurrentTimeInIOSFormat () returns hh:mm:ss day_of_month month_of_year year

How do I obtain the octets from the given IP address?

Prime Fulfillment provides the functions that can return the octets when called.

Usage:

$TMSystem.getOctet1($ipAddr) returns the first octet of **ipAddr**
$TMSystem.getOctet2($ipAddr) returns the second octet of **ipAddr**
$TMSystem.getOctet3($ipAddr) returns the third octet of **ipAddr**
$TMSystem.getOctet4($ipAddr) returns the fourth octet of **ipAddr**

How do I call a subtemplate in a template?

A subtemplate can be called in a main template. The subtemplate being called should be called with its data file. The variable is declared as a subtemplate. The location of the subtemplate is specified in the data file.

Usage: In the template body the subtemplate is declared as:

$a. callWithDatafile("data1")

where:

the variable **a** is declared as a subtemplate in the variables

data1 is the name of the data file of the subtemplate, and

in the data file the path of the subtemplate path is specified.

How do I concatenate two strings?

Concatenation of strings is simple.

For example:

where: **$a=vpnsc** and **$b=properties**
then: **${a}${b}** concatenates these two strings and gives the result as **vpnscproperties**.
or, **${a}_${b}** gives the result as **vpnsc_properties**.

How can I convert a string to an integer and how can I increase the last octet of the IP address by one?

The last octet of the IP address can be increased by using the following code:

```
#set($d=$TMSystem.getOctet1($c))
#set($e=$TMSystem.getOctet2($c))
#set($f=$TMSystem.getOctet3($c))
#set($g=$TMSystem.getOctet4($c))
#set($valueOfString = $g)
#set($valueOfCharsCount = $valueOfString.length() - 1)
#set($valueOfVector = "0123456789")
#set($valueOfBase = 1)
#set($valueOfInt = 0)
#foreach($valueOfCharIterator in $valueOfCharsCount..0)
#set($valueOfChar=$valueOfString.charAt($valueOfCharIterator).toString())
```

#set($valueOfInt = $valueOfInt +$valueOfVector.indexOf($valueOfChar) * $valueOfBase)
#set($valueOfBase = $valueOfBase * 10)
#end
#set($valueOfInt = $valueOfInt+1)

The incremental value is **$d.$e.$f.$valueOfInt**

Can I use nested if statements?

If statements can be nested. Proper care must be taken for indentation when nesting **if** statements. The following code shows the usage of nested **if** statements, **elseif** statements, and the comparisons made in the **if** clause.

```
#if($a==”a”) // here: string comparison is made
  --
    #if($b || $d) // here: $b and $d are the Boolean expressions. || equals OR and && equals AND
      --
      #if(!$c) // here: $c can be integer, string, or Boolean.
        ---
          #if($p<10)// here: $p is a integer.
          #elseif($p==10)
          #end
    #end
  #end
#end
```

How can I perform basic arithmetic operations?

Velocity Template Language (VTL) supports built-in mathematical functions that can be used in the templates with the set directives.

Usage:

```
#set($a = $b + 3)
#set($a = $b - 6)
#set($a = $b * 6)
#set($a = $b / 5)
#set($a = $b % 2)
```

Note Only integers are valid for performing mathematical operations in the VTL.

How can I retrieve data from a two-dimensional array and what is the use of $velocityCount?

The default name for the loop counter variable reference, which is specified in the velocity.properties file, is **$velocityCount**. By default the counter starts at **1**, but this can be set to either **0** or **1** in the **velocity.properties** file at:
$PRIMEF_HOME/resources/webserver/tomcat/shared/lib/velocity-dep-*VelocityVersion***.jar** (where the current *VelocityVersion* is 1.3.1-rc2). The associated settings are:

directive.foreach.counter.name=velocityCount
directive.foreach.counter.initial.value=1

Data from an array can be obtained by using **get($i)**

where: **$i** is the $velocityCount.

The following example illustrates the usage of the method **get()**:

Usage:
```
#foreach ($Acl in $ACL-List)
    #set ($i = $velocityCount)
    #foreach ($protocol in $Protocol-Lists.get($i))
    #set ($j = $velocityCount)
       access-list $Acl permit $protocol $Source-IP.get($i).get($j)
    #end
    #end
```

where:

$ACL-List is a one-dimensional array.
$Protocol-Lists and **$Source-IP** are two-dimensional arrays.

Here the $velocityCount is set to **1** by default. It can be changed in velocity.properties, if desired.

How can I print $a instead of its value?

Printing a value without processing is done by use of the character **** ,even if the value of the variable for **a** is defined.

Usage:

\$a gives output as **$a** if **$a** is defined. If **$a** is not defined, it is printed as **\$a**.

What is the difference between #include() and #parse()?

The **#include("velocity.txt")** directive allows you to import a file and then include the file in the location where it is defined. The content of the file is made available to the template engine. The *.vm files can also be called by using **#include**. The name of the file can also be passed by a variable. For security reasons, the file should be included under **TEMPLATE_ROOT** (/vob/ntg/dev/resources/templatesystem).

The **#parse("velocity.vm")** directive allows you to import a local file that contains VTL.Velocity will parse the VTL and render the template specified. The template that **#parse** references must be included under **TEMPLATE_ROOT**. The **#parse** directive only takes a single argument. VTL templates can have **#parse** statements referring to templates that in turn have **#parse** statements. The default value of the **directive.parse.max.depth** property is set to 10, in the **velocity.properties** file at: **$PRIMEF_HOME/resources/webserver/tomcat/shared/lib/velocity-dep-***VelocityVersion***.jar** (where the current *VelocityVersion* is 1.3.1-rc2) and can be modified, if desired.

Note If the **directive.parse,max.depth** property is not present in the **velocity.properties** file, the default is set to 10.

Example:

In **TEMPLATE_ROOT,** the file **velocity.vm** has the following content:

```
welcome to the parse file
The count is $count
#set($count = $count - 1)
#set($cl-list="cl1","cl2","cl3")
#foreach($i in $cl-list)
ipcommunity-list permit $i 30:20
#end
The count is $count
returning from parse
```

The template body contains the following:

```
#set($count=8)
#include("velocity.vm")
-------------------------
#parse("velocity.vm")
-------------------------
welcome back to template
The value of count is $count

The following O/P is obtained:
welcome to the parse file
The count is $count
#set($count = $count - 1)
#set($cl-list="cl1","cl2","cl3")
#foreach($i in $cl-list)
ipcommunity-list permit $i 30:20
#end
The count is $count
returning from parse
-------------------------
welcome to the parse file
The count is 8
ipcommunity-list permit cl1 30:20
ipcommunity-list permit cl2 30:20
ipcommunity-list permit cl3 30:20
The count is 7
returning from parse
-------------------------
welcome back to template
The value of count is 7.
```

Note The previous examples clearly show that variables are parsed in the **#parse** directive and not in the **#include directive.**

What is a macro and how is it used?

The directive macro is almost similar to a function. This has a set of statements, which can be called repetitively.

Example:

```
#macro(community $CL $bgp-list)
   #foreach($bgp in $bgp-list)
      ip $CL standard permit $bgp
   #end
#end

#set($bgp_list ="20:10","30:10","40:10","50:10")
#set($CL = "community-list")

#community($CL $bgp_list)
```

Here, the macro name of **community** is defined. The macro takes two arguments **$CL** and **$bgp-list**. The macro is called at the end line.

The output of the previous template is:

```
ip community-list standard permit 20:10
ip community-list standard permit 30:10
ip community-list standard permit 40:10
ip community-list standard permit 50:10
```

What is a range operator and how can I use it?

The range operator can be used in conjunction with **#set** and **#foreach** statements. It is used to produce an object array containing integers. The range operator has the following construction **n..m.**

Example:

```
#set($a=0..2)
#foreach($b in $a)
    $b
#end
#foreach($c in -2..2)
$c
#end
```

How can I split strings containing special characters?

```
#foreach ($i in $PE_Intf_Name.split('\.')) $i #end
```

here: In the first iteration, **$i** contains the string before the period, and in the second iteration, **$i** contains the string after the period.

How can I use repository variables?

Repository variables can be selected in the data file. When a template along with a data file is associated with a Service Request and the Service Request is deployed, then the value of the repository variable gets substituted.

How can I use a variable as a dynamic URL?

A variable declared as a dynamic URL can call the URL, by the method:

callUrl(String S)

For example: **$a. callUrl("http://www.cisco.com")**

Can I see more examples?

Examples are given for:

- Usage of Strings, page 9-62
- Usage of a Macro, page 9-64
- Usage of Subtemplates, page 9-64

Usage of Strings

The body of the template contains:

```
## This example illustrates the usage of strings

#set($a="Fast")
#set($b="ethernet")
interface ${a}_${b}

#foreach ($i in $PE_Intf_Name.split('\.'))
$i
#end

#set($c="10.11.230.145")
#set($b=$TMSystem.substringToDelim($c, ".230.145", "0"))
interface Loopback1
description By VPN-SC
ip vrf forwarding V31:eigrpfm
ip address ${b}.20.34 255.255.255.255
no ip directed-broadcast

#set($b=$TMSystem.substringToDelim($c, ".230.145", "1"))
interface Loopback1
description By VPN-SC
ip vrf forwarding V31:eigrpfm
ip address 20.45.${b} 255.255.255.255
no ip directed-broadcast

#set($c="10.33.4.5/30")
#set($d=$TMSystem.getAddr($c))
The Address of $c is $d
#set($d=$TMSystem.getMask($c))
The mask of $c is $d
#set($d=$TMSystem.getReverseMask($c))
The Reverse mask of $c is $d
```

```
#set($d=$TMSystem.getNetworkAddr($c))
The network address of $c is $d

#set($e=$TMSystem.currentTimeInIOSFormat())
The current time in IOS format is : $e

-----------------------------------------------------
getting the octets from the ipaddress
#set($c="10.33.4.5")
#set($e=$TMSystem.getOctet1($c))
The first Octet of $c is $e
#set($e=$TMSystem.getOctet2($c))
The second Octet of $c is $e
#set($e=$TMSystem.getOctet3($c))
The third Octet of $c is $e
#set($e=$TMSystem.getOctet4($c))
The fourth Octet of $c is $e

The variables are declared as strings, integers, or sub-templates accordingly.

The Output of the above template body is:

interface Fast_ethernet

10
11
12
13

interface Loopback1
description By VPN-SC
ip vrf forwarding V31:eigrpfm
ip address 10.11.20.34 255.255.255.255
no ip directed-broadcast

interface Loopback1
description By VPN-SC
ip vrf forwarding V31:eigrpfm
ip address 20.45.230.145 255.255.255.255
no ip directed-broadcast

The Address of 10.33.4.5/30 is 10.33.4.5
The mask of 10.33.4.5/30 is 255.255.255.252
The Reverse mask of 10.33.4.5/30 is 0.0.0.3
The network address of 10.33.4.5/30 is 10.33.4.4

The current time in IOS format is: 00:17:01 21 Aug 2006

-----------------------------------------------------
getting the octets from the ipaddress
The first Octet of 10.33.4.5 is 10
```

The second Octet of 10.33.4.5 is 33
The third Octet of 10.33.4.5 is 4
The fourth Octet of 10.33.4.5 is 5

Usage of a Macro

The body of the template contains:

```
## This example illustrates the usage of macro

#macro(community $CL $bgp-list)
#foreach($bgp in $bgp-list)
ip $CL standard permit $bgp
#end
#end

#set($bgp_list = "20:10","30:10","40:10","50:10")
#set($CL = "community-list")

#community($CL $bgp_list)
```

The Output is obtained as:

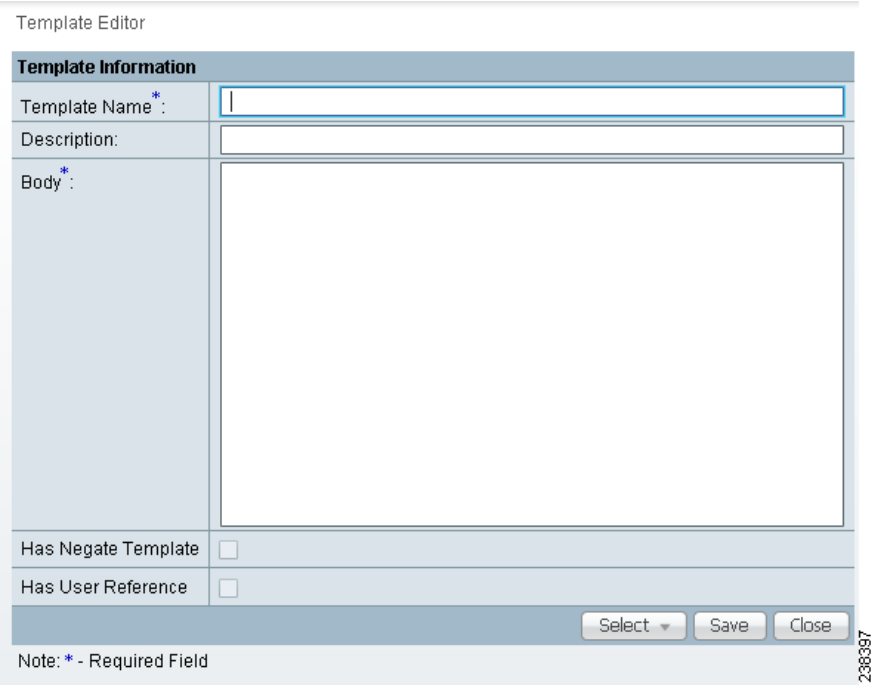
```
ip community-list standard permit 20:10
ip community-list standard permit 30:10
ip community-list standard permit 40:10
ip community-list standard permit 50:10
```

Usage of Subtemplates

The body of the template is as follows:

```
## This example illustrates the usage of the sub-template

$a.callWithDatafile("data1")
```



The variable **a** is declared as a subtemplate. The data file provided here, **data**, must be a data file for the template **a**, which must also exist. In the data file of the main template, the path of the subtemplate is specified.

In the data file of the main template, the specified path of the subtemplate might be the same directory or a different directory.

CHAPTER 10

Monitoring

This chapter explains the monitoring activity. It contains the following sections:

Ping

Ping is the way Cisco Prime Fulfillment monitors the VPN connectivity, that is, verifies the connectivity among various edge devices comprising the VPN.

Note Ping features are not supported on devices running IOS XR.

To achieve this, you can perform a series of pings among these devices. Ping has the following benefits:

- Service independent and therefore can be used for functional auditing of MPLS applications.
- Can establish whether a service is working without doing a functional audit for that service.
- Can be used to verify IPv4 connectivity among CPEs prior to VPN service deployment.

However, Ping does not do the following:

- Ping does not work in environments where ICMP traffic is blocked, for example, in a Cisco IOS router with an access-list denying all ICMP traffic.
- Ping can only inform you that there is a connectivity problem. It does not offer any service-specific information. The connectivity problem can be due to many reasons, such as device failure, misconfiguration, and so on, which ping cannot distinguish.
- Only the immediate subnet behind the router's customer-facing (also, inside or nonsecured) interface is supported. Campus subnets cannot be supported.

The Ping GUI supports all possible pings for MPLS service requests. This section explains how to ping MPLS service requests.

Note Cisco Prime Fulfillment has a component Prime Diagnostics that might help you. See the *Cisco Prime Fulfillment User Guide 6.2.*

After you choose **Inventory > Device Tools > Ping.** The Services window appears.

The **Type** field indicates **MPLS**. Follow these steps:

Step 1 Check the check box next to each row for which you want to configure ping parameters.

Step 2 Click the **Configure Ping Parameters** button, which becomes enabled.

The MPLS Parameters window appears.

Fill in the following and then click **Start Ping**:

- **Ping Type: Do PE to CE Ping**—When this radio button is chosen, a VRF ping occurs for all PE CE pairs that form an MPLS VPN link. The IP addresses taken for this ping are the link endpoint addresses. For example, assume that an MPLS service request has two linked PE1<>CE1 and PE2<>CE2. Then this selection initiates four VRF pings: (PE1, CE1), (PE2, CE2), (PE1, CE2), and (PE2, CE1). When this selection is chosen, then after you click **Start Ping**, you go directly to and receive a result page.
- **Ping Type: Do CE to CE Ping**—When this radio button is chosen, a ping occurs between all CEs that make the endpoint in the service request. When this selection is chosen, then after you click **Start Ping**, you go to Step 3.
- **Two-way Ping** (default: unavailable and deselected)—This check box is only available when you select **Do CE to CE Ping**. When a ping occurs from device1 to device2 and this check box is checked, then a ping from device2 to device1 also occurs.
- **Packet Repeat Count** (default: 5)—This value indicates how many ICMP packets to use for a ping.
- **Datagram size** (default: 100)—This value is the packet size of ICMP used for pinging.

Step 3 For **Do CE to CE Ping**, a MPLS CE Selection window appears.

Step 4 Check the check box next to each row for which you want to select a CE.

Step 5 Click the **Start MPLS CE Ping** button, which becomes enabled.

You receive a MPLS Ping Test Results window.

The buttons at the bottom of the window are as follows:

- **Redo Ping**—When you click this button, you restart all the pings. The parameters used are the same as those specified in the last request.
- **View Job Logs**—When you click this button, you receive logs of all the Prime Fulfillment jobs created for doing ping. The ping application creates one job per selected service request.
- **Refresh**—To selectively refresh, turn off the **Auto Refresh** button and click this button whenever you want to update the results.
- **Close**—Click this button to close the current ping request. You return to the **Monitoring** page.

Note Any column heading in blue indicates that by clicking that column header, you can sort on that column.

Step 6 Click **Close** and you are finished with this Ping session.

SLA

A service-level agreement (SLA) defines a level of service provided by a service provider to any customer. Performance is monitored through the SLA server. Cisco Prime Fulfillment monitors the service-related performance criteria by provisioning, collecting, and monitoring SLAs on Cisco IOS routers that support the Service Assurance Agent (SA Agent) devices. To provision the SLAs and to collect statistics for each SLA, the data collection task requires minimal user input.

Note SLA features are not supported on devices running IOS XR.

The SLA collection task collects the relevant performance data, stores it persistently, aggregates it, and presents useful reports. The SLA collection task collects from the SA Agent MIB on devices. Prime Fulfillment leverages the SA Agent MIB to monitor SLA performance on a 24 x 7 basis. Using the MIB, you can monitor network traffic for the popular protocols:

- Dynamic Host Configuration Protocol (DHCP)
- Domain Name System (DNS)
- File Transfer Protocol (FTP)
- Hyper text Transfer Protocol (HTTP)
- Internet Control Message Protocol Echo (ICMP Echo)
- Jitter (voice jitter)
- Transmission Control Protocol Connect (TCP Connect)
- User Datagram Protocol Echo (UDP Echo).

Note SLA uses the embedded Sybase database, independent of whether you choose Oracle as your database.

Note The SLA operations **Create**, **Delete**, **Enable Probes**, **Disable Probes**, **Enable Traps**, and **Disable Traps** automatically result in the creation of a task, which executes the actual operation. You can view the status of the task by navigating **Inventory > Task Manager > Logs**.

This section explains how to configure SLA probes, collect SLA data, and view SLA reports about these SLA probes.

Before you choose **Inventory > Device Tools > SLA**, implement the setup procedures in the "Setup Prior to Using SLA" section on page 10-3."

Then choose **Inventory > Device Tools > SLA** and you can select one of the following:

- Probes, page 10-6 is the default selection.
- Reports, page 10-18

Setup Prior to Using SLA

SLA is an SNMP activity. Be sure SNMP is enabled and the SNMP settings on the router match the settings in the repository.

When creating an SLA **From MPLS CPE** or **From MPLS PE or MVRF-CE**, the service requests associated with the devices *must* be in the Deployed state.

Setting Up SNMP

To work with Prime Fulfillment, SNMP must be configured on each CPE device in the customer network. In Prime Fulfillment, SNMP is used to:

- collect from the Interface MIB
- provision and collect SLA data.

Two security models are available: SNMPv1/v2c and SNMPv3. Table 10-1 identifies the combinations of security models and levels.

Table 10-1 SNMP Security Models and Levels

Model	Level	Authentication	Encryption	Description
v1/v2c	No Authentication/ No Encryption	Community String	No	Uses a community string match for authentication.
v3	No Authentication/ No Encryption	Username	No	Uses a username match for authentication.
v3	Authentication/ No Encryption	MD5 or SHA	No	Provides authentication based on the HMAC-MD5 or HMAC-SHA algorithms.
v3	Authentication/ Encryption	MD5 or SHA	DES	Provides authentication based on the HMAC-MD5 or HMAC-SHA algorithms, and provides DES 56-bit encryption in addition to authentication based on the CBC-DES (DES-56) standard.

SNMPv3 provides for both security models and security levels. A *security model* is an authentication strategy that is set up for a user and the group in which the user resides. A *security level* is the permitted level of security within a security model. A combination of a security model and a security level determines which security mechanism is employed when handling an SNMP packet.

The security features provided in SNMPv3 are as follows:

- Message integrity—Ensures that a packet has not been tampered with in-transit.
- Authentication—Determines the message is from a valid source.
- Encryption—Encoding the contents of a packet to prevent it from being read by an unauthorized source.

SNMPv3 objects have the following characteristics:

- Each user belongs to a group.
- The group defines the access policy for a set of users and determines the list of notifications its users can receive. The group also defines the security model and security level for its users.
- The access policy defines which SNMP objects can be accessed for reading, writing, or creation.
- SNMPv3 is not supported for Discovery.

Setting Up SNMPv1/v2c on Cisco IOS Routers

To determine whether SNMP is enabled, and to set the SNMP community strings on a Cisco IOS router, perform the following steps for each router:

	Command	Description
Step 1	`Router> enable` `Router> <enable_password>`	Enters enable mode, and then enters the enable password.
Step 2	`Router# show snmp`	Check the output of the **show snmp** command to see whether the following statement is present: "`SNMP agent not enabled.`" If SNMP is not enabled, complete the steps in this procedure.
Step 3	`Router# configure terminal`	Enters global configuration mode.
Step 4	`Router(config)# snmp-server community <userstring> RO`	Sets the community read-only string.
Step 5	`Router(config)# snmp-server community <userstring> RW`	Sets the community read-write string.
Step 6	`Router(config)# Ctrl+Z`	Returns to Privileged Exec mode.
Step 7	`Router# copy running startup`	Saves the configuration changes to NVRAM.

Tip The SNMP community strings defined in Prime Fulfillment for each target device must be identical to those configured on the device.

Setting SNMPv3 Parameters on Cisco IOS Routers

This section describes how to set the SNMPv3 parameters on Cisco IOS routers. SNMPv3 is only supported on IOS crypto images. For Authentication/Encryption, the IOS image must have DES56.

Tip The SNMP users defined in Prime Fulfillment for each target device must be identical to those configured on the device.

To check the existing SNMP configuration, use these commands in the router terminal session:

- **show snmp group**
- **show snmp user**

To set the SNMPv3 server group and user parameters on a Cisco IOS router, perform the following steps:

Note The group must be created first and then the user.

	Command	Description
Step 1	`Router> enable` `Router> <enable_password>`	Enters enable mode, then enter the enable password.
Step 2	`Router# configure terminal`	Enters global configuration mode.

	Command	Description
Step 3	`Router(config)# snmp-server group [<groupname> {v1 | v2c | v3 {auth | noauth | priv}}] [read <readview>] [write <writeview>] [notify <notifyview>] [access <access-list>]`	The **snmp-server group** command configures a new SNMP group or a table that maps SNMP users to SNMP views. Each group belongs to a specific security level. Example: `snmp-server group v3auth v3 auth read v1default write v1default`
Step 4	`Router(config)# snmp-server user <username> [<groupname> remote <ip-address> [udp-port <port>] {v1 | v2c | v3 [encrypted] [auth {md5 | sha} <auth-password> [priv des56 <priv-password>]] [access <access-list>]`	The **snmp-server user** command configures a new user to an SNMP group. Example: `snmp-server user user1 v3auth v3 auth md5 user1Pass`
Step 5	`Router(config)# Ctrl+Z`	Returns to Privileged Exec mode.
Step 6	`Router# copy running startup`	Saves the configuration changes to NVRAM.

Manually Enabling RTR Responder on Cisco IOS Routers

Note SNMP must be configured on the router.

To manually enable an RTR Responder on a Cisco IOS router, execute the following steps:

	Command	Description
Step 1	`Router> enable` `Router> <enable_password>`	Enters enable mode, and then enters the enable password.
Step 2	`Router# configure terminal`	Enters the global configuration mode.
Step 3	`Router(config)# rtr responder`	Enables the SA responder on the target router of SA Agent operations.
Step 4	`Router(config)# Ctrl+Z`	Returns to Privileged Exec mode.
Step 5	`Router# copy running startup`	Saves the configuration changes to NVRAM.

Probes

When you choose **Inventory > Device Tools > SLA**, a SLA Probes window appears.

The default button that is enabled is **Create** and from the **Create** drop-down list, you can choose to create SLA probes **From Any SA Agent Device(s)**; **From MPLS CPE**; or **From MPLS PE or MVRF-CE**. However, if you select one or more existing probes by clicking the row(s) of existing probe(s), then you have access to the other buttons, **Details**, **Delete**, **Enable**, and **Disable**. For **Enable** and **Disable**, the drop-down list contains options to enable or disable SLA **Probes** and SLA **Traps**.

The explanations of the buttons and subsequent drop-down lists is given as follows:

- Create Common Parameters, page 10-7—This section explains the SLA common parameters for all of the probe creation types: **From Any SA Agent Device(s)**; **From MPLS CPE**; or **From MPLS PE or MVRF-CE**.
- Create From Any SA Agent Device(s), page 10-9—This section explains how to create probes from any SA Agent device(s) and begins after creating common parameters.

- Create from MPLS CPE, page 10-11—This section explains how to create probes from an MPLS CPE and begins after creating common parameters.
- Create From MPLS PE or MVRF-CE, page 10-13—This section explains how to create probes from an MPLS PE or MVRF-CE and begins after creating common parameters.
- Protocols, page 10-14—This section is common Probes information for each of the **Create** paths.
- Details, page 10-16—This section gives details about a specified probe.
- Delete, page 10-16—This section explains how to delete a probe.
- Enable Probes, page 10-17—This section explains how to enable the Probe and change its status from Created to Active state.
- Enable Traps, page 10-17—This section explains how to enable traps.
- Disable Probes, page 10-17—This section explains how to disable the Probe and change its status from Active to Disabled.
- Disable Traps, page 10-18—This sections explains how to disable traps.

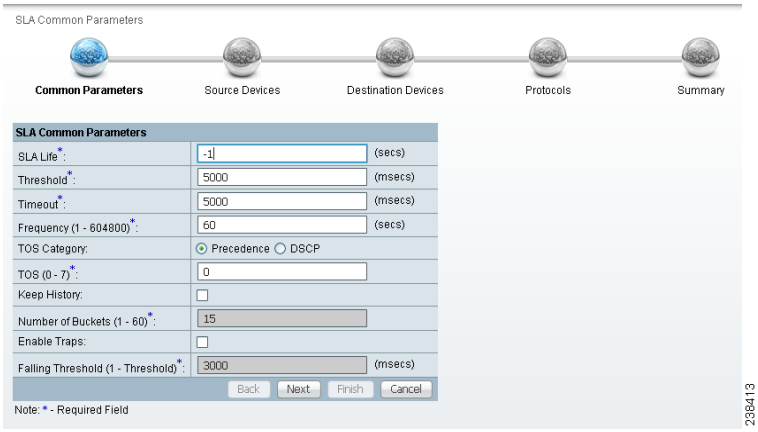
Create Common Parameters

When you choose **Inventory > Device Tools > SLA**, the default is the **Probes** page with only the **Create** button enabled. From the **Create** drop-down list, you can choose **From Any SA Agent Device(s)**, **From MPLS CPE**, or **From MPLS PE or MVRF-CE**. The first window to appear in all ways of creation is specified here. Then you proceed to the specific creation type you have chosen.

Follow these steps:

Step 1 Choose **Create**, and the window to appear is as shown in Figure 10-1.

Figure 10-1 SLA Common Parameters



Accept the defaults or change the information in the fields of the common SLA parameters, as follows, and then click **Next**:

- **SLA Life** (required)—The number of seconds that the probe is active (with the maximum value of a 32-bit integer in seconds). If the value is set to **-1**, the typical and default value, the probe is active forever.

- **Threshold** (required)—An integer that defines the threshold limit in milliseconds. When this threshold is exceeded and traps are enabled, a trap is sent. The maximum value is the maximum value of a 32-bit integer. If the service affecting agent (SA Agent) operation time exceeds this limit, the threshold violation is recorded by the SA Agent. The value for **Threshold** must not exceed the value for **Timeout**. The default value is **5000**.
- **Timeout** (required)—Duration in milliseconds to wait for an SA Agent operation completion. The value for **Timeout** must be less than or equal to the value for **Frequency** and greater than or equal to the value for **Threshold**. The default value is **5000**
- **Frequency (0 - 604800)** (required)—Duration in seconds between initiating each SA Agent operation. The value for **Frequency** must be greater than or equal to the value for **Timeout**. The default value is **60**.
- **TOS Category** (default: **Precedence**)—If you choose the **Precedence** radio button for **TOS Category**, you have one set of type of service (TOS) values. If you choose the **DSCP** radio button for **TOS Category**, you have a different set of TOS values.
- **TOS** (required)—An integer. The range and meanings of the values depend on whether the radio button in the **TOS Category** is set to **Precedence** (values: 0 to 7) or **DSCP** (values: 0 to 63).
  - When the **TOS Category** is set to **Precedence**, the valid values are **0** to **7**. These values represent the three most significant bits of the ToS field in an IP header. The default value is **0**. The meanings of the **Precedence** values are specified in Table 10-2.

Note Type of Service does not apply to the **DNS** and **DHCP** types of SLA probes. Prime Fulfillment ignores any ToS value set for these two types of SLA probes. For example, if you first choose a ToS value of 5, then choose the **DNS**, **DHCP,** and **ICMP Echo** protocols for an SLA probe, Prime Fulfillment applies the selected ToS value to the **ICMP Echo** probe only.

Table 10-2 Meanings of Precedence Values

ToS Value	Binary Value	Meaning
7	111	Network Control
6	110	Internetwork Control
5	101	CRITIC/ECP
4	100	Flash Override
3	011	Flash
2	010	Immediate
1	001	Priority
0	000	Routine

  - When the **TOS Category** is set to **DSCP**, the valid values are **0** to **63**. These values represent the six most significant bits of this ToS field in an IP header. The default value is **0**. The interpretation of these **TOS** values is user specified.

Note Prime Fulfillment maps the 0 - 7 PRECEDENCE values to the three most significant ToS bits by left-shifting the value by five positions. Similarly, the 0 - 63 DSCP values are left-shifted by two positions.

- **Keep History (default: unchecked)**—If you check the **Keep History** check box, you indicate to keep the recent History Table on the router. Specifically, it is kept in the SA Agent MIB that keeps the raw round-trip time (RTT) SLA measurement. This selection also enables you to indicate the **Number of Buckets** of raw history data to keep. If you leave the default of an unchecked check box for **Keep History**, no raw history data is kept. **Keep History** is not supported for **HTTP** and **Jitter**.
- **Number of Buckets (1 - 60)** (required)—The default is **15** when the **Keep History** check box is checked. The range is 1 to 60 and indicates the number of most recent raw data entries to be kept in the raw history data. When the specified **Number of Buckets** is surpassed, removal of buckets starts with the oldest bucket to keep only the number of raw data entries specified.
- **Enable Traps** (default: unchecked, which means No)—If you check the **Enable Traps** check box, the created SLA is configured to send three types of traps. This selection also enables you to indicate the **Falling Threshold**. If you leave the **Enable Traps** check box unchecked, the traps are disabled on the SLAs created in this task.
- **Falling Threshold (1 - Threshold)** (required)—The default is **3000** in milliseconds when the **Enable Traps check box is checked**. The range is **1** to the **Threshold** value in milliseconds. When traps are enabled and the delay meets the specified number of milliseconds, a trap is sent.

Step 2 Next you proceed to Create From Any SA Agent Device(s), page 10-9, Create from MPLS CPE, page 10-11, or Create From MPLS PE or MVRF-CE, page 10-13.

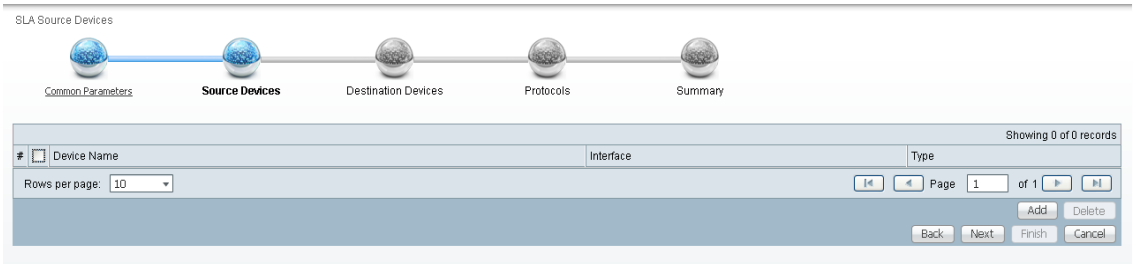
Create From Any SA Agent Device(s)

After you have completed the steps in Create Common Parameters, page 10-7, follow these steps:

Note IP connectivity must be available between the SA Agent devices.

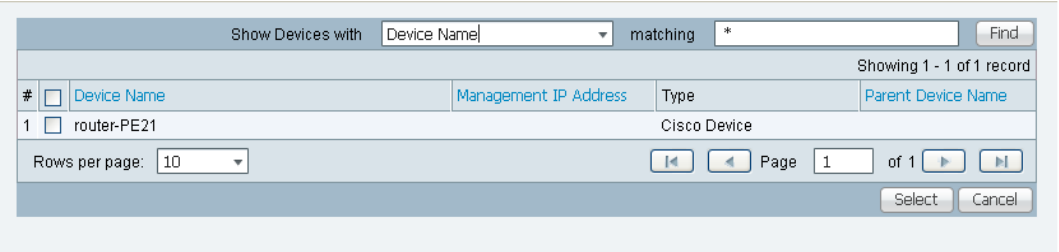
Step 1 The next window to appear is as shown in Figure 10-2.

Figure 10-2 SLA Source Devices



Step 2 Click the **Add** button and a window appears as shown in Figure 10-3, which lists all the devices in the database that have a minimum of one interface. Check the check box next to each row for the device you want to select, then click **Select**.

Figure 10-3 SLA Devices > Add



You return to Figure 10-2 and the newly added source device(s) appear. The information about this source device is specified in the following columns:

- **Device Name**—You can click this heading and the device names are organized alphabetically.
- **Interface**—You can click **Select** and from the resulting window, you can update the IP address. Select one radio button for an interface and click **Select** and the IP address changes in Figure 10-2.
- **Type**—Gives you the type of the source device.

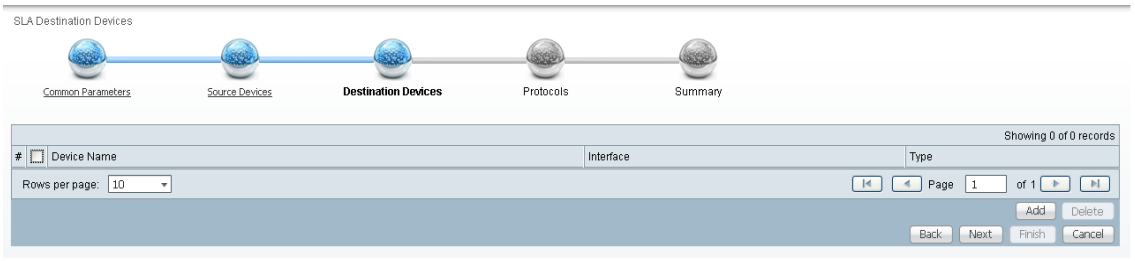
Step 3 You can repeat Step 2 to to add more devices, or you can delete any of the currently selected source devices. To delete, check the check box next to each row for the device you want to delete and then click **Delete**.

Note There is no second chance for deleting source devices. There is no confirm window.

Step 4 Click **Next**.

The next window to appear is as shown in Figure 10-4.

Figure 10-4 SLA Destination Devices



Step 5 Click the **Add** button and a window appears as shown in Figure 10-3. Check the check box next to each row for the device you want to select. Then click **Select**.

Step 6 You return to Figure 10-4 and the newly added destination device(s) appear. The information about this destination device is specified in the following columns:

- **Device Name**—You can click this heading and the device names are organized alphabetically.
- **Interface**—You can click **Select** and from the resulting window, you can update the IP address. Select one radio button for an interface and click **Select** and the IP address changes in Figure 10-4.
- **Type**—Gives you the type of the source device.

Step 7 You can repeat Step 5 to Step 6 to add more devices, or you can delete any of the currently selected destination devices. To delete, check the check box next to each row for the device you want to delete and then click **Delete**.

Note There is no second chance for deleting destination devices. There is no confirm window.

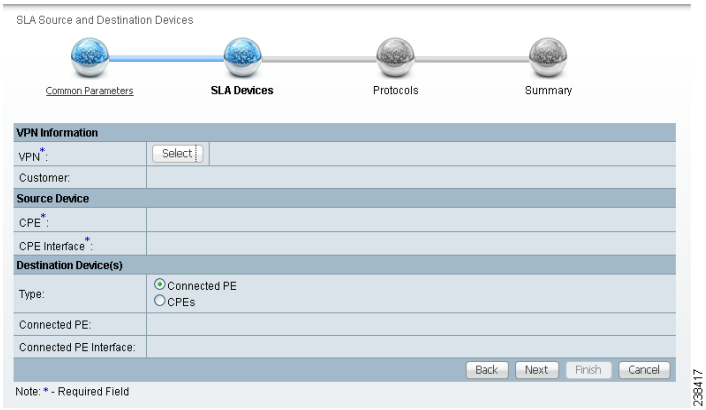
Step 8 Click **Next**. Proceed to the "Protocols" section on page 10-14."

Create from MPLS CPE

After you have completed the steps in Create Common Parameters, page 10-7, follow these steps:

Step 1 Complete the steps in the "Create Common Parameters" section on page 10-7 and the next window to appear is as shown in Figure 10-5.

Figure 10-5 SLA CPE Parameters



Step 2 Click the **Select** button for **VPN** and a window appears, which lists all the VPNs in the database.

Step 3 Click the radio button for the VPN you want to select. Then click **Select**. You return to Figure 10-5 and the newly added VPN and Customer information appear and a **Select** button appears for **CPE**. You can change the VPN by repeating Step 2.

Step 4 Click the **Select** button for **CPE** and a window appears which lists the CPEs associated with the selected VPN. Click the radio button for the CPE you want to select. Then click **Select**.

Step 5 You return to Figure 10-5 and the newly added **CPE** and its first interface appear and a **Select** button appears for **CPE Interface**. You can change the CPE by repeating Step 4.

Step 6 If you want to change the default **CPE Interface** information that appears, click **Select** and you receive a window appears.

Step 7 Click the radio button next to the row for the interface you want to select. Then click **Select**. You return to Figure 10-5 and the newly added **CPE Interface** appears.

Step 8 You can change the CPE Interface by repeating Step 6.

Step 9 You can keep the default **Type**, by leaving the radio button for **Connected PE** chosen, which creates an SLA between the CPE and its directly connected PE, or you can select the radio button for **CPEs** in the same VPN. If you keep the default of **Connected PE**, proceed to Step 10. If you click the **CPEs** radio button, proceed to Step 14.

Step 10 Click **Select** for **Connected PE Interface** and a window appears.

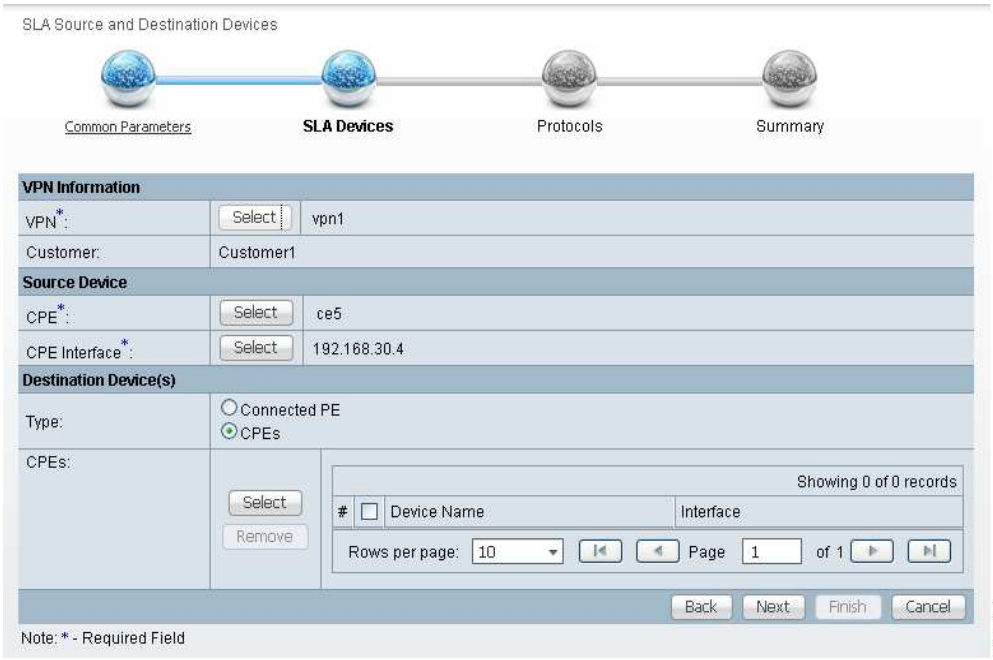
Step 11 Click the radio button next to the row for the interface you want to select. Then click **Select**.

Step 12 You return to Figure 10-5 and the newly added **Connected PE Interface** appears. You can change the Connected PE Interface by repeating Step 10.

Step 13 Click **Next** and proceed to the "Protocols" section on page 10-14.

Step 14 When you click **CPEs**, the window is as shown in Figure 10-6, "CPEs."

Figure 10-6 CPEs



Step 15 Click the **Select** button for **CPEs** and a window appears which lists all the CPEs associated with the specified VPN in the database.

Step 16 Check the check box next to the row(s) for the CPE(s) you want to select. Then click **Select**.

Note Do *not* add a device chosen as a **Source Device** to **Destination Device(s)**.

You return to Figure 10-6 and the newly added **Device Name** appears.

Step 17 Click **Select** in the **Interface** column and a window appears.

Step 18 Click the radio button next to the row for the CPE you want to select. Then click **Select**.

Step 19 You return to Figure 10-6 and the newly added **CPE Interface** appears. You can change the CPE Interface by repeating Step 17.

Step 20 Check the check box next to each row for the Devices you want to remove. Then click the **Remove** button and a window as shown in Figure 10-6 appears without the removed Device(s).

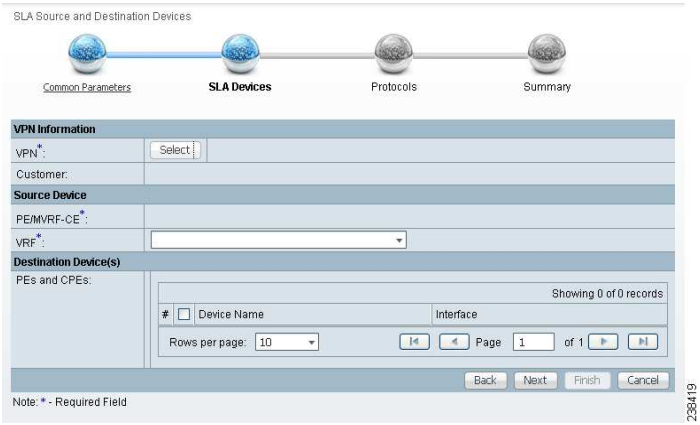
Step 21 When Figure 10-6 reflects what you want, click **Next** and proceed to the "Protocols" section on page 10-14.

Create From MPLS PE or MVRF-CE

After you have completed the steps in Create Common Parameters, page 10-7, follow these steps:

Step 1 Complete the steps in the "Create Common Parameters" section on page 10-7 and the next window to appear is as shown in Figure 10-7, "SLA Source and Destination Devices."

Figure 10-7 SLA Source and Destination Devices



Step 2 Click the **Select** button for **VPN** and a window appears which lists all the VPNs in the database. Click the radio button next to the row for the VPN you want to select.

Step 3 Then click **Select**.

Step 4 You return to Figure 10-7 and the newly added VPN and Customer information appears. You can change the VPN and Customer by repeating Step 2.

Step 5 Click the new **Select** button for **PE/MVRF-CE** and you receive a drop-down list from which you can choose **PE** or **MVRF-CE**. If you choose **PE**, a window appears, which lists all the PEs associated with the selected VPN. If you choose MVRF-CE, a window appears, which lists all the MVRF-CEs associated with the selected VPN. Click the radio button next to the row for the PE or MVRF-CE you want to select. Then click **Select** or **OK**.

Step 6 You return to Figure 10-7 and the newly added PE or MVRF-CE information appears. You can change this selection by repeating Step 5.

Step 7 If in Step 5 you chose MVRF-CE information, you can click the **VRF** drop-down list.

Step 8 Click the new **Select** button for **Destination Device(s)—PEs and CPEs** and from a drop-down list, choose **PEs** or **CPEs**. If you choose **PEs**, a window appears, which lists all the PE Interfaces in the database. If you choose **CPEs**, a window appears, which lists all the CPE Interfaces in the database. Click the radio button next to the row for the Device Interface you want to select. Then click **Select**.

Note Do *not* add a device chosen as a **Source Device** to **Destination Device(s)**.

Step 9 You return to Figure 10-7 and you receive interface information. Click **Select** and you get a window from which you can click a radio button next to a different interface. Click **Select** and the new interface replaces the old interface. You can change the Interface by repeating this step.

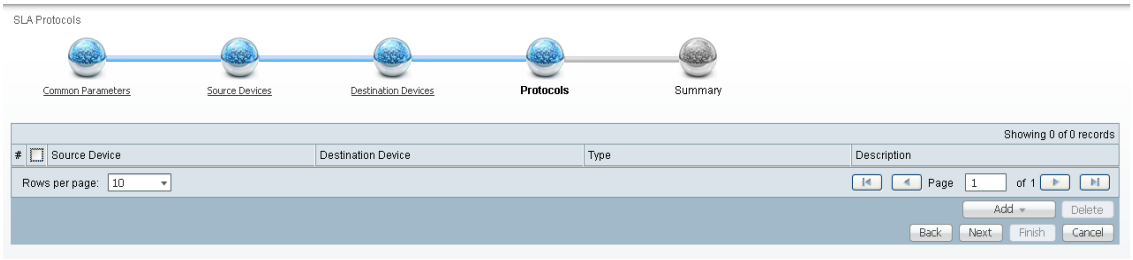
Step 10 Click **Next** and proceed to the "Protocols" section on page 10-14.

Protocols

You choose this location after you have completed all the steps in one of the **Create** functions: Create Common Parameters, page 10-7; Create from MPLS CPE, page 10-11; or Create From MPLS PE or MVRF-CE, page 10-13. Follow these steps:

Step 1 Complete the steps in the "Create Common Parameters" section on page 10-7 and the next window to appear is as shown in Figure 10-8.

Figure 10-8 Protocols



Step 2 Click the **Add** drop-down list and select:

- **ICMP Echo** (only available if destination devices are available)—Proceed to Step 3.
- **TCP Connect** (not available for Create From MPLS PE or MVRF-CE; for all the other Creates, TCP Connect is only available if destination devices are available)—Proceed to Step 4.
- **UDP Echo** (only available if destination devices are available)—Proceed to Step 5.
- **Jitter** (only available if destination devices are available)—Proceed to Step 6.
- **FTP** (not available for Create from MPLS PE or MVRF-CE)—Proceed to Step 7.
- **DNS** (not available for Create from MPLS PE or MVRF-CE)—Proceed to Step 8.
- **HTTP** (not available for Create from MPLS PE or MVRF-CE)—Proceed to Step 9.
- **DHCP** (not available for Create from MPLS PE or MVRF-CE)—Proceed to Step 10.

Step 3 From Step 2, if you chose **ICMP Echo**, a Protocol ICMP Echo window appears.

Enter the required information as follows, click **OK**, and then proceed to Step 11.

- Request Size (0 - 16384) (required)—Number that represents the number of octets (in bytes) to be placed into the data portion of the packet. The default is **28**.

Step 4 From Step 2, if you chose **TCP Connect**, a Protocol TCP Connect window appears.

Enter the required and optional information as follows, click **OK**, and then proceed to Step 11.

- **Destination Port (1 - 65535)** (required)—Port number on the target where the monitoring packets is sent. If you do not specify a specific port, port **23** is used.
- **Request Size (1 - 16384)** (optional)—Number that represents the number of octets (in bytes) to be placed into the data portion of the packet. The default is **1**.

Step 5 From Step 2, if you chose **UDP Echo**, a Protocol UDP Echo window appears.

Enter the required and optional information as follows, click **OK**, and then proceed to Step 11.

- **Destination Port (1 - 65535) (required)**—Port number on the target to where the monitoring packets are sent. If you do not specify a specific port, port **7** is used.

- **Request Size (4 - 8192)** (optional)—Number that represents the number of octets (in bytes) to be placed into the data portion of the packet. The default is **16**.

Step 6 From Step 2, if you chose **Jitter**, a Protocol Jitter window appears.

Enter the required and optional information as follows, click **OK**, and then proceed to Step 11.

- **Destination Port (1 - 65535)** (required)—Port number on the target where the monitoring packets are sent. If you do not specify a specific port, port **8000** is used.
- **Request Size (16 - 1500)** (optional)—Number that represents the number of octets (in bytes) to be placed into the data portion of the packet. The default is **32**.
- **Number of Packets (1 - 1000)** (optional)—Integer that represents the number of packets that must be transmitted. The default value is **10**.
- **Interval (1 - 1000)** (optional)—Integer, **1** to **1,000**, that represents the inter-packet delay between packets in milliseconds.The default value is **20**.

Step 7 From Step 2, if you chose **FTP**, a Protocol FTP window appears.

Enter the required and optional information as follows, click **OK**, and then proceed to Step 11.

- **User Name** (optional)**—**If blank, anonymous is used.
- **Password** (optional)—If blank, test is used.
- **Host IP Address** (required)—Enter the IP address for File Transfer Protocol (FTP).
- **File Path** (required)—Enter the path of the file you want to FTP on the FTP server.

Step 8 From Step 2, if you chose **DNS**, a Protocol DNS window appears.

Enter the required information as follows, click **OK**, and then proceed to Step 11.

- **Name Server** (required)—String that specifies the IP address of the name server. The address is in dotted IP address format.
- **Name to be Resolved** (required)—String that is either the name or the IP address that is to be resolved by the DNS server. If the string is a name, the length is 255 characters. If the string is an IP address, it is in dotted IP address format.
- **Request Size** (0 - 16384) (required)—Number that represents the number of octets (in bytes) to be placed into the data portion of the packet. The default is **1**.

Step 9 From Step 2, if you chose **HTTP**, a Protocol HTTP window appears.

Enter the optional and required information as follows, click **OK**, and then proceed to Step 11.

- **Version** (default: 1.0)—String that specifies the version of the HTTP server. Do not change this. Prime Fulfillment only supports version 1.0.
- **URL** (required)—String that represents the URL to which an HTTP probe should communicate, *HTTPServerName*[*/directory*]*/filename* or *HTTPServerAddress*[*/directory*]*/filename* (for example: **http://www.cisco.com/index.html** or **http://209.165.201.22/index.html**). If you specify the *HTTPServerName*, the **Name Server** is required. If you specify the *HTTPServerAddress*, the **Name Server** is not required.
- **Cache** (default: selected, which means Yes)—For an unchecked check box, the HTTP request should not download cached pages. For a checked check box, the HTTP request downloads cached pages if available, otherwise the request is forwarded to the HTTP server.
- **Proxy Server** (optional)—String that represents the proxy server information (with a maximum of 255 characters). The default is the null string.
- **Name Server** (optional, dependent on the **URL** setting)—String that specifies the IP address of the name server. The address is in dotted IP address format.

- **Operation** (default: HTTPGet)—If you want **HTTPRaw**, which represents the HTTP request with user defined payload, instead of the default **HTTPGet** which represents the HTTP get request, use the drop-down list and make that choice.
- **Raw Request** (required if the **Operation** is **HTTPRaw**; not available if the **Operation** is **HTTPGet**)—String that is only needed if the **Operation** is **HTTPRaw**. It allows you to invoke other types of HTTP operations other than the simple GET operation.
- **Request Size** (1 - 16384) (required)—Number that represents the number of octets (in bytes) to be placed into the data portion of the packet. The default is **28**.

Step 10 From Step 2, if you chose **DHCP**, a Protocol DHCP window appears.

Enter the required information as follows, click **OK**, and then proceed to Step 11.

- **Destination IP Address** (required)

Step 11 You return to Figure 10-8 and additional columns of information now appear based on the Protocol information you provided. Before you click **Next** to proceed, determine if you want to **Add** more protocols, in which case repeat Step 2 to Step 10, or **Delete** any of the currently selected protocols, in which case, click **Delete** and proceed much as in Step 2 to Step 10 to now delete protocols.

Note There is no second chance for deleting destination devices. There is no confirm window.

Step 12 The next window to appear is a Probe Creation Task Summary window that shows the **Description** (date and time created), **Common Parameters**, **Source Devices**, **Destination Devices**, and **Protocols** that you have defined. If all exists the way you want it, click **Finish**. Otherwise, click **Back** and make corrections.

Details

When you choose **Inventory > Device Tools > SLA**, you can get details by following these steps:

Step 1 Select an existing probe by checking the corresponding check box for which you want details. Then you have access to the **Details** button.

Step 2 After you click the **Details** button, you receive a SLA Probes Details window. This includes the **Common Attributes** information defined when you first **Create** and the **Protocol Specific Attributes** information defined in the section Protocols.

Step 3 Click **OK** to return. You can continue to select more **Details** or complete another function.

Delete

When you choose **Inventory > Device Tools > SLA**, you can delete probes from the list by following these steps:

Step 1 Select one or more existing probes by checking the check box(es) for the row(s) of existing probe(s). Then you have access to the **Delete** button.

Step 2 After you click the **Delete** button, a confirmation window appears.

Step 3 Click **OK** if it reflects what you want to delete or click **Cancel** if it does not.

Note After the probe is deleted, it is deleted from the probe list page but still remains in the database.

You return to window with updated information.

Enable Probes

When you choose **Inventory > Device Tools > SLA**, you can enable probes by following these steps:

Step 1 Select one or more existing probes by checking the check box(es) for the row(s) of existing probe(s). Then you have access to the **Enable** button. From the **Enable** drop-down list, you have access to **Probes**.

Step 2 After you choose **Enable > Probes,** a confirm enable probes window appears.

Step 3 Click **OK** if it reflects the probes you want to enable or click **Cancel** if it does not.

If this was successful, you receive a Status window with a green check mark for **Succeeded**. The Status column is set to **Active** when the probe is created successfully on the router.

Enable Traps

When you choose **Inventory > Device Tools > SLA**, you can enable traps by following these steps:

Step 1 Select one or more existing probes by checking the check box(es) for the row(s) of existing probe(s). Then you have access to the **Enable** button. From the **Enable** drop-down list, you have access to **Traps**.

Step 2 After you choose **Enable > Traps**, a confirm enable traps window appears. All the traps have 3000 ms as the falling threshold set automatically

Step 3 Click **OK** if it reflects the traps you want to enable or click **Cancel** if it does not.

If this was successful, you receive a Status window with a green check mark for **Succeeded**. The Traps Enabled column is set to **yes** when the probes on the router are successfully changed.

Disable Probes

When you choose **Inventory > Device Tools > SLA**, you can use **Disable Probes** to delete probes on the devices. Follow these steps:

Step 1 Select one or more enabled probes by checking the check box(es) for the row(s) of existing probe(s). Then you have access to the **Disable** button. From the **Disable** drop-down list, you have access to **Probes**.

Step 2 After you choose **Disable > Probes,** a confirm disable probes window appears.

Step 3 Click **OK** if it reflects the probes you want to disable or click **Cancel** if it does not.

If this was successful, you receive a Status window with a green check mark for **Succeeded**, and the probe's status becomes Disabled when the probe on the router is successfully removed.

Disable Traps

When you choose **Inventory > Device Tools > SLA**, you can disable traps by following these steps:

Step 1 Select one or more existing probes by checking the check box(es) for the row(s) of existing probe(s). Then you have access to the **Disable** button. From the **Disable** drop-down list, you have access to **Traps**.

Step 2 After you choose **Disable > Traps,** a confirm disable traps window appears.

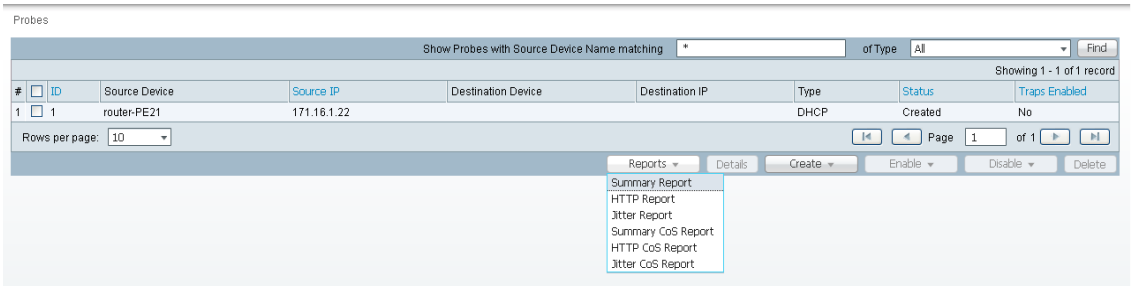
Step 3 Click **OK** if it reflects the traps you want to disable or click **Cancel** if it does not.

If this was successful, you receive a Status window with a green check mark for **Succeeded**. The traps are disabled when the probes on the router are successfully changed.

Reports

When you choose **Inventory > Device Tools > SLA**, you receive a window as shown in Figure 10-9.

Figure 10-9 SLA Reports



You can then click on any of the following choices and receive that report

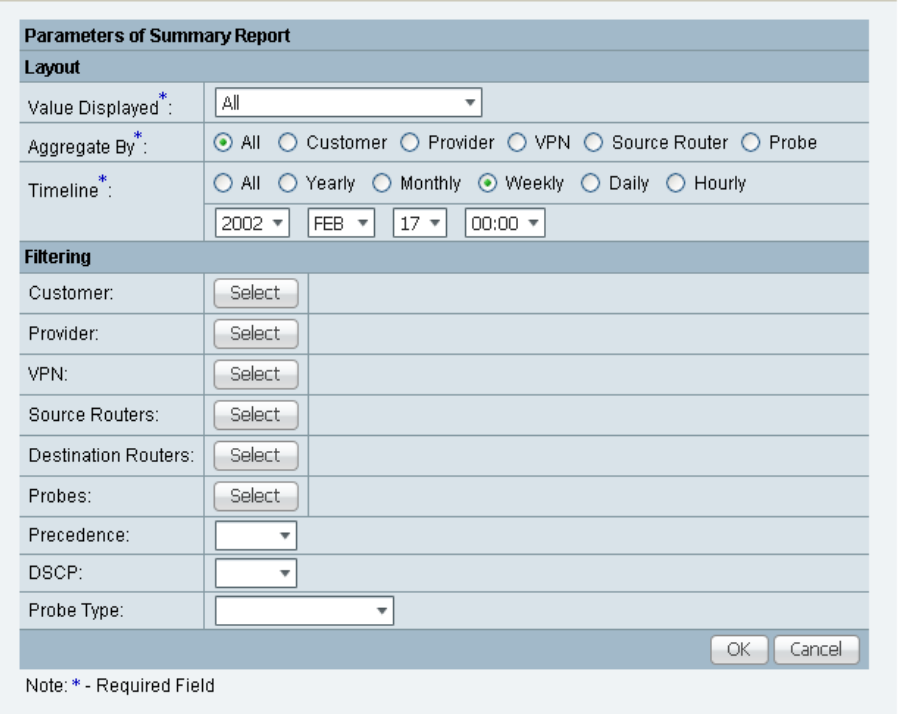
- Summary Report, page 10-18—This report summarizes all the information other than for HTTP and Jitter (ICMP Echo, TCP Connect, UDP Echo, FTP, DNS, and DHCP).
- HTTP Report, page 10-21—This is a summary report for HTTP information.
- Jitter Report, page 10-21—This is a summary report for Jitter information.
- Summary CoS Report, page 10-22—This report a summary report for Class of Service (CoS) other than for HTTP and Jitter (ICMP Echo, TCP Connect, UDP Echo, FTP, DNS, and DHCP).
- HTTP CoS Report, page 10-23—This report is for HTTP CoS information.
- Jitter CoS Report, page 10-23—This report is for Jitter CoS information.

Summary Report

From Figure 10-9, choose **Summary Report** and follow these steps:

Step 1 Choose **Summary Report,** and the resulting window is shown in Figure 10-10.

Figure 10-10 Parameters of Summary Report



Step 2 For Figure 10-10, fill in the **Layout** fields, as follows:

- **Value Displayed** (required) (default: **All**) Click the drop-down list and choose one of the following:
  - **All**—To display all the values.
  - **Connections (#)**—To display the number of connections.
  - **Timeouts (#)**—To display the number of timeouts.
  - **Connectivity (%)**—To display connectivity as a percentage.
  - **Threshold Violations (%)**—To display threshold violations as a percentage.
  - **Max Delay (ms)**—To display the maximum delay in milliseconds.
  - **Min Delay (ms)**—To display the minimum delay in milliseconds.
  - **Avg Delay (ms)**—To display the average delay in milliseconds.
- **Aggregate By** (required) (default: All) Click the radio button for how you want to aggregate the data, by **All**, **Customer**, **Provider**, **VPN, Source Router**, or **Probe**.
- **Timeline** (required) (default: **Weekly**; starting with midnight of the first day of the selected week) Click the radio button for the report data that you want to display, **All** data; **Yearly** data; **Monthly** data; **Weekly** data; **Daily** data; or **Hourly** data. Also click the drop-down lists for the year, month, day of the month, and time of day for which to start the report.

Step 3 For Figure 10-10, fill in the **Filtering** fields, as follows.

Note The report contains only the data that fulfills all the conditions in the filtering fields (all the conditions are ANDed together).

- **Customer** (optional)—Click the **Select** button and from the resulting list of Customers, filter the list if you choose. From the listed Customers, click the radio button for the Customer for which you want this SLA report. Then click Select. The result is that you return to Figure 10-10 and the selected customer is listed for **Customer**. You can repeat this process if you want to change your selection.
- **Provider** (optional)—Click the **Select** button and from the resulting list of Providers, filter the list if you choose. From the listed Providers, click the radio button for the Provider for which you want this SLA report. Then click **Select**. The result is that you return to Figure 10-10 and the selected provider is listed for **Provider**. You can repeat this process if you want to change your selection.
- **VPN** (optional)—Click the **Select** button and from the resulting list of VPNs, filter the list if you choose. From the listed VPNs, click the radio button for the VPN for which you want this SLA report. Then click **Select**. The result is that you return to Figure 10-10 and the selected VPN is listed for **VPN**. You can repeat this process if you want to change your selection.
- **Source Routers** (optional)—Click the **Select** button and from the resulting list of devices, filter the list if you choose. From the listed devices, check the check box(es) for device(s). Then click **Select**. The result is that you return to Figure 10-10 and **Source Routers** contains the selected device(s). You can repeat this process if you want to change your selection.
- **Destination Routers** (optional)—Click the **Select** button and from the resulting list of devices, filter the list if you choose. From the listed devices, check the check box(es) for device(s). Then click **Select**. The result is that you return to Figure 10-10 and **Destination Routers** contains the selected device(s). You can repeat this process if you want to change your selection.
- **Probes** (optional)—Click the **Select** button and from the resulting list of source probes, filter the list if you choose. From the listed source probes, check the check box(es) for source probe(s). Then click **Select**. The result is that you return to Figure 10-10 and **Probes** contains the selected source probe(s). You can repeat this process if you want to change your selection.
- **Precedence** (default: **All**)—Click the drop-down list to select the other **Precedence** TOS choices, **0** to **7**. These values represent the three most significant bits of the ToS field in an IP header. The meanings of the **Precedence** values are specified in Table 10-2.

Note Prime Fulfillment maps the 0 - 7 PRECEDENCE values to the three most significant ToS bits by left-shifting the value by five positions.

Note Type of Service does not apply to the **DNS** and **DHCP** types of SLA probes. Prime Fulfillment ignores any ToS value set for these two types of SLA probes. For example, if you first choose a ToS value of 5, then choose the **DNS**, **DHCP,** and **ICMP Echo** protocols for an SLA probe, Prime Fulfillment applies the selected ToS value to the **ICMP Echo** probe only.

- **DSCP** (default: **All**)—Click the drop-down list to select the other **DSCP TOS** choices, **0** to **63**. These values represent the six most significant bits of this ToS field in an IP header. The interpretation of these **TOS** values is user specified.

Note Prime Fulfillment maps the 0 - 63 DSCP values to the six most significant ToS bits by left-shifting the values by two positions.

- **Probe Type** (default: **All**)—Click the drop-down list to select from the following types of probes: ICMP Echo; UDP Echo; TCP Connect; HTTP; DNS; Jitter; DHCP; FTP.

Note These probe types are explained in detail in the "Protocols" section on page 10-14.

Step 4 Click **OK** in Figure 10-10 after you have the information you want.

The result is a Summary Report with the selections you made listed. You can **Modify**, **Refresh**, **Print**, or **Close** this report with the appropriate button.

Note If you choose **Modify**, you receive a window such as Figure 10-10 in which you can modify your selections as explained in the previous steps.

HTTP Report

From Figure 10-9, choose **HTTP Report** and proceed similarly to the "Summary Report" section on page 10-18, with the following exceptions:

- **Value Displayed** has different drop-down choices.
- There is no **Destination Routers** selection.
- There is no **Probe Type** drop-down list in the equivalent of Figure 10-10, because the probe type is automatically **HTTP**. The result is an HTTP Report.

Jitter Report

From Figure 10-9, choose **Jitter Report** and proceed similarly to the "Summary Report" section on page 10-18, with the following exceptions:

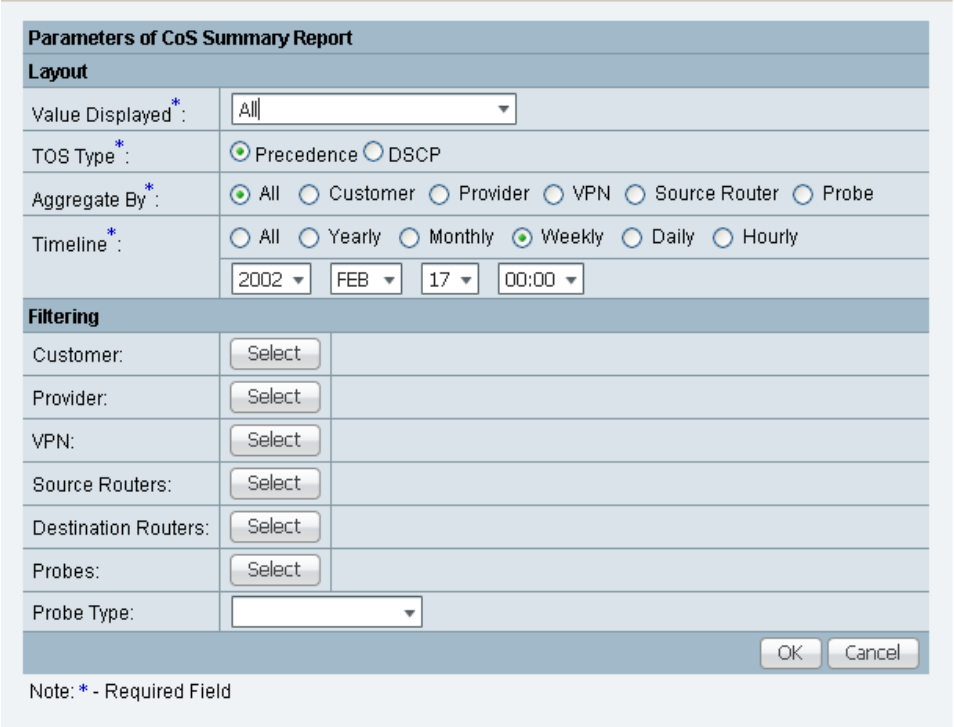
- **Value Displayed** has different drop-down choices.
- There is no **Destination Routers** selection.
- There is no **Probe Type** drop-down list in the equivalent of Figure 10-10, because the probe type is automatically **Jitter**. The result is a Jitter Report.

Summary CoS Report

From Figure 10-9, choose **Summary CoS Report** for a summary of the Class of Service (CoS) reports, which are based on the TOS values of the SLA probes, and follow these steps:

Step 1 Choose **Summary CoS Report**, and the resulting window is shown in Figure 10-11.

Figure 10-11 Parameters of CoS Summary Report



Step 2 For Figure 10-11, fill in the **Layout** fields, as shown in Step 2 of the "Summary Report" section on page 10-18, with the following exception. After **Value Displayed** and before **Aggregate By**, select the radio button **Precedence** (default) or **DSCP** for the new **TOS Type**. The explanations are given in the Filtering section, Step 3 of the "Summary Report" section on page 10-18.

Step 3 For Figure 10-11, fill in the **Filtering** fields, as shown in Step 3 of the "Summary Report" section on page 10-18, with the exception that there are no **Precedence** or **DSCP** drop-down lists. They are now in the **Layout** fields, as explained in Step 2 in this section.

Step 4 Click **OK** in Figure 10-11 after you have the information you want.

The result is a CoS Summary Report with the selections you made listed. You can **Modify**, **Refresh**, **Print**, or **Close** this report with the appropriate button.

Note If you choose **Modify**, you receive a window such as Figure 10-11 in which you can modify your selections as explained in the previous steps.

HTTP CoS Report

From Figure 10-9, choose **HTTP Report** and proceed exactly as in the "Summary CoS Report" section on page 10-22, with the following exceptions:

- **Value Displayed** has the same drop-down choices as **HTTP Report**.
- There is no **Destination Routers** selection.
- There is no **Probe Type** drop-down list in the equivalent of Figure 10-11, because the probe type is automatically **HTTP CoS**. The result is a CoS HTTP Report. This CoS HTTP report is based on the TOS values of the SLA probes.

Jitter CoS Report

From Figure 10-9, choose **Jitter Report** and proceed exactly as in the "Summary CoS Report" section on page 10-22, with he following exceptions:

- **Value Displayed** has the same drop-down choices as **Jitter Report**.
- There is no **Destination Routers** selection.
- There is no **Probe Type** drop-down list in the equivalent of Figure 10-11, because the probe type is automatically **Jitter CoS**. The result is a CoS Jitter Report. This CoS Jitter report is based on the TOS values of the SLA probes.

Task Manager

Cisco Prime Fulfillment provides a Task Manager that allows you to view pertinent information about both current and expired tasks of all types, and to create and schedule new tasks, delete specified tasks, and delete the active and expired tasks.

This section contains the following subsections:

Tasks

This section contains the following topics:

Starting Task Manager

To start Task Manager, click **Operate > Tasks > Task Manager**. The Tasks list page appears.

The Tasks window displays information about each task by **Task Name**, **Type**, **Targets**, **Schedules** date and time, the **User Name** who created those tasks, and the date **Created on**. To view, schedule, or delete the listed tasks, check the corresponding check box.

New Tasks can also be created or audited using this window.

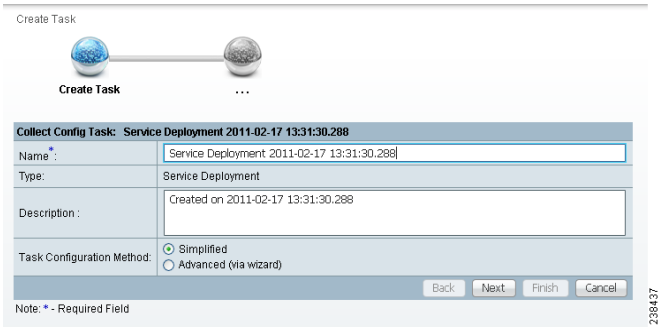
Create

To create a new task, follow these steps:

Step 1 From the Task Manager Window, click **Create**. From the resulting drop-down list, you can choose from the following and that choice becomes the **Type** in Figure 10-12.

- **Collect Config**—Collects configuration from devices.
- **Collect Config From Files**—Collects configurations from files for Prime Diagnostics only.
- **Enable Disable VFW Traps**—Enable or disable the VFW traps.
- **L2VPN (L2TPv3) Functional Audit**—
- **Password Management**—Manages user passwords and SNMP community strings.
- **SLA Collection**—Collects data from SLA enabled devices.
- **Service Deployment**—Deploys an existing SR.
- **TE Full Discovery**—
- **TE Incremental Discovery**—
- **TE Interface Performance**—Calculates tunnel and interface bandwidth utilization using SNMP.

Figure 10-12 Create Tasks



Step 2 **Name**—Enter the name of the task. You can accept the default value.

Step 3 **Type**—Defined in Step 1.

Step 4 **Description** (optional)—Enter a description.

Step 5 **Task Configuration Method** (default: **Simplified**)—Choose **Simplified** or **Advanced (via wizard)**. If you choose **Simplified**, you can make many selections in one window. If you choose **Advanced (via wizard)**, you navigate through many windows to make your selections.

Step 6 Click **Next** to continue.

Depending on what type of task you select, the Task Devices, Task Service Requests, or Configurations File Directory page appears with variations.

Step 7 Where appropriate, click **Select/Deselect** to add devices or service requests.

Note Step 7 to Step 10 do not apply for Collect Config From Files and TE Interface Performance.

Step 8 In the resulting selection window, select the devices or service requests and click **Select**.

The selected devices or service requests appears.

Step 9 **Groups** might or might not appear depending on the task you specify in the previous step. If it does appear, you can add groups of devices, similarly to Step 7 and Step 8. If it does not appear or after you complete this device group selection, proceed to Step 10.

Step 10 Choose the **Options**.

If the **Retrieve Interfaces** check box is checked, Prime Fulfillment uses Simple Network Management Protocol (SNMP) to retrieve device interface information, such as ifIndex, and so on. If the **Retrieve Interfaces** check box is unchecked, configuration collection information is still retrieved, but SNMP is not used. All scenarios other than doing IP Service Level Agreement (SLA) probes do not require SNMP or this option.

Step 11 If **Configuration File Directory** appears, enter the path to the directory on your Prime Fulfillment server into the **Configuration File Directory** text box, to indicate the directory on the Prime Fulfillment server where the offline configuration files are stored.

Step 12 For **Schedule**, click **Now**, **Later**, or **None**. If you choose **Later**, a Later Schedule category appears. You are then required to click the **Edit** button and the Task Scheduler page appears.

Step 13 Select information to schedule the task and click **OK** (default is to schedule **Now**).

Step 14 Click **Submit** to continue.

The new task is added to the list of tasks.

Audit

To get audit information, click **Audit** from the **Tasks** page. From the resulting drop-down list, you can choose from the following and that choice becomes the **Type**:

- **Config Audit**—Compares Prime Fulfillment generated configlet against the one in the device.
- **L2VPN (L2TPv3) Functional Audit**—Audits L2TPv3 functionality.
- **MPLS Functional Audit**—Audits MPLS functionality.
- **TE Functional Audit**—Checks the Label-Switch Path (LSP) on a router against the LSP stored in the repository.

Details

To get details about a particular task, follow these steps:

Step 1 From the **Tasks** page, check a check box for one task for which you want to see a detailed list of information.

Step 2 Click **Details.**

Step 3 Click **OK** to return.

Schedules

To change the scheduling of an existing task, follow these steps:

Step 1 From the **Tasks** page, check a check box for the one task for which you want to reset the scheduling directions.

Step 2 Click **Schedules.**

Step 3 If you want to delete this task, proceed to Step 4. If you want to reset the scheduling directions, proceed to Step 5.

Step 4 In the new window, check the check box for the task you want to delete and click the **Delete** button. Then proceed to Step 7.

Step 5 In the new window, click **Create**.

Step 6 Make the new scheduling selections you want and click **Save** to reset the scheduling directions.

Step 7 Uncheck any check boxes and click **OK** to return.

Logs

This selection from the **Tasks** page, is another way of doing what is explained in the "Task Logs" section on page 10-26.

Delete

To delete one or more tasks, follow these steps:

Step 1 From the **Tasks** page, check one or more check boxes for the task(s) you want to delete.

You receive a confirmation window.

Step 2 If you want to delete, click **OK**. If not, click **Cancel**.

Step 3 You return to an updated **Tasks** page.

Task Logs

Task Logs can be used to understand the status of a task, whether it completed successfully. You can also use the Task Logs to troubleshoot why a task has failed. To view the Task Logs, follow these steps:

Step 1 Click Operate > **Tasks > Task Logs**.

The Task Logs window appears.

This window displays the task by **Runtime Task Name**, and the **Action**, **Start Time**, **End Time**, and the **Status** of the task. You can use this window to view or delete the logs.

Step 2 To view the log, check the check box for the row that represents the task and click the **View Log** button.

The Task Log page appears.

It is possible to set the types of log level you want to view. Specify the Log Level and click on the Filter button to view that information you want to view.

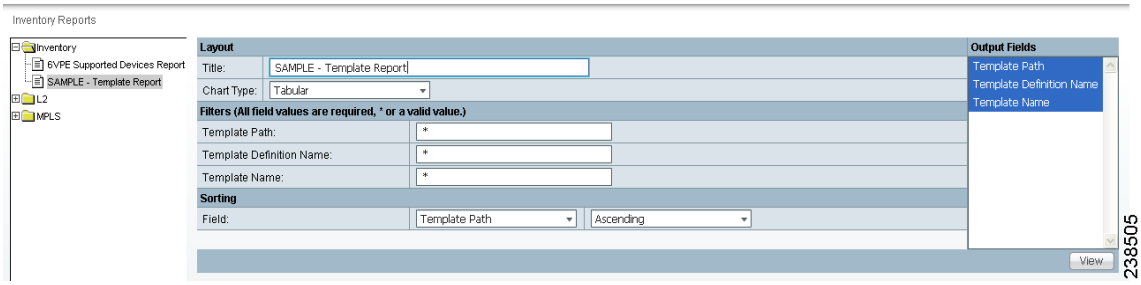
Step 3 Click **Return to Logs** to specify another log to view.

Reports

When you choose **Inventory > Reports > Inventory Reports**, a tree of reports appears in the data pane. Click on the **+** sign for each folder in the data pane and you receive a listing of all the provided reports. The non-SAMPLE reports in the L2VPN folder and the non-SAMPLE reports in the MPLS folder are explained in the *Cisco Prime Fulfillment User Guide 6.2*.

Click on any of the specific reports and you can define how to set up the report. Figure 10-13, shows the sample file under the folder **Inventory**.

Figure 10-13 Inventory > SAMPLE - Template Report - Report Window



This section explains the Reports feature and how to use it in the following areas:

- Introducing Reports, page 10-28
- Accessing Reports, page 10-28
- Using Reports GUI, page 10-28
- Running Reports, page 10-29
- Creating Custom Reports, page 10-31

Introducing Reports

Network operators often want to have detailed reports on the services provisioned. For example, for a given customer, you might want to see a list of the PE-CE connections and their detailed PE-CE configuration parameters or you might want to see specific Layer2 or Layer3 service requests on a PE. These reports help network operators by providing a centralized location for finding Service Requests (SRs) and VPN information.

When you choose **Inventory > Reports > Inventory Reports**, reports are grouped by type to allow for easy navigation. Prime Fulfillment displays only predefined (canned) reports for which the user has RBAC permission.

You can select the filtering criteria and the outputs to be displayed in the report. You can save reports to a variety of formats.

In addition to the predefined reports that are documented in the *Cisco Prime Fulfillment User Guide 6.2*, Prime Fulfillment provides additional sample reports. Sample reports are provided for informational purposes only and are untested and unsupported.

The data structures that Prime Fulfillment uses to provide reports in the GUI are defined in an XML format.

Accessing Reports

To access the reports, follow these steps:

Step 1 To access the reports framework in the Prime Fulfillment GUI, choose **Inventory > Reports > Inventory Reports**.

Step 2 Click on the folders to display the available reports.

The Reports window appears, as shown in Figure 10-13.

Step 3 From the reports listed under one of the folders in the left navigation tree, click on the desired report to bring up the window associated with that report.

Note Several sample reports are provided in the each of the reports folders. These reports begin with the title **SAMPLE-**. These reports are provided for informational purposes only. They are untested and unsupported. You might want to use them, along with the supported reports, as a basis for creating your own custom reports. See the "Creating Custom Reports" section on page 10-31 for information about custom reports.

Using Reports GUI

This section provides some general comments on using the reports GUI. This information applies to all reports. When you invoke a report, you see a window like the one shown in Figure 10-13.

The window is divided into several areas:

- Layout, page 10-29
- Filters, page 10-29

Layout

This area displays the title of the report and allows you to select the chart type. You can enter your own report title by overwriting the Title field.

Note Only tabular output is supported.

Filters

In this pane you can define inputs or search criteria for the reports. Values entered here are compared against corresponding values associated with data objects in the Prime Fulfillment repository. Values must be entered for all fields. An asterisk (*) can be used as a wild-card character for an entire string.

For each filterable field, the GUI displays a label and a text input field. For certain fields, the GUI also displays a Select button that allows you to choose an existing object (for example, customer, Service Type, SR State, and so on). All available output fields are displayed in the window, allowing you to select the fields to include in the report. All output fields are selected by default.

Note Filter values must be in the same format as the values represented within Prime Fulfillment. For example, a Service Request (SR) ID must be a number.

Output Fields

In this pane you can choose output fields to be displayed in the report. You can choose any or all of the output fields by selecting them with the mouse. Use the Shift key to select a continuous range of output values. Or, use the Control key to select random output values.

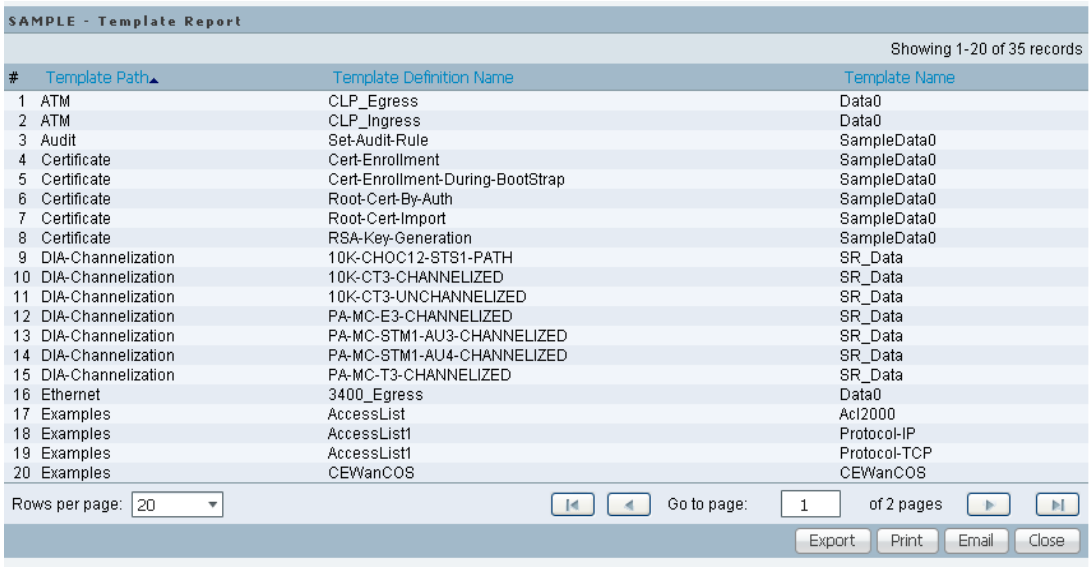
Sorting

This pane allows you to select how you want to sort the report output. For Field:, use the first drop-down list to select each filter field and then the second drop-down list to choose whether to display the report fields in ascending or descending order. The sort order can also be changed after you have the report output displayed (see Figure 10-14).

Running Reports

To run the report, click **View** in the lower right corner of the report window. This generates the report output. An example of a report output is shown in Figure 10-14.

Figure 10-14 Report Output


SAMPLE - Template Report

Showing 1-20 of 35 records

#	Template Path	Template Definition Name	Template Name
1	ATM	CLP_Egress	Data0
2	ATM	CLP_Ingress	Data0
3	Audit	Set-Audit-Rule	SampleData0
4	Certificate	Cert-Enrollment	SampleData0
5	Certificate	Cert-Enrollment-During-BootStrap	SampleData0
6	Certificate	Root-Cert-By-Auth	SampleData0
7	Certificate	Root-Cert-Import	SampleData0
8	Certificate	RSA-Key-Generation	SampleData0
9	DIA-Channelization	10K-CHOC12-STS1-PATH	SR_Data
10	DIA-Channelization	10K-CT3-CHANNELIZED	SR_Data
11	DIA-Channelization	10K-CT3-UNCHANNELIZED	SR_Data
12	DIA-Channelization	PA-MC-E3-CHANNELIZED	SR_Data
13	DIA-Channelization	PA-MC-STM1-AU3-CHANNELIZED	SR_Data
14	DIA-Channelization	PA-MC-STM1-AU4-CHANNELIZED	SR_Data
15	DIA-Channelization	PA-MC-T3-CHANNELIZED	SR_Data
16	Ethernet	3400_Egress	Data0
17	Examples	AccessList	Acl2000
18	Examples	AccessList1	Protocol-IP
19	Examples	AccessList1	Protocol-TCP
20	Examples	CEWanCOS	CEWanCOS

Rows per page: 20 Go to page: 1 of 2 pages

Export Print Email Close

238506

The reports GUI supports output in tabular format. The output is listed in columns, which are derived form the outputs you selected in the reports window.

Each row (or record) represents one match of the search criteria you set using the filter fields in the reports window.

In some cases, the value returned in a field can be displayed as one of the following:

- **-1** means no information updated for this field
- **F** means false
- **T** means true

The column heading with a triangle icon is the output by which the records are sorted. By clicking on any column heading, you can toggle between ascending and descending sort order. To sort on another output value, click on the heading for that value.

From the report output window, you can export, print, or e-mail using the following button:

- Export explained in the "Exporting Reports" section on page 10-30
- Print explained in the "Printing Reports" section on page 10-31
- E-mail explained in the "E-mailing Reports" section on page 10-31

Exporting Reports

Click on the **Export** icon in Figure 10-14 and then follow these steps.

Step 1 Select the appropriate radio button for the format you want:

- **PDF** file—Adobe's portable document format.
- **CSV** file—Comma Separated Values format that allows for the data to be easily exported into a variety of applications.

Step 2 Select the rows you would like to save, then click **OK**.

Prime Fulfillment generates the report in the format you selected.

Note You must have the appropriate application on your system (for example, Acrobat Reader or Excel) to view and save the output.

Printing Reports

Click on the **Print** icon in Figure 10-14.

This window allows you to display the report in a form more appropriate for printing. Select the desired rows, then click **OK**. The results are displayed in your web browser, from which you can print the report.

E-mailing Reports

Click on the **E-mail** icon in Figure 10-14 and then follow these steps.

Step 1 In the To: field (required), specify one or more e-mail addresses to which the report should be sent.

Step 2 In the From: field (optional), enter an e-mail address you want to appear in the message header.

This allows a reply message to be sent to a valid e-mail address.

Step 3 In the CC: field (optional), enter e-mail addresses for recipients you want to receive copies of this report.

Step 4 The subject field shows the title of the report being sent.

You can overwrite this field to rename the report. This is what appears in the Subject field of the e-mail message.

Step 5 Select the radio button for the output format (PDF or CSV) in which you want the report sent.

Step 6 Select the number of rows you want sent.

Step 7 If applicable, in the Message field, write a message to announce the report, then click **Send**.

Creating Custom Reports

The reports listed in the Prime Fulfillment GUI in the each folder are derived from an underlying configuration file. The file is in XML format. You can access the file in the following location:

$PRIMEF_HOME/resources/nbi/reports/PrimeFulfillment/<*folder_name*>**_report.xml**

where <*folder_name*> *is* **Inventory**, **L2**, *or* **MPLS**.

Each of the available reports (including sample reports) is defined by XML content contained within an <objectDef name> start and end tag under **packageDef name = "**<*folder_name*>**"**. The intervening XML content specifies the title of the report, all allowable filter parameters, outputs, and the default sorting behavior. You can modify existing reports or copy them to use as templates for new reports.

To do this, follow these steps:

Step 1 Stop the Prime Fulfillment server using the **./prime.sh stopall** command.

See Appendix C, "WatchDog Commands" for information on starting and stopping Prime Fulfillment.

Step 2 Open the **$PRIMEF_HOME/resources/nbi/reports/PrimeFulfillment/**<*folder_name*>**_report.xml (where:** <*folder_name*> *is* **Inventory**, **L2**, *or* **MPLS**) configuration file using an editing tool of your choice.

Note You should back up the file before making any changes to it.

Step 3 Depending on your needs, either modify an existing report or copy one and use it as the basis for a new one.

Step 4 Save the modified **$PRIMEF_HOME/resources/nbi/reports/PrimeFulfillment/**<*folder_name*>**_report.xml** file.

Step 5 Restart the Prime Fulfillment server using the **./prime.sh startwd** command.

See Appendix C, "WatchDog Commands" for information on starting and stopping Prime Fulfillment.

After restarting Prime Fulfillment, the modifications take effect, based on changes you made to the **$PRIMEF_HOME/resources/nbi/reports/PrimeFulfillment/**<*folder_name*>**_report.xml** file.

Generating L2 and VPLS Reports

The Prime Fulfillment reporting GUI is used across multiple Prime Fulfillment modules, including L2 and VPLS. For a general coverage of using the reports GUI, running reports, using the output from reports, and creating customized reports, see Reports, page 10-27. The rest of this section provides information about the L2 and VPLS reports available in Prime Fulfillment.

This section provides information on generating L2 and VPLS reports. It contains the following sections:

Accessing L2 and VPLS Reports

To access the L2 and VPLS reports, perform the following steps:

Step 1 To access the reports framework in the Prime Fulfillment GUI, choose **Inventory > Reports > Inventory Reports**.

The Reports window appears.

Step 2 Click the L2 folder to display the available L2 and VPLS reports.

Step 3 Click the icon of a report to bring up the window associated with that report.

Details on each of the reports are provided in L2 and VPLS Reports, page 10-33.

L2 and VPLS Reports

This section provides details on the following L2 and VPLS reports:

Note Several sample reports are provided in the L2 reports folder. These reports begin with the title **SAMPLE-**. These reports are provided for informational purposes only. They are untested and unsupported. You might want to use them as a basis for creating your own custom reports. For more information, see Creating Custom L2 and VPLS Reports, page 10-39.

The following information is provided for each report:

- Description or purpose of the report.
- An illustration of the report window.
- List of filter values and descriptions.
- List of output values and descriptions.

L2 End-to-End Wire Report

An L2 end-to-end wire is a point-to-point connection containing two attachment circuits. The L2 EndtoEndWire report displays the services that are running on L2 end-to-end connections. You can use this report to view all the services and respective attachment circuit attributes for each connection.

Click the L2 EndtoEndWire Report icon to bring up the window for this report.

Filter Values:

- **EndToEndWire ID**—End-to-end wire identification number.
- **Customer Name**—Name of the customer.
- **VC ID**—Virtual circuit identification number.
- **SR Job ID**—Service request job identification number.
- **Service Type**—Type of service. Values can be:
  - ATM
  - ATM_NO_CE
  - FRAME_RELAY
  - FRAME_RELAY_NO_CE
  - L2VPN_ERS
  - L2VPN_ERS_NO_CE
  - L2VPN_EWS
  - L2VPN_EWS_NO_CE

- **SR State**—Service request state. Values can be:
  - BROKEN
  - DEPLOYED
  - FAILED_AUDIT
  - FAILED_DEPLOY
  - FUNCTIONAL
  - INVALID
  - LOST
  - PENDING
  - REQUESTED
  - WAIT_DEPLOY
- **AC1-ID**—First attachment circuit (AC1) identification number.
- **AC2-ID**—Second attachment circuit (AC2) identification number.

Output Values:

- **EndToEndWire ID**—End-to-end wire identification number.
- **Customer Name**—Name of the customer.
- **VPN**—Name of the VPN.
- **VC ID**—Virtual circuit identification number.
- **SR ID**—Service request identification number.
- **SR Job ID**—Service request job identification number.
- **Service Type**—Type of service.
- **SR State**—Service request state.

Note The **SR State** output does not list service requests in the **CLOSED** state. Service requests in other states are listed, as determined by the filter values.

- **AC1-ID**—Identification number of the first attachment circuit (AC1).
- **AC1-UNI Device Interface**—UNI device interface of the first attachment circuit (AC1).
- **AC1-NPC**—Named physical circuit for the first attachment circuit (AC1).
- **AC2-VLAN ID/DLCI/VCD**—VLAN identification number, DLCI (data-link connection identifier) or VCD (virtual circuit descriptor) of the first attachment circuit (AC1).
- **AC1-VPI**—Virtual path identifier for the first attachment circuit (AC1).
- **AC1-VCI**—Virtual channel identifier for the first attachment circuit (AC1).
- **AC1-Interface Encap Type**—Encapsulation type used for the first attachment circuit (AC1).
- **AC1-AccessDomain**—Access domain name for the first attachment circuit (AC1).
- **AC1-Customer Facing UNI**—Customer-facing UNI port of the first attachment circuit (AC1).
- **AC1-Loopback IP Address**—Loop back address for the first attachment circuit (AC1).
- **AC1-STP Shutdown Threshold**—Spanning Tree Protocol shutdown threshold (in packets/second) for the first attachment circuit (AC1).

- **AC1-VTP Shutdown Threshold**—VLAN Trunk Protocol shutdown threshold (in packets/second) for the first attachment circuit (AC1).
- **AC1-CDP Shutdown Threshold**—Cisco Discovery Protocol shutdown threshold (in packets/second) for the first attachment circuit (AC1).
- **AC1-STP Drop Threshold**—Spanning Tree Protocol drop threshold (in packets/second) for the first attachment circuit (AC1).
- **AC1-CDP Drop Threshold**—Cisco Discovery Protocol drop threshold (in packets/second) for the first attachment circuit (AC1).
- **AC1-VTP Drop Threshold**—VLAN Trunk Protocol drop threshold (in packets/second) for the first attachment circuit (AC1).
- **AC1-UNI Recovery Interval**—Recovery interval (in seconds) of the UNI port for the first attachment circuit (AC1).
- **AC1-UNI Speed**—UNI port speed for the first attachment circuit (AC1).
- **AC1-UNI Shutdown**—Shutdown status of the UNI port for the first attachment circuit (AC1).
- **AC1-UNI PortSecurity**—Status of UNI port security for the first attachment circuit (AC1).
- **AC1-UNI Duplex**—Duplex status (none, full, half, or auto) of the UNI port for the first attachment circuit (AC1).
- **AC1-Maximum MAC Address**—Maximum MAC addresses allowed on the UNI port for the first attachment circuit (AC1).
- **AC1-UNI Aging**—Length of time, in seconds, that MAC addresses can stay in the UNI port security table for the first attachment circuit (AC1).
- **AC2-ID**—Second attachment circuit (AC2) identification number.
- **AC2-UNI Device Interface**—UNI device interface of the second attachment circuit (AC2).
- **AC2-NPC**—Named physical circuit for the second attachment circuit (AC2).
- **AC2-VLAN ID/DLCI/VCD**—The VLAN ID, DLCI or VCD of the second attachment circuit (AC2).
- **AC2-VPI**—Virtual path identifier for the first attachment circuit (AC2).
- **AC2-VCI**—Virtual channel identifier for the first attachment circuit (AC2).
- **AC2-Interface Encap Type**—Encapsulation type used for the second attachment circuit (AC2).
- **AC2-AccessDomain**—Access domain name for the second attachment circuit (AC2).
- **AC2-Customer Facing UNI**—Customer-facing UNI port of the second attachment circuit (AC2).
- **AC2-Loopback IP Address**—Loop back address for the second attachment circuit (AC2).
- **AC2-STP Shutdown Threshold**—Spanning Tree Protocol shutdown threshold for the second attachment circuit (AC2).
- **AC2-VTP Shutdown Threshold**—VLAN Trunk Protocol shutdown threshold for the second attachment circuit (AC2).
- **AC2-CDP Shutdown Threshold**—Cisco Discovery Protocol shutdown threshold for the second attachment circuit (AC2).
- **AC2-STP Drop Threshold**—Spanning Tree Protocol drop threshold for the second attachment circuit (AC2).
- **AC2-CDP Drop Threshold**—Cisco Discovery Protocol drop threshold for the second attachment circuit.

- **AC2-VTP Drop Threshold**—VLAN Trunk Protocol drop threshold for the second attachment circuit (AC2).
- **AC2-UNI Recovery Interval**—Recovery interval of the UNI port for the second attachment circuit (AC2).
- **AC2-UNI Speed**—UNI port speed for the second attachment circuit (AC2).
- **AC2-UNI Shutdown**—Shutdown status of the UNI port for the second attachment circuit (AC2).
- **AC2-UNI PortSecurity**—Status of UNI port security for the second attachment circuit (AC2).
- **AC2-UNI Duplex**—Duplex status (none, full, half, or auto) of the UNI port for the second attachment circuit (AC2).
- **AC2-Maximum MAC Address**—Maximum MAC addresses allowed on the UNI port for the second attachment circuit (AC2).
- **AC2-UNI Aging**—Length of time, in seconds, that MAC addresses can stay in the UNI port security table for the second attachment circuit (AC2).

L2 PE Service Report

The L2 PE Service report allows you to choose PEs and display their roles (for example, N-PE, U-PE or PE-AGG) and L2-related services that are running on them.

Click the L2 PE Service Report icon to bring up the window for this report.

Filter Values:

- **PE Role**—PE device role (N-PE, U-PE, or PE-AGG).
- **PE Name**—PE device name.

Output Values:

- **PE Role**—PE device role (N-PE, U-PE, or PE-AGG).
- **PE Name**—PE device name.
- **SR ID**—Service request identification number.
- **SR Job ID**—Service request job identification number.
- **SR State**—Service request state.

Note The **SR State** output does not list service requests in the **CLOSED** state. Service requests in other states are listed, as determined by the filter values.

- **Service Type**—Type of service.

L2 VPN Report

The L2 VPN Report provides a way to track a VLAN ID and/or VC ID back to the VPN and customer without having to iterate through every link and every VPN service. Given a VLAN ID or VC ID, the respective customer and VPN details are displayed in the report.

Click the L2 VPN Report icon to bring up the window for this report.

Filter Values:

- **VLAN ID**—VLAN identification number.
- **VC ID**—Virtual circuit identification number.

- **Customer Name**—Name of the customer.
- **Access Domain**—Access domain name.

Output Values:

- **VLAN ID**—VLAN identification number.
- **VC ID**—Virtual circuit identification number.
- **SR Job ID**—Service request job identification number
- **VPN**—Name of the VPN.
- **Customer Name**—Name of the customer.
- **Service Type**—Type of service.
- **Access Domain**—Access domain name.
- **Provider Name**—Name of the provider.

VPLS Attachment Circuit Report

The VPLS Attachment circuit report displays details of attachment circuits for a given customer VPN. Click the VPLS Attachment Circuit Report icon to bring up the window for this report.

Filter Values:

- **SR ID**—Service request identification number.
- **SR Job ID**—Service request job identification number.
- **SR State**—Service request state. Values can be:
  - BROKEN
  - DEPLOYED
  - FAILED_AUDIT
  - FAILED_DEPLOY
  - FUNCTIONAL
  - INVALID
  - LOST
  - PENDING
  - REQUESTED
  - WAIT_DEPLOY
- **Customer Name**—Name of the customer.
- **VPN**—Name of the VPN.
- **Service Type**—Type of service. Values can be:
  - VPLS_ERS
  - VPLS_ERS_NO_CE
  - VPLS_EWS
  - VPLS_EWS_NO_CE
- **VLAN ID**—VLAN identification number.
- **AccessDomain**—Access domain name.

Output Values:

- **VPLS Link ID**—VPLS link identification number.
- **SR ID**—Service request identification number
- **SR Job ID**—Service request job identification number.
- **SR State**—Service request state.

Note The **SR State** output does not list service requests in the **CLOSED** state. Service requests in other states are listed, as determined by the filter values.

- **Customer Name**—Name of the customer.
- **VPN**—Name of the VPN.
- **Service Type**—Type of service.
- **VLAN ID**—VLAN identification number.
- **Policy Name**—Name of the VPLS policy.
- **VFI Interface**—Virtual forwarding interface name.
- **Customer Facing UNI**—Customer-facing UNI port.
- **AccessDomain**—Access domain name.
- **NPC**—Named physical circuit.
- **UNI Port**—UNI port.
- **UNI Shutdown**—Shutdown status of the UNI port.
- **UNI Aging**—Length of time, in seconds, that MAC addresses can stay in the UNI port security table.
- **UNI Speed**—UNI port speed.
- **UNI Duplex**—Duplex status (none, full, half, or auto) of the UNI port.
- **Maximum MAC Address**—Maximum MAC addresses allowed on the UNI port.
- **CDP Shutdown Threshold**—Cisco Discovery Protocol shutdown threshold (in packets/second) on the UNI port.
- **STP Shutdown Threshold**—Spanning Tree Protocol shutdown threshold (in packets/second) on the UNI port.
- **VTP Shutdown Threshold**—VLAN Trunk Protocol shutdown threshold (in packets/second) on the UNI port.
- **CDP Drop Threshold**—Cisco Discovery Protocol drop threshold (in packets/second) on the UNI port.
- **VTP Drop Threshold**—VLAN Trunk Protocol drop threshold (in packets/second) on the UNI port.
- **STP Drop Threshold**—Spanning Tree Protocol drop threshold (in packets/second) on the UNI port.
- **Recovery Interval**—Recovery interval (in seconds) of the UNI port.

VPLS PE Service Report

The VPLS PE Service report allows you to choose PEs and display their roles (for example, N-PE, U-PE or PE-AGG) and the VPLS services that are running on them.

Click the VPLS PE Service Report icon to bring up the window for this report.

Filter Values:

- **PE Role**—PE device role (N-PE, U-PE, or PE-AGG).
- **PE Name**—PE device name.

Output Values:

- **PE Role**—PE device role (N-PE, U-PE, or PE-AGG).
- **PE Name**—PE device name.
- **SR ID**—Service request identification number.
- **SR Job ID**—Service request job identification number.
- **Service Type**—Type of service.
- **SR State**—Service request state.

Note The **SR State** output does not list service requests in the **CLOSED** state. Service requests in other states are listed, as determined by the filter values.

VPLS VPN Report

The VPLS VPN report provides a way to track a VLAN ID and/or VFI Name back to the VPN and customer without having to iterate through every link and every VPN service. Given a VLAN ID or VFI name, the respective customer and VPN details are displayed in the report.

Click the VPLS VPN Report icon to bring up the window for this report.

Filter Values:

- **VLAN ID**—VLAN identification number.
- **Customer Name**—Name of the customer.
- **VFI Name**—Virtual forwarding interface name.
- **Access Domain**—Access domain name.

Output Values:

- **VLAN ID**—VLAN identification number.
- **SR Job ID**—Service request job identification number.
- **VPN**—Name of the VPN.
- **Customer Name**—Name of the customer.
- **Service Type**—Type of service.
- **VFI Name**—Virtual forwarding interface name.
- **Access Domain**—Access domain name.
- **Provider Name**—Name of the provider.

Creating Custom L2 and VPLS Reports

The reports listed in the Prime Fulfillment GUI in the L2 folder are derived from an underlying configuration file. The file is in XML format. You can access the file in the following location:

$ISC_HOME/**resources/nbi/reports/ISC/l2_report.xml**

See Reports, page 10-27 for details on how to modify report configuration files to create custom reports.

Generating MPLS Reports

The Prime Fulfillment reporting GUI is used across multiple Prime Fulfillment modules, including MPLS. The rest of this chapter provides information about the MPLS reports available in ISC.

This section provides information on generating MPLS reports. It contains the following sections:

Accessing MPLS Reports

To access MPLS reports, perform the following steps:

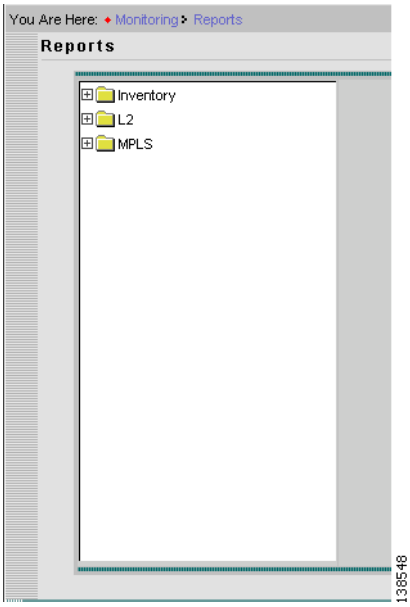
Step 1 Log into Prime Fulfillment.

Step 2 Go to: **Monitoring > Reports**.

Step 3 Click on the MPLS folder to display the available MPLS reports.

The Reports window appears, as shown in Figure 10-15.

Figure 10-15 Reports List



Step 4 From the reports listed under MPLS in the left navigation tree, click on the desired report to bring up the window associated with that report.

Note Several sample reports are provided in the MPLS reports folder. These reports begin with the title **SAMPLE-**. These reports are provided for informational purposes only. They are untested and unsupported. You might want to use them, along with the supported reports, as a basis for creating your own custom reports. See Creating Custom Reports, page 10-45, for information on custom reports.

Running Reports

To run the report, click **View** in the lower right corner of the report window. This generates the report output. An example of an MPLS service request report output.

In the current release of ISC, the reports GUI supports output in tabular format. The output is listed in columns, which are derived form the outputs you selected in the reports window.

Each row (or record) represents one match of the search criteria you set using the filter fields in the reports window.

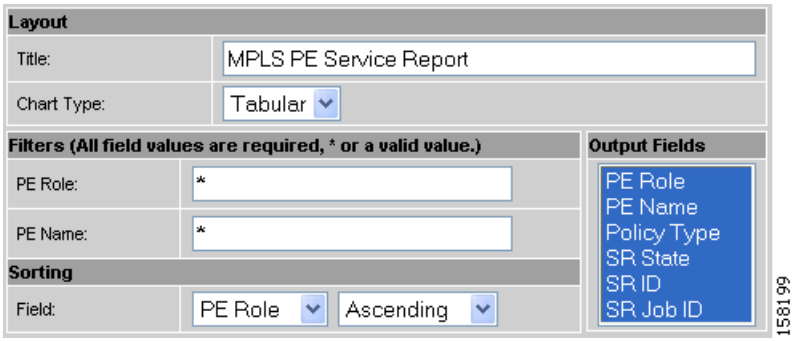
The column heading with a triangle icon is the output that the records are sorted by. By clicking on any column heading, you can toggle between and ascending and descend sort order. To sort on another output value, click on the heading for that value.

MPLS PE Service Report

The MPLS PE Service report allows you to choose PEs and display their roles (for example, N-PE, U-PE or PE-AGG) and MPLS-related services that are running on them.

Click the MPLS Service Report icon to bring up the window for this report, as shown in Figure 10-16.

Figure 10-16 MPLS PE Service Report



Filter Values

- **PE Role**—PE device role (N-PE, U-PE, or PE-AGG).
- **PE Name**—PE device name.

Output Values

- **PE Role**—List by PE device role (N-PE, U-PE, or PE-AGG).
- **PE Name**—List by PE device name.
- **Policy Type**—List by type of Policy.
- **SR State**—List by service request state (see Service Request States, page 8-13).

Note The **SR State** output does not list service requests in the **CLOSED** state. Service requests in other states are listed, as determined by the filter values.

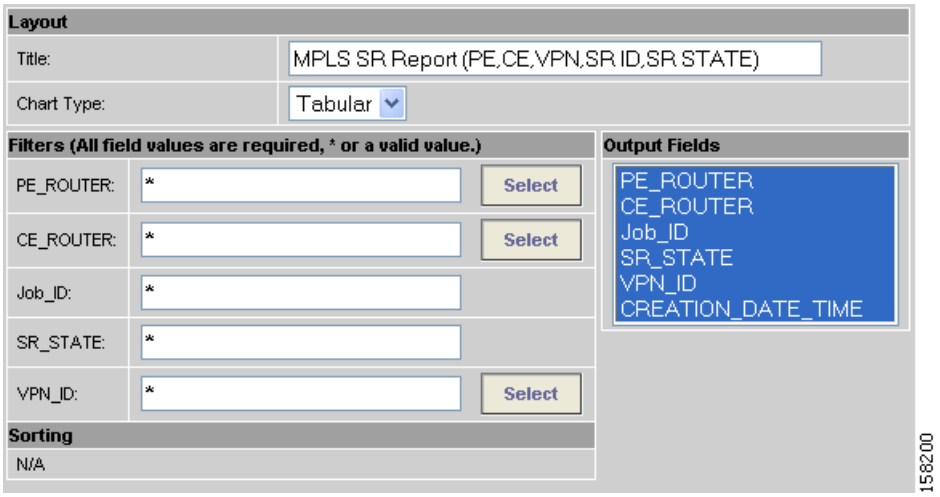
- **SR ID**—List by service request ID.
- **SR Job ID**—List by service request job ID.

MPLS Service Request Report

The MPLS service request report feature allows you to list service requests as related to PE, CE, VPN, SR ID, SR STATE.

Click the MPLS Service Request Report icon to bring up the window for this report, as shown in Figure 10-17.

Figure 10-17 MPLS Service Request Report



Filter Values

- **PE ROUTER**—Choose some or all (*) PE routers.
- **CE ROUTER**—Choose some or all (*) CE routers.
- **Job ID**—Service request job IDs.
- **SR STATE**—Service request states (see Service Request States, page 8-13).
- **VPN ID**—Choose some or all (*) VPNs by ID.

Output Filters

- **PE ROUTER**—Show PE routers.
- **CE ROUTER**—Show CE routers.
- **Job ID**—List by Job ID.
- **SR STATE**—Service request states (see Service Request States, page 8-13).

Note The **SR State** output does not list service requests in the **CLOSED** state. Service requests in other states are listed, as determined by the filter values.

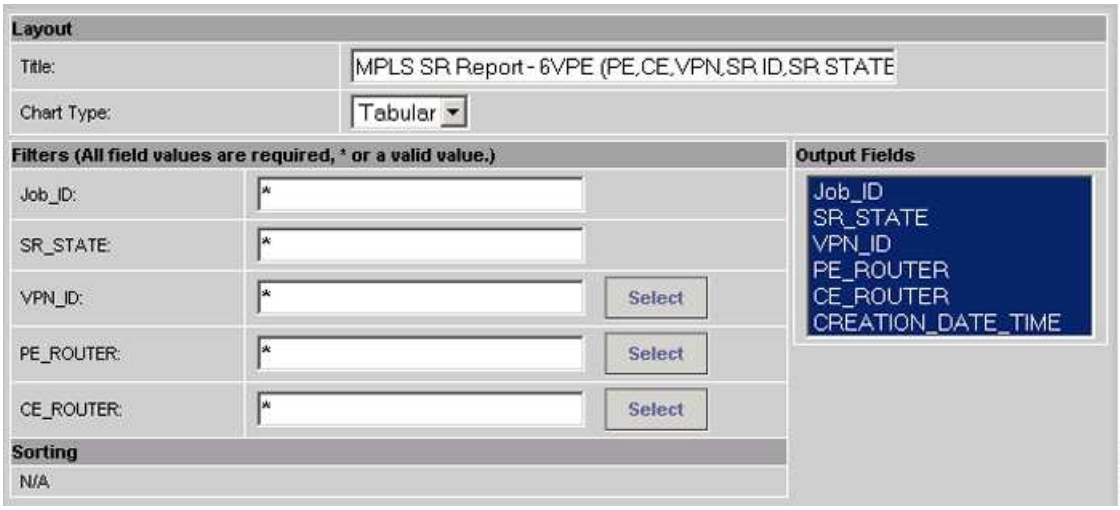
- **VPN ID**—List by VPN ID.
- **CREATION DATE TIME**—List by date and time report created.

MPLS Service Request Report - 6VPE

The MPLS Service Request - 6VPE report feature allows you to list service requests as related to PE, CE, VPN, SR ID, SR STATE.

Click the MPLS Service Request Report - 6VPE icon to bring up the window for this report, as shown in Figure 10-17.

Figure 10-18 MPLS Service Request Report - 6VPE



Filter Values

- **Job ID**—Service request job IDs.
- **SR STATE**—Service request states (see Service Request States, page 8-13).
- **VPN ID**—Choose some or all (*) VPNs by ID.
- **PE ROUTER**—Choose some or all (*) PE routers.
- **CE ROUTER**—Choose some or all (*) CE routers.

Output Filters

- **Job ID**—List by Job ID.
- **SR STATE**—Service request states (see Service Request States, page 8-13).

Note The **SR State** output does not list service requests in the **CLOSED** state. Service requests in other states are listed, as determined by the filter values.

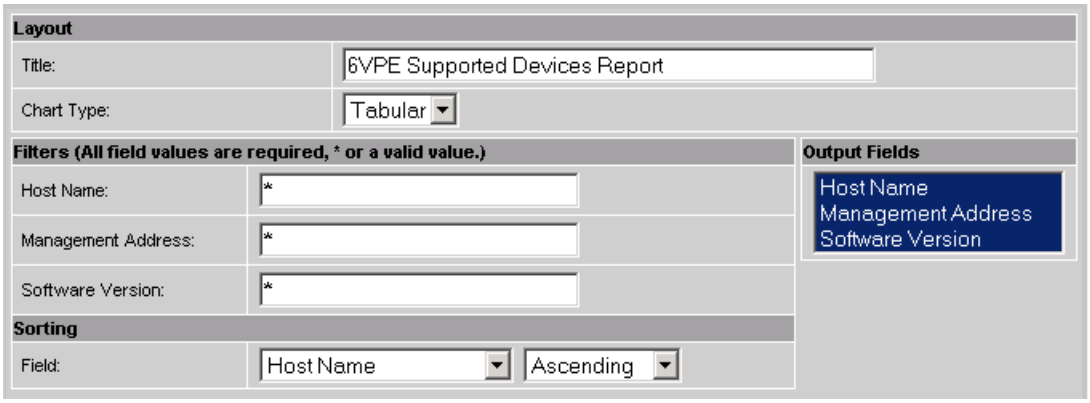
- **VPN ID**—List by VPN ID.
- **PE ROUTER**—Show PE routers.
- **CE ROUTER**—Show CE routers.
- **CREATION DATE TIME**—List by date and time report created.

6VPE Supported Devices Report

Note In the Prime Fulfillment GUI, this report is located under **Monitoring > Reports > Inventory**.

Click the 6VPE Supported Devices Report icon to bring up the window for this report, as shown in Figure 10-17.

Figure 10-19 6VPE Supported Devices Report



Filter Values

- **Host Name**—Hostname.
- **Management Address**—Management address.
- **Software Version**—Software version.

Output Filters

- **Host Name**—Hostname.
- **Management Address**—Management address.
- **Software Version**—Software version.

Creating Custom Reports

The reports listed in the Prime Fulfillment GUI in the MPLS folder are derived from an underlying configuration file. The file is in XML format. You can access the file in the following location:

$ISC_HOME/resources/nbi/reports/ISC/mpls_report.xml

Generating TEM Reports and Logs

All deployment and collection tasks are monitored and the details of the tasks are logged. The information can be viewed using the task monitoring pages.

This chapter includes the following sections:

TE Task Logs

The TE task logs are used to view the result of running one or more TE tasks. Different task logs are generated by different events:

- SR deployment logs
- Logs generated by tasks issued from the Task Manager, such as:
  - TE Discovery
  - TE Functional Audit
  - TE Interface Performance.

SR Deployment Logs

When any service request is deployed, whether a managed or unmanaged primary tunnel or a backup tunnel, a log is generated. For tunnel SRs, deployment takes place in multiple phases depending on the type of SR and the task logs are created similarly:

- Primary tunnel SR—a three-phase logging process corresponding to a three-phase deployment
- Protection SR—a two-phase logging process corresponding to a two-phase deployment

In addition to the deployment logs, a ConfigAudit log is created regardless of the type of SR deployment, providing the deployment was successful.

Logs Created from Task Manager

Specific instructions for how to generate and view a task log for a TE Discovery task are found in Task Logs, page 10-26.

Instructions for how to generate and view a task log for the TE Functional Audit and TE Interface Performance tasks are found in Creating a TE Task, page 7-74.

Viewing a Task Log

A task log can be accessed from two different locations:

- The Tasks window
- The Service Requests window.

From the Tasks Window

To view the task log for a TE task, you need to:

1. Access the Task Logs window.
2. Select the desired log and open it.

To view the task logs, use the following steps. A task log from the deployment of a managed primary tunnel has been used as an example.

Step 1 Choose **Operate > Task Logs**.

The Task Logs window appears.

The Task Logs window includes the following:

- **Runtime Task Name**—Automatically attributed task name specifying when the runtime task was created.

- **Action**—Type of task, for example **TE Discovery**, **TE Functional Audit**, or **TE Interface Performance**.
- **Start Time**—The date and time when the runtime task was started.
- **End Time** —The date and time when the runtime task ended.
- **Status**—Indicates the present status of the runtime task.

Step 2 Select a Task Log for viewing.

A task that has been scheduled for multiple runs might have multiple instances to view.

Step 3 Click the desired task in the **Action** column.

The corresponding Task Log window appears. The GUI elements in this window are also found in the Service Request Manager window.

The logged messages are shown in a table. This includes the time the log message was created and the severity level assigned to the log message.

There is a filter setting for the logging, which defaults to SEVERE. This means that only SEVERE messages in the log are shown. There are several different filter settings that can be selected according to the desired level of detail. To change the filter level, select the one that is required and click **Filter**.

How the log is structured depends on the type of task that was run.

Step 4 Click **Return to Logs** to close the log window.

This takes you back to the main Task Logs window.

Step 5 To see the task SR, which in some cases is associated with a particular task log, select the desired task log and click the **Service Requests** button.

The Task SRs window appears.

From the Service Requests Window

To access the logs from the Service Requests window:

Step 1 Choose **Operate > Service Request Manager**.

Step 2 Select a service request (only one).

Step 3 Click the **Status** button and select **Logs.**

Step 4 Select the log to view and click **View Log**.

The Task Log window appears.

Step 5 Select the log level from the drop-down menu and click **Filter**.

The log levels are All, Severe, Warning, Info, Config, Fine, Finer, and Finest.

TE Performance Reports

A TE Performance Report is created when you run a TE Interface Performance task as described in Creating a TE Interface Performance Task, page 7-75.

It shows the traffic data collected from the TE Interface Performance task for selected tunnels and/or links. The TE Interface Performance task can run multiple times.

To view a TE Performance Report, use the following steps:

Step 1 Choose **Monitoring > TE Performance Report**.

The TE Performance Report Table appears.

The TE Performance Report Table window includes the following GUI elements:

- Report table—The table shows a list of Interface Performance tasks:
  - **Start Time**—The date and time when the runtime task was started.
  - **End Time**—The date and time when the runtime task ended.
  - **Device Name**—Name of the device.
  - **Interface Name**—IP addresses of the interfaces on the link.
  - **Octets In**—Number of inbound octets of traffic.
  - **Octets Out**—Number of outbound octets of traffic.
  - **Speed**—Speed of the interface.
  - **Util In**—Interface utilization for inbound traffic.
  - **Util Out**—Interface utilization for outbound traffic.
- **Reconcile Data**—When an Interface Performance task has been run multiple times on an interface, you can choose to reconcile the data according to the following criteria:
  - **Peak**—Select the highest interface utilization.
  - **Valley**—Select the lowest interface utilization.
  - **Average**—Select the average interface utilization.
  - **First**—Select the first occurrence of interface utilization.

CHAPTER 11

Performing Diagnostics

This chapter describes the Diagnostics application in Cisco Prime Fulfillment 6.2.

Introduction

This section provides an overview of the Cisco Prime Fulfillment Diagnostics application.

The chapter contains the following sections:

Diagnostics Overview

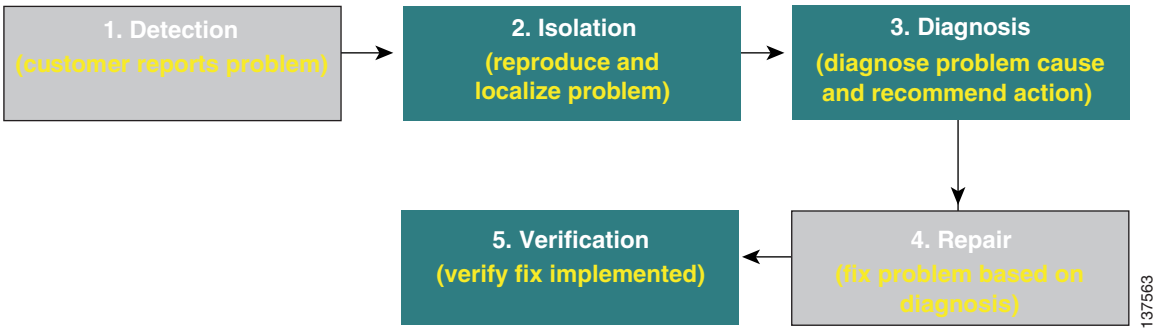
Diagnostics is an automated, workflow-based network management application that troubleshoots and diagnoses problems in Multiprotocol Label Switching (MPLS) VPNs. Diagnostics offers users the capability to reduce the amount of time required to diagnose MPLS-related network outages—in many cases from hours to minutes. It performs diagnostics based on analysis of network failure scenarios, across MPLS access, edge, and core networks. It is equally applicable to both service provider and enterprise “self-deployed” MPLS VPN networks. Network operations center (NOC) support technicians as well as second-line and third-line support can benefit from this product. Diagnostics optionally integrates with the Prime Fulfillment MPLS VPN provisioning component. To diagnose MPLS VPN core problems, Cisco IOS and IOS XR software releases supporting MPLS operations and maintenance (OAM) features including label-switched path (LSP) ping and LSP traceroute are required.

In effective fault finding and troubleshooting, there are five steps:

1. Detection
2. Isolation
3. Diagnosis
4. Repair
5. Verification

Diagnostics is designed to support reactive situations in which an end customer reports a problem with their VPN service. This is essentially the Detection step in Figure 11-1. The Repair function is not supported because many providers prefer to be in complete control of any changes made to router devices and might have specific in-house procedures for doing so.

Figure 11-1 The Reactive Fault Lifecycle



Note Steps 2, 3, and 5 are performed by Diagnostics. Steps 1 and 4 must be performed manually.

Diagnostics focuses on the Isolation, Diagnosis, and Verification steps. It provides invaluable functionality for isolating and diagnosing failures in the network, determining the device(s) at fault, and checking appropriate device status and configuration to determine the likely reason for the failure. Diagnostics also provides the ability to rerun tests to verify that changes made to the device configuration have resolved the issue.

The functionality can be used on its own, without any dependency on any other modules in Prime Fulfillment (for example, VPN provisioning or Traffic Engineering Management). It can also be used in Prime Fulfillment installations where some or all of the other Prime Fulfillment modules are used. If the MPLS VPN Provisioning functionality is used, then Customer and VPN data can be used as a starting point for troubleshooting, to locate the endpoints (for example, Customer Edge devices) between which connectivity is tested.

In addition to troubleshooting, Diagnostics can also be used for VPN post-provisioning checks. After deploying a VPN, either manually or using Prime Fulfillment VPN provisioning, a connectivity test can be run to verify that the VPN has been provisioned successfully.

Note Diagnostics does not have any support for underlying configuration or routing changes during troubleshooting. During the execution of Diagnostics, any changes made either by the operator or through the control plane of the routers, will not be reflected in the actual troubleshooting performed. Diagnostics does not guarantee that the correct Failure Scenario or observation will be found in cases where such changes are made.

Prerequisite Knowledge

Diagnostics has been designed for use by users who have minimal MPLS VPN knowledge. A Diagnostics MPLS VPN Connectivity Verification Test can be performed by a user with little or no MPLS VPN knowledge, and, where necessary, the test results can be exported for interpretation by an engineer familiar with MPLS VPNs. However, due to the complex nature of MPLS VPNs, it is

recommended that you will gain maximum advantage from Diagnostics if you are familiar with MPLS VPNs, in accordance with RFC 2547. In particular, knowledge of RFC 2547 architecture, topology, control, and data planes is helpful to understand how to best use the application and interpret the results.

Diagnostics now diagnoses Cisco devices and networks that use IETF RFC 4379 compliant Label Switched Path (LSP) ping and LSP traceroute. Diagnostics continues to support the earlier draft (draft 3) available in Cisco IOS. You must use a consistent draft of LSP ping and traceroute across all devices in your network.

Recommended reading:

- MPLS and VPN Architectures: Ivan Pepelnjak, Jim Guichard, Cisco Press
- Troubleshooting Virtual Private Networks: Mark Lewis, Cisco Press
- LSP ping/trace RFC: http://www.ietf.org/rfc/rfc4379.txt
- RFC 2547: http://www.ietf.org/rfc/rfc2547.txt?number=2547
- RFC 4379: http://www.ietf.org/rfc/rfc4379.txt?number=4379
- MPLS Embedded Management—LSP Ping/Traceroute and AToM VCCV: http://www.cisco.com/en/US/docs/ios/12_0s/feature/guide/gslsppt.html

Supported Hardware, IOS, and IOS XR Versions

For details of Provider (P) and Provider Edge (PE) network device types and related Cisco IOS and IOS XR versions supported, see the *Cisco Prime Fulfillment Installation Guide 6.2.*

Note Support for additional device types, IOS, and IOS XR versions could be added in patch releases. For details of the latest patch releases and the supported device types, IOS, and IOS XR versions, see Cisco.com.

Device types, IOS, and IOS XR versions detailed in Setting Up the Prime Fulfillment Services, page 3-5 support the MPLS label switched path (LSP) Ping and Traceroute feature. This feature is required for Diagnostics troubleshooting. If all P and PE devices comply with the list of supported device types, IOS, and IOS XR versions, Diagnostics can troubleshoot access circuit, MPLS VPN, and MPLS core problems. Diagnostics is tolerant to other device types, IOS, and IOS XR versions, including other vendors' equipment. However, when the network includes P or PE devices that do not comply with this list, a complete diagnosis might not be possible. Table 11-1 shows the possible scenarios and likely outcome.

Table 11-1 Hardware, IOS, and IOS XR Version Compliance

Scenario	Outcome
All P and PE devices comply with the supported Cisco hardware, IOS, and IOS XR versions.	MPLS VPN Connectivity Verification test successfully troubleshoots access circuit, MPLS VPN, and MPLS core issues.
All PE devices comply with the supported Cisco hardware, IOS, and IOS XR versions. One or more P device(s) do not comply with the supported Cisco hardware, IOS, and IOS XR versions, including other vendors' equipment.	MPLS VPN Connectivity Verification test successfully troubleshoots access circuit and MPLS VPN issues, but might be unable to complete troubleshooting of MPLS core issues.

Table 11-1 Hardware, IOS, and IOS XR Version Compliance (continued)

Scenario	Outcome
PE devices are Cisco hardware running unsupported IOS and IOS XR versions that do not support the MPLS LSP Ping and Traceroute feature.	MPLS VPN Connectivity Verification test *may* be able to successfully troubleshoot access circuit and MPLS VPN issues. The MPLS VPN Connectivity Verification test is unable to perform troubleshooting of the MPLS core.
PE devices are non-Cisco hardware.	MPLS VPN Connectivity Verification test cannot be run.

Diagnostics supports both managed and unmanaged CE routers from any vendor. There are no device type, IOS, or IOS XR version requirements for CE devices.

Diagnostics can work with other device types, IOS, and IOS XR versions that support the MPLS LSP Ping and Traceroute feature. Use the Cisco Feature Navigator for details of device types, IOS, and IOS XR versions that support this feature. See http://tools.cisco.com/ITDIT/CFN/jsp/index.jsp

Note If the PE devices are running unsupported IOS or IOS XR versions, that do not implement the MPLS Ping and Traceroute features, access circuit and VPN edge troubleshooting is performed, but no troubleshooting of the MPLS core is possible. In this scenario some core failures are reported as a Label Forwarding Information Base (LFIB) mismatch on a PE device. The LFIB mismatch is a symptom of the core failure, but the actual core failure cannot be diagnosed because core troubleshooting is not possible.

IPv6

The IPv4 address free pool held by the Internet Assigned Numbers Authority (IANA) is running out. Cisco is addressing this shortage by adopting IPv6 addressing.

Diagnostics supports configuration and selection of devices with both IPv4 and IPv6 addresses. Diagnostics can troubleshoot MPLS VPN services where the attachment circuits:

- use IPv6 addressing
- use dual stack IPv4/IPv6 addressing.

Dual stack is a technique that allows both IPv4 and IPv6 to coexist on the same interfaces. For many years, if not forever, there will be a mix of IPv6 and IPv4 nodes on the Internet. Thus compatibility with the large installed base of IPv4 nodes is crucial for the success of the transition from IPv4 to IPv6. For example, a single interface can be configured with an IPv4 address and an IPv6 address. All the elements referenced as dual-stacked, such as provider edge and customer edge routers, run IPv4 as well as IPv6 addressing and routing protocols.

Note Diagnostics supports only global unicast IPv6 addresses. A global unicast address is very similar in function to an IPv4 unicast address such as 131.107.1.100. In other words, these addresses are conventional and publicly routable addresses. A global unicast address includes a global routing prefix, a subnet ID, and an interface ID.

Table 11-2 General Unicast Address Structure

Fields	Network prefix	Subnet	Interface Identifier
Bits	48	16	64

Note Diagnostics permits to launch a test where both attachment circuit endpoints are either IPv6 and IPv6 or IPv4 and IPv4. No mixed addressing formats can be specified

For more details about when a test is initiated on an IPv6 address, see Understanding the Diagnostics Connectivity Tests, page 11-14.

Diagnostics Features

Diagnostics troubleshooting and diagnostics supports the following four domains:

- Access Circuit—Access circuit troubleshooting includes basic routing protocol troubleshooting, basic layer 1 and layer 3 troubleshooting, and advanced layer 2 troubleshooting for ATM, Frame Relay, and Ethernet.
- MPLS VPN—MPLS VPN troubleshooting supports MPLS/MP-BGP VPNs based on RFC2547. The following topologies are supported: hub and spoke, central services, full mesh, and intranet or extranet VPN.
- MPLS Core—MPLS core troubleshooting supports data plane and control plane diagnostics. This is provided for all MPLS core and edge devices (including troubleshooting of any discovered MPLS Traffic Engineered Tunnels) running a Cisco IOS or Cisco IOS XR version with MPLS Operation, Administration, and Maintenance (OAM) support. For details of Cisco IOS, and Cisco IOS XR versions with MPLS OAM support, see the "Supported Hardware, IOS, and IOS XR Versions" section on page 11-3.

Note Diagnostics does not troubleshoot routing protocols within the core (except OSPF failures on first hop and PE-P-PE topology if the IGP protocol is OSPF), IP connectivity within the core, and some variants of inter-Autonomous Systems (AS) or Carrier-Supporting-Carrier (CsC), specifically Inter AS option B and CsC where there is no LSP.

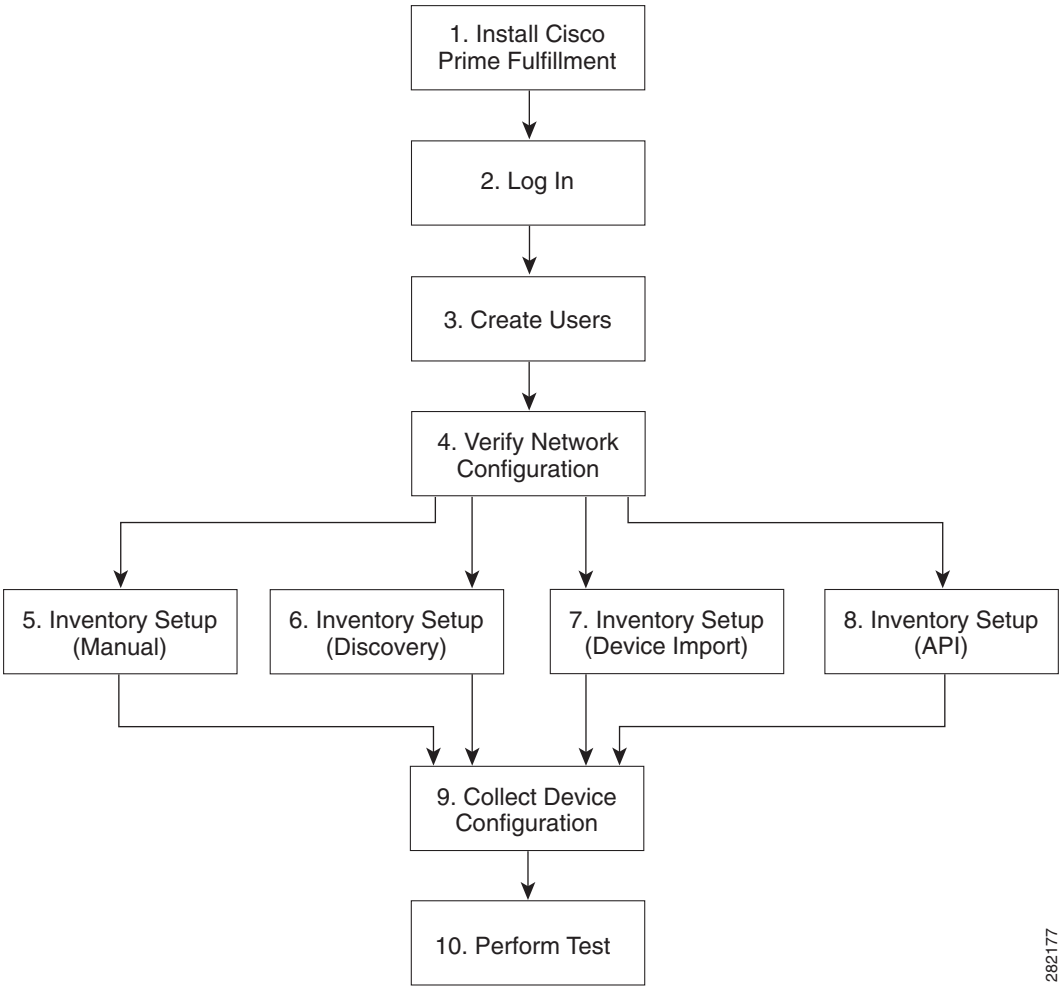
Getting Started

This section describes how to get started using Cisco Prime Fulfillment Diagnostics.

It contains the following sections:

Figure 11-2 describes the getting started workflow for Diagnostics.

Figure 11-2 Getting Started with Diagnostics



6. Create Users—Create users and assign Diagnostics user roles. See User Roles, page 11-7, and Creating Users, page 11-7.
7. Verify Network Configuration—Verify that all network devices have the configuration required for Diagnostics. See Network Configuration, page 11-7.
8. Inventory Setup (Manual)—Manually create required Prime Fulfillment inventory objects. See Inventory Setup, page 11-8.
9. Inventory Setup (Discovery)—Create required Prime Fulfillment inventory objects using Prime Fulfillment Discovery. See Inventory Setup, page 11-8.
10. Inventory Setup (Device Import)—Create required Prime Fulfillment inventory objects using Inventory Manager Import Devices feature. See Inventory Setup, page 11-8.
11. Inventory Setup (API)—Create required inventory objects through Prime Fulfillment APIs. See Inventory Setup, page 11-8.
12. Collect Device Configuration—Collect device configuration, including interface configuration, and add to Prime Fulfillment inventory. A scheduled task can be set up to periodically synchronize Prime Fulfillment inventory with actual device configuration. See Device Configuration Collection, page 11-11.

13. Perform Test—Select, configure, and run an MPLS VPN Connectivity Verification test. See Performing an MPLS VPN Connectivity Verification Test, page 11-18.

User Roles

The functionality available to you as an Prime Fulfillment user is determined by your assigned user roles. User roles also allow you to create and delete devices, collect device configuration, and to perform an MPLS VPN Connectivity Verification test.

To use the Diagnostics functionality, you must be assigned one or more of the following predefined Diagnostics roles depending on the type of connectivity tests you are entitled to perform:

1. MplsDiagnosticsRole—You can perform an MPLS VPN connectivity test between two CEs.
2. MplsDiagnosticsPeToAttachedCeTestRole—You can perform an MPLS VPN connectivity test between a PE and an attached CE.
3. MplsDiagnosticsCetoPeAcrossCoreTestRole—You can perform an MPLS VPN connectivity test between a CE and a PE across the MPLS core.
4. MplsDiagnosticsPetoPeInVrfTestRole—You can perform an MPLS VPN connectivity test between two PEs.
5. MplsDiagnosticsPeToPeCoreTestRole—You can perform a core MPLS connectivity test between two PEs.

Note All Diagnostics roles allow you to create and delete devices, collect device configuration, and to perform an MPLS VPN Connectivity Verification test.

Creating Users

For details on how to create Prime Fulfillment users, see Manage Security, page 14-9.

Network Configuration

This section describes the network configuration required to allow Diagnostics to troubleshoot your network.

MPLS IP Time To Live Propagation

MPLS IP Time To Live (TTL) propagation is enabled by default on Cisco devices. Diagnostics requires that MPLS IP TTL propagation is enabled within the MPLS core. If MPLS IP TTL propagation is not enabled, then Diagnostics is unable to troubleshoot problems within the MPLS core. Troubleshooting of problems in the access circuit, or on the edge of the MPLS core is still possible.

In Cisco IOS, it is possible to disable MPLS IP TTL propagation for packets forwarded to the MPLS core by using the **no mpls ttl-propagate forward** IOS command. This command stops TTL propagation for packets forwarded in to the MPLS core, but allows TTL propagation for packets sent from within the MPLS core. Diagnostics functions correctly in this situation.

When TTL propagation is disabled using the Cisco IOS command **no mpls ip propagate-ttl**, or the Cisco IOS XR command **mpls ip-ttl-propagate disable**, then all TTL propagation is disabled and Diagnostics is unable to troubleshoot your MPLS network.

Note Timestamp must be disabled for the devices, that are selected for troubleshooting and as well as for the devices that are part of the same network.

MPLS LSP Ping/Trace Route Revision

Diagnostics supports IOS MPLS LSP Ping/Traceroute implementations based on version 3 of the IETF LSP Ping draft (draft-ietf-mpls-lsp-ping-03.txt). Later versions of the IETF LSP Ping draft are not supported. Recent IOS versions (including 12.4(6)T), and IOS XR implement later versions of the IETF LSP Ping draft / RFC 4379. To use Diagnostics with these IOS or IOS XR versions you must configure IOS or IOS XR to use version 3 of the IETF LSP Ping draft. To do so you should enter the **mpls oam** command followed by the **echo revision 3** command in IOS or IOS XR global configuration mode. You should ensure that all routers in your core are using the same version of the IEFT LSP ping draft or RFC as appropriate.

31-Bit Prefixes on Point-to-Point Access Circuit Links

For Access circuit links that use IPv4 addressing, Diagnostics supports troubleshooting over access circuit links configured with a 31-bit prefix. However, for each classful network, Diagnostics does not support troubleshooting over two possible 31-bit prefix configurations. These are the subnets that use the classful network address or network broadcast address as a host address. For example, in the class A network, 10.0.0.0, the 31-bit prefix subnet that uses the IP addresses 10.0.0.0 and 10.0.0.1 as host addresses, and the subnet that uses the IP addresses 10.255.255.254 and 10.255.255.255 as host addresses, are not supported. All subnets between these ranges are supported.

If a Diagnostics test is configured using an unsupported 31-prefix subnet, then the test is not run and a message is displayed informing you of the unsupported 31-bit prefix configuration. In this situation, you must manually troubleshoot this link or reconfigure the link to use a supported subnet configuration.

Inventory Setup

Diagnostics can be used without any dependency on other Prime Fulfillment modules. However, before it can be used, the Prime Fulfillment repository must be populated with a number of objects. As a minimum this includes Provider, Provider Region, Device, and PE Device objects. The role of each of these objects is explained below:

- Provider—A Provider is typically a service provider or large corporation that provides network services to a customer. A Provider is a logical inventory object that represents a particular provider.
- Provider Region—A Provider Region is considered to be a group of provider edge routers (PEs) within a single Border Gateway Protocol (BGP) autonomous system. The primary objective for defining Provider Regions is to allow a provider to employ unique IP address pools in large Regions, such as Europe, Asia Pacific, and so forth.
- Device—A Device in Prime Fulfillment is a logical representation of a physical device in the network. Every network element that Prime Fulfillment manages must be defined as a device in the system.

- PE Device—A PE Device is a logical representation of a Provider Edge (PE) or Provider (P) router that has been associated with a particular Provider Region. A PE Device must first be added as a Device and then assigned a PE Device type.

All Provider Edge (PE) and Provider (P) routers in the MPLS network must be added to the Prime Fulfillment inventory. Each Provider Edge router should be created as a Device and then as a PE Device with a Role Type of N-PE (Network-facing PE). Each Provider device should be created as a Device and then as a PE Device with a role type of P (Provider). Adding customer premises equipment (CPE) devices to the Prime Fulfillment inventory is optional.

Note Where a Device is acting as both a Provider and Provider Edge Device it should be created as a PE Device with a Role Type of N-PE (Network-facing PE).

Many MPLS VPN networks employ a Route Reflector. It is recommended that Route Reflectors should be added to the Prime Fulfillment inventory. A Route Reflector should be added as a Device and then as a PE Device with role type of P. By adding the Route Reflector to the Prime Fulfillment inventory, Diagnostics is able to identify possible failures involving this device.

Note If other Prime Fulfillment features are being used to manage the MPLS network, many of the required inventory objects might already exist. For example, if the Prime Fulfillment MPLS VPN feature is being used, the required Provider, Provider Region, and Provider Edge devices might already exist. In this case only the Provider devices must be added.

A number of options exist for creating the required inventory objects. These objects can be created manually through the Prime Fulfillment GUI, using the Prime Fulfillment Discovery functionality, using the Inventory Manager Import Devices functionality, or using third-party Operations Support System (OSS) client programs that utilize the Prime Fulfillment APIs. Each of these options is described in the following sections:

- Manual Creation, page 11-9
- Discovery, page 11-10
- Inventory Manager Device Import, page 11-10
- Prime Fulfillment APIs, page 11-11
- Prime Fulfillment APIs, page 11-11

Note When creating Devices, the Device access information (login and passwords) must match that configured on the physical device.

Manual Creation

Manual creation allows you to add objects to the Prime Fulfillment Repository by entering the required configuration through the Prime Fulfillment Graphical User Interface (GUI). Manual object creation is recommended where a small number of objects are being added to the Prime Fulfillment Repository. The sequence for manual object creation is shown below:

1. Create Provider
2. Create Provider Region
3. Create Devices

4. Collect Device configuration, including interface configuration
5. Create PE Devices, including assigning roles for Provider and Provider Edge devices

Note Both Provider (P) and Provider Edge (PE) devices should be added to the Prime Fulfillment repository as PE Device objects with an appropriate PE Role Type. For details of the PE Role Types that should be assigned to Provider and Provider Edge devices, see Inventory Setup, page 11-8. When selecting the transport mechanism to be used between the Prime Fulfillment server and the device, Cisco CNS Configuration Engine cannot be used with Diagnostics as it does not support the necessary commands that Diagnostics requires. If attempts are made to use Cisco CNS Configuration Engine with Diagnostics, then Diagnostics incorrectly reports that the device cannot be contacted.

For details of how to manually create Provider, Provider Region, Device and PE Device objects, see Setting Up Resources, page 2-40.

When manually creating Devices, you must also add the interface configuration for these devices.

Interface configuration can either be added manually during Device creation, or by using a Task Manager Collect Configuration task. For details of how to perform a Task Manager Collect Configuration task, see Device Configuration Collection, page 11-11. We recommend that you use a Collect Configuration task.

Discovery

Discovery allows you to add the devices in your network to the Prime Fulfillment Repository by configuring minimal device and topology information in XML files. The Discovery process then queries these devices and populates the Prime Fulfillment Repository with the required device and topology information. We recommend that Discovery is used where a large number of objects are being added to the Repository.

Prime Fulfillment Discovery provides two methods for discovering devices: CDP or Device/Topology. Before performing Device Discovery it is necessary to create the required Discovery XML configuration files. For details of how to discover devices, see Appendix G, “Inventory - Discovery.”

Note Both Provider (P) and Provider Edge (PE) devices should be added to the Prime Fulfillment repository as PE Device objects with an appropriate PE Role Type. For details of the PE Role Types that should be assigned to Provider and Provider Edge devices, see Inventory Setup, page 11-8.

Note After Discovery has completed, you must run a Task Manager Collect Configuration task for all discovered devices. If you do not run a Collect Configuration task, Diagnostics is unable to log in to the discovered devices to perform troubleshooting. For details of how to perform a Task Manager Collect Configuration task, see Device Configuration Collection, page 11-11.

Inventory Manager Device Import

The Inventory Manager Import Devices feature allows you to import multiple devices in to the Prime Fulfillment Repository from files containing the Cisco IOS running configuration of the devices. We recommend that the Inventory Manager Import Devices feature is used where a large number of objects are being added to the Repository. For details of how to import devices, see Appendix G, “Inventory - Discovery.”

Before importing Provider (P) and Provider Edge (PE) devices you must create the required Provider and Provider Region objects. For details of how to manually create Provider and Provider Region objects, see Appendix G, "Inventory - Discovery."

When importing devices you must specify the directory where files containing the Cisco IOS running configuration are located. Do not specify the file names. The files must be located in a file system directory accessible from the Prime Fulfillment server.

Note Both Provider (P) and Provider Edge (PE) devices should be added to the Prime Fulfillment repository as PE Device objects with an appropriate PE Role Type. For details of the PE Role Types that should be assigned to Provider and Provider Edge devices, see Inventory Setup, page 11-8.

Note The enable secret password is encrypted before it is added to the Cisco IOS running configuration. As a result, the Device Import feature is unable to set the enable secret password for devices imported in to the Prime Fulfillment Repository. If the enable secret password is set on any devices being imported, you must manually configure the enable password for these devices in the Prime Fulfillment Repository. If both the enable and enable secret passwords are set for a device, the Inventory Manager Import Devices feature uses the enable password for the device added to the Prime Fulfillment Repository. You must override this password with the correct enable secret password. The enable password for devices in the Prime Fulfillment Repository can be set during or after device import.

Note After Device Import has completed, you must run a Task Manager Collect Configuration task for all imported devices. If you do not run a Collect Configuration task, Diagnostics is unable to log in to the imported devices to perform troubleshooting. For details of how to perform a Task Manager Collect Configuration task, see Device Configuration Collection, page 11-11.

Prime Fulfillment APIs

The Prime Fulfillment application program interface (API) allows you to use operations support system (OSS) client programs to connect to the Prime Fulfillment system. The Prime Fulfillment APIs provide a mechanism for inserting, retrieving, updating, and removing data from Prime Fulfillment servers. It is possible to add the required Provider, Provider Region, Device and PE Device objects using the APIs.

Note The Prime Fulfillment API is not included as standard with Diagnostics, it can be purchased separately.

For details of how to use the Prime Fulfillment APIs, see the *Cisco Prime Fulfillment API Programmer Guide 6.2* and the *Cisco Prime Fulfillment API Programmer Reference 6.2*.

Device Configuration Collection

We recommend that a Task Manager Collect Configuration task is used to add interface configuration to Devices in the Prime Fulfillment Repository. A Task Manager Collect Configuration task connects to the physical device in the network, collects the device information from the router (including interface configuration), and populates the Prime Fulfillment Repository with this information.

For details of how to add Device interface configuration using a Task Manager Collect Configuration task, see Task Manager, page 10-23.

Synchronizing the Prime Fulfillment Repository with Device Configuration

Note The accuracy of Diagnostics is dependent on up-to-date device information. We recommend that the device configuration is resynchronized with the physical devices after any configuration changes and at periodic intervals. This ensures that the device configuration held in the Prime Fulfillment inventory is consistent with the physical devices in the network.

We recommend that device configuration is kept up-to-date using a scheduled Task Manager task. Either Collect Configuration or Collect Configuration from File can be used. For details of how to create a scheduled Task Manager Collect Configuration task, see Task Manager, page 10-23. All PE and P routers in the MPLS network should have their configuration collected using a scheduled Task Manager Collect Configuration task. The Task Manager Collect Configuration task collects details of interface configuration and other device attributes. The interval at which Task Manager Collect Configuration tasks should be scheduled to run depends on the frequency of configuration changes to the network. We recommend running the Task Manager Collect Configuration task daily on each P and PE router.

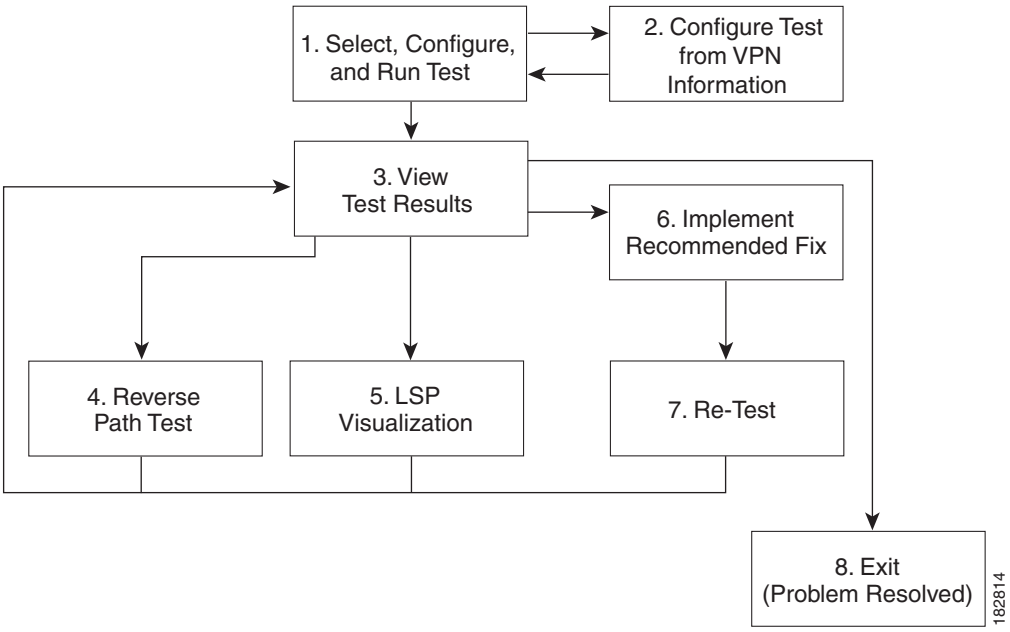
Using Cisco MPLS Diagnostics Expert

This section describes how to use Diagnostics.

It contains the following sections:

Figure 11-2 describes the workflow for using Diagnostics.

Figure 11-3 Using Diagnostics Workflow



1. Select, Configure, and Run Test—Configure and run an MPLS VPN Connectivity Verification test. See Performing an MPLS VPN Connectivity Verification Test, page 11-18.
2. Configure Test from VPN Information—Optionally configure an MPLS VPN Connectivity Verification test using VPN information. This is only possible if Prime Fulfillment VPN Provisioning functionality is used to provision VPNs within the network. See Configuring Using Customer VRF Information, page 11-27 and Configuring Using Customer VPN/VRF Information, page 11-29.
3. View Test Results—View results of MPLS VPN Connectivity Verification test, including the Test Log. See Interpreting the Test Results, page 11-37.
4. Reverse Path Test—Perform Reverse Path Test advanced troubleshooting. See Reverse Path Testing, page 11-44.
5. LSP Visualization—Perform LSP Visualization advanced troubleshooting. See LSP Visualization, page 11-44.
6. Implement Recommended Fix—Manually implement fix as recommended by test results.
7. Retest—Rerun the MPLS VPN Connectivity Verification test. This would typically be done to verify the fix implemented.

Understanding the Diagnostics Connectivity Tests

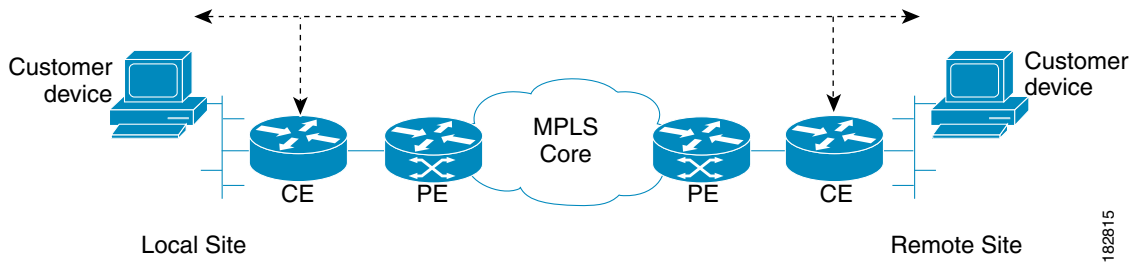
The connectivity tests are designed to troubleshoot subsections of the overall CE to CE network. The provided connectivity tests are as follows:

1. L3VPN - CE to CE—Checks the MPLS VPN connectivity between two CEs. See L3VPN - CE to CE Connectivity Test, page 11-14
2. L3VPN - PE to attached CE—Checks the MPLS VPN connectivity between a PE and the attached CE. See L3VPN - PE to Attached CE Connectivity Test, page 11-15
3. L3VPN - CE to PE across Core—Checks the MPLS VPN connectivity between a CE and a PE across the MPLS core. See L3VPN - CE to PE Across Core Connectivity Test, page 11-16
4. L3VPN - PE to PE (in VRF)—Checks the MPLS VPN connectivity between two PEs. See L3VPN - PE to PE in VRF Connectivity Test, page 11-16
5. MPLS - PE to PE —Checks the MPLS Core connectivity between two PEs. See L3VPN - PE to PE Connectivity Test, page 11-17

L3VPN - CE to CE Connectivity Test

The L3VPN - CE to CE test (Figure 11-4) checks the MPLS VPN connectivity between two CEs or Customer devices where the Customer device IP address is known.

Figure 11-4 L3VPN - CE to CE Connectivity Test



Diagnostics performs core, edge, and attachment circuit troubleshooting in this case.

IPv6 troubleshooting

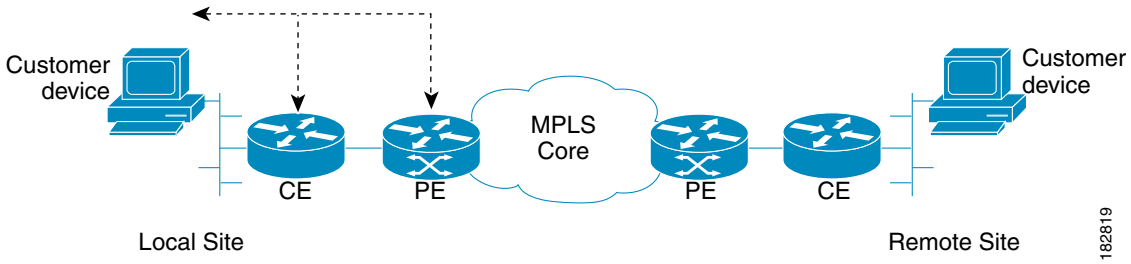
A L3VPN - CE to CE test launches troubleshooting on the IPv6 segment when all the following conditions are met:

- If a row with global unicast IPv6 address is selected from the interface selection screen or if the specified local and remote PE access circuit interfaces are having a global unicast IPv6 address or when the interface details are not available in the database.
- If the specified CE access circuit interface IP address for both local and remote-site is a global unicast IPv6 address.
- Optionally, if the specified customer device IP address for both local and remote site or for local or remote site is a global unicast IPv6 address.

L3VPN - PE to Attached CE Connectivity Test

The L3VPN - PE to attached CE connectivity test (Figure 11-5) performs a VPN connectivity test between a PE and the locally attached CE. Diagnostics performs edge and attachment circuit troubleshooting in this case.

Figure 11-5 L3VPN - PE to Attached CE Connectivity Test



The L3VPN - PE to attached CE connectivity test cannot be run in the reverse direction.

The local attachment circuit is often responsible for a connectivity failure. You can test the local attachment circuit on its own, without requiring remote site PE and CE details that might not be available.

The L3VPN - PE to attached CE connectivity test allows you to diagnose the same attachment circuit connectivity outage reported by a VRF-aware IP SLA probe. The notification has all the information required to set up the corresponding access circuit connectivity test in Diagnostics.

IPv6 troubleshooting

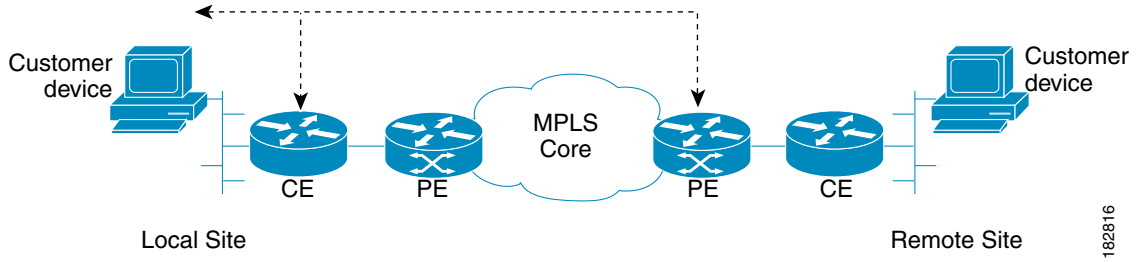
A L3VPN - PE to attached CE test launches troubleshooting on the IPv6 segment when all the following conditions are met:

- If a row with global unicast IPv6 address is selected from the interface selection screen or if the specified PE access circuit interface is having a global unicast IPv6 address or when the interface details are not available in the database.
- If the specified CE access circuit interface IP address is a global unicast IPv6 address.
- Optionally, if the specified Customer device IP address is a global unicast IPv6 address.

L3VPN - CE to PE Across Core Connectivity Test

The L3VPN - CE to PE across core connectivity test (Figure 11-6) checks the MPLS VPN connectivity between a CE or Customer devices (where the Customer device IP address is known), and a PE across the MPLS core.

Figure 11-6 L3VPN - CE to PE Across Core Connectivity Test



Diagnostics troubleshoots the core, both edges, and the attachment circuit in this case.

IPv6 troubleshooting

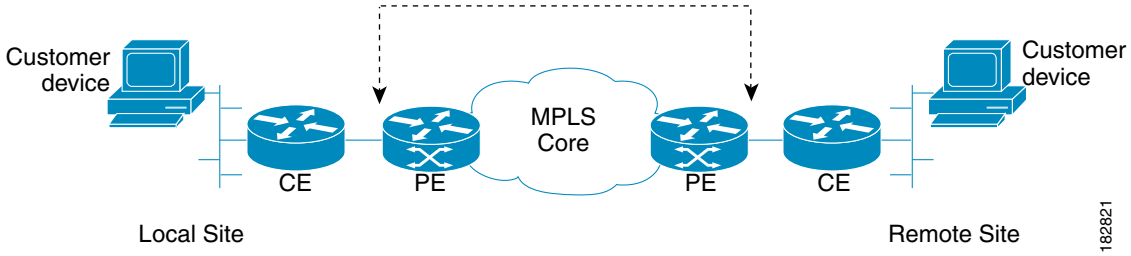
A L3VPN - CE to PE across core test launches troubleshooting on the IPv6 segment when all the following conditions are met:

- If a row with global unicast IPv6 address is selected from the interface selection screen or if the specified PE access circuit interface is having a global unicast IPv6 address or when the interface details are not available in the database.
- If the specified CE access circuit interface IP address is a global unicast IPv6 address.
- Optionally, if the specified customer device IP address is a global unicast IPv6 address.
- If the selected or specified PE access circuit interface is having a global unicast IPv6 address or when the interface details are not available in the database.

L3VPN - PE to PE in VRF Connectivity Test

The L3VPN - PE to PE in VRF connectivity test (Figure 11-7) checks the MPLS VPN connectivity between two PEs. Diagnostics troubleshoots the core and the edge on both sides.

Figure 11-7 L3VPN - PE to PE in VRF Connectivity Test



Some organizations provision the core or edge network but do not immediately allocate CEs. The L3VPN - PE to PE connectivity in VRF test allows you to deploy and test your network in phases. This test option also provides more flexibility and allows the edge or core network segment to be tested when CE information is not readily available.

The L3VPN - PE to PE connectivity in VRF connectivity test also allows you to diagnose the same short reach (PE to remote PE) VPN connectivity outage reported by a VRF-aware IP SLA probe. The notification has all the information to set up the corresponding edge connectivity test in Diagnostics.

IPv6 troubleshooting

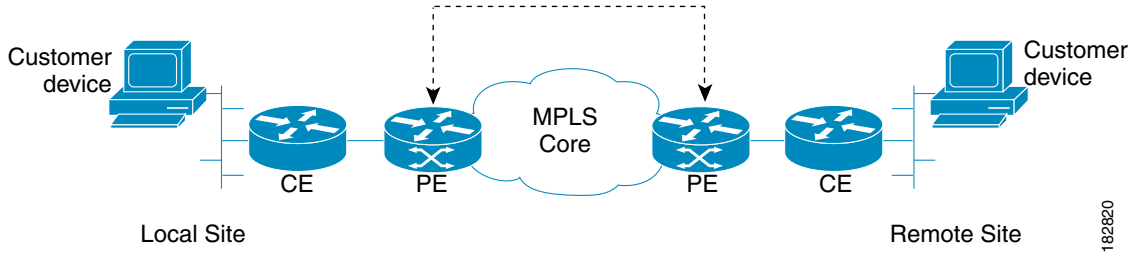
A L3VPN - PE to PE in VRF test launches troubleshooting on the IPv6 segment when all the following conditions are met:

- Either the local site PE access circuit interface or the remote site PE access circuit interface with global unicast IPv6 address needs to be selected from the interface selection screen.
- If a row with global unicast IPv6 address is selected from the interface selection screen or if the specified local PE access circuit interface is having only a global unicast IPv6 address.
- If a row with global unicast IPv6 address is selected from the interface selection screen or if the specified remote PE access circuit interface IP address is having only a global unicast IPv6 address.

L3VPN - PE to PE Connectivity Test

The L3VPN - PE to PE core connectivity test (Figure 11-8) checks the MPLS connectivity between two PEs.

Figure 11-8 L3VPN - PE to PE Core Connectivity Test



The L3VPN - PE to PE core test is intended for cases where there is blocked access to the CE interface, such as using an access list, or cases where different groups within an organization are responsible for different network segments. For example, a Core group might have a P issue but does not have the end customer context to perform a full CE-CE or PE-PE test.

The L3VPN - PE to PE core test allows you to diagnose the same core connectivity outage reported by IP SLA Health monitor probes testing connectivity between MPLS enabled PEs. The notification has all the information to set up the corresponding core connectivity test in Diagnostics.

IPv6 troubleshooting

In case of a L3VPN - PE to PE in core test, an IPv6 troubleshooting cannot be initiated as this test type uses only the IPv4 address.

Performing an MPLS VPN Connectivity Verification Test

This section describes how to perform an MPLS VPN Connectivity Verification test. This section contains the following information:

Note For every command executed on a device with IOS XR version 3.8.0 or onwards, the first line of the output shows the current time stamp of the device, which Diagnostics fails to handle. The *timestamp disable* command should be used to disable the time stamp on XR devices before launching a test.

Opening the MPLS Diagnostics Expert Feature Selection Window

Note When performing parallel MPLS VPN Connectivity Verification tests on the same client machine, ensure each test is performed using a different HTTP session. To do so, run each test in a separate browser, launched from the command line, or by clicking on the browser icon on the desktop, or Start menu. Do not run parallel tests in tabs within the same browser window or in browser windows launched from existing browser windows.

Step 1 Log in to Prime Fulfillment. For details of how to log in, see the *Cisco Prime Fulfillment Installation Guide 6.2* (**Installing and Logging Into Prime Fulfillment > Logging In for the First Time**).

The Prime Fulfillment home window appears.

Step 2 Click the Diagnostics tab.

The MPLS Diagnostics Expert Feature Selection window displaying the available MPLS VPN connectivity verification test types appears.

Note You must check that you have at least one Diagnostics user role assigned to you, see User Roles, page 11-7.

Note The tests types available to you are determined by your assigned user roles. A user role must be defined for each test type. If you do not have access to a test type, that test type does not appear on the MPLS Diagnostics Expert Feature Selection window. See User Roles, page 11-7 for further information.

Selecting, Configuring, and Running a L3VPN - CE to CE Test

This section details how to select, configure, and run a L3VPN - CE to CE test type.

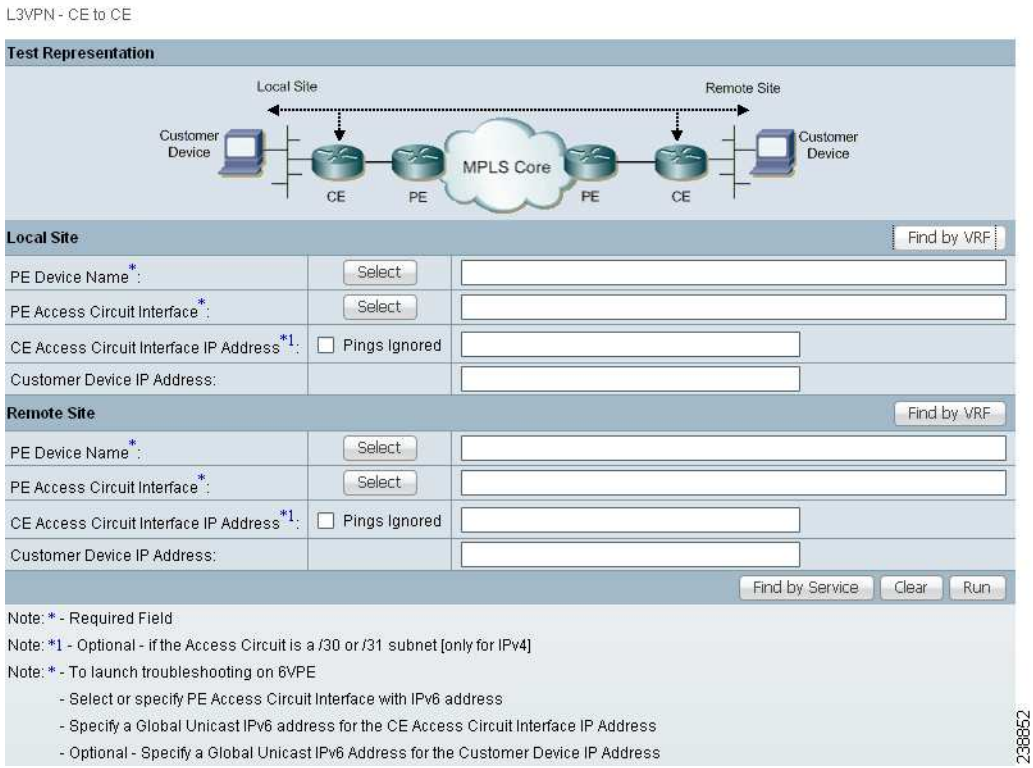
Step 1 From the Diagnostics menu, select the L3VPN - CE to CE test type.

Step 2 Click on the L3VPN - CE to CE connectivity verification test type.

See L3VPN - CE to CE Connectivity Test, page 11-14 for information on L3VPN - CE to CE connectivity verification test type. The L3VPN - CE to CE window appears displaying the input window corresponding to the L3VPN - CE to CE test type.

Tip Each available test type has its own input window and requests a different sets of parameters, for example, the L3VPN - CE to CE test requires information for both the local and the remote sites, while the test set up window for a L3VPN - PE to attached CE test only requires local site details.

Figure 11-9 L3VPN - CE to CE Test Type



The L3VPN - CE to CE window allows you to configure the connectivity test you would like to perform.

This window displays the following components:

- Network diagram
- Local Site configuration area
- Remote Site configuration area

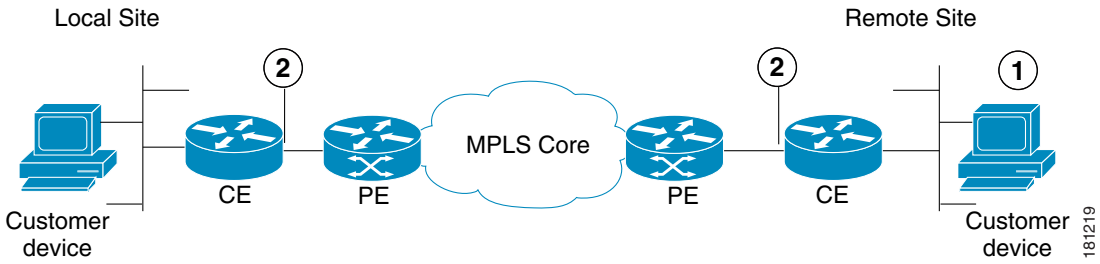
The network diagram is a static image that provides you with context for the information you must enter to configure the test.

MPLS VPN Connectivity Verification tests connectivity between two sites in a VPN. Throughout the test, these sites are referred to as the local site and remote site. It is anticipated that a connectivity problem will be reported or detected from the perspective of a particular site. This particular site would typically be used as the local site, and the test is performed from this site. However, this is not mandatory, as any site can be used as the local or remote site, because connectivity can be tested in both directions.

The scope of the L3 VPN connectivity test (see Figure 11-10) can be changed on a per-site basis. For each site you can test connectivity to a customer device within the site (shown in Figure 11-10 as *1*), or to the CE access circuit interface (shown in Figure 11-10 as *2*). The test scope is determined by the configuration that you provide.

Where the IP address of a customer device is known, it might be desirable to perform a connectivity verification test to that device. Where the IP address of a customer device is not known, the connectivity verification test can be performed to the CE for the site.

Figure 11-10 Test Scope



1. Customer device.
2. CE access circuit interface.

To test connectivity to a device within the customer site subnetwork, you should enter the IP address of the device in the Customer Device IP Address field. By default, if you specify only the required fields for a site, the test is performed to the CE access circuit interface.

Note Required fields are denoted by a blue asterisk in the L3VPN - CE to CE Diagnostics - Test Setup window. You are unable to continue until all required fields have been completed with valid information.

Note Diagnostics automatically populates the CE Access Circuit Interface IP Address field if /30 or /31 addressing is used.

Cisco IOS and Cisco IOS XR Access Control Lists (ACL) allow selected traffic to be blocked based on a wide variety of criteria. ACLs configured on the CE can lead to inconsistent results being reported when an MPLS VPN Connectivity Verification test is performed to a customer device or CE interface. Where possible, an MPLS VPN Connectivity Verification test reports that traffic is blocked by an ACL configured on the CE device. However, depending on ACL configuration, it is not always possible to determine that traffic is blocked by an ACL configured on the CE device. In some cases an MPLS VPN Connectivity Verification test might report an access circuit failure or unknown failure. In cases where it is suspected that traffic is being blocked at the CE, the Pings Ignored check box should be checked for that site. This allows Diagnostics to take account the blocking access ACL when troubleshooting and therefore return a more accurate diagnosis of any problem found.

Note When checking the Pings Ignored check box for a site, the CE IP address and optionally the Customer Device IP Address field are used to perform troubleshooting and configuration checks on the PE device.

Step 3 Configure the fields in the L3VPN - CE to CE window as required.

Table 11-3 provides field descriptions of the L3VPN - CE to CE window.

Note The fields displayed depend on the type of test you selected, for example, the CE to CE test requires information for both the local and the remote sites, while the test set up window for a PE to attached CE test only requires local site details.

Note An alternative way to configure the test is to use customer VPN information. See Configuring Using Customer VPN/VRF Information, page 11-29 for further information.

Table 11-3 Field Descriptions for the L3VPN - CE to CE Diagnostics - Test Setup Window

Field	Valid for Test Type	Description
Find by VRF	All	Click the Find by VRF button to configure the test using PE hostname or PE interface details identified using a VRF search. (See the "Configuring Using Customer VRF Information" section on page 11-27.)
PE Device Name	All	Enter the site PE Device Name in the PE Device Name field or select the site PE Device Name by clicking the Select button. **Note** Clicking the Select button opens the Select PE Device window. (See the "Selecting a PE Device" section on page 11-23). The Device Name is the fully qualified hostname and domain name of the device. For example, router1.cisco.com. However, the domain name is optional so in many cases the Device Name is the device hostname. For example, router1. The Device Name specified must match that of a PE device with role type of N-PE.

Table 11-3 Field Descriptions for the L3VPN - CE to CE Diagnostics - Test Setup Window (continued)

Field	Valid for Test Type	Description
LSP Endpoint Loopback IP Address	L3VPN - PE to PE Core only	Enter the BGP next hop if different from the BGP router ID of the peer PE. You can enter the loopback IP address, or you can enter the loopback name that will be resolved to the IP address. When testing the core, an MPLS OAM ping and trace is performed from the local PE to the remote PE. The destination of this ping causes an LSP to be selected based on the routing information on the local PE. Customer traffic uses the BGP next hop address of the customer route as its destination, and to select the LSP. Make sure that the IP prefix Diagnostics tests to matches the BGP next hop address used by the customer traffic. This ensures that Diagnostics tests the same LSP as the customer traffic traverses. In the case of L3VPN - PE to PE core testing, Diagnostics does not have any customer route information. Diagnostics therefore has no way to determine the BGP next hop and chooses the ping destination, not based on the next hop, but on the BGP router ID on the remote PE. In some network configurations, this router ID does not match the next hop used by the customer traffic and the incorrect (or no) LSP is tested. This happens when: • The BGP router ID is the address of a loopback that has no LSP assigned to it. • The BGP router ID is not the address of a loopback. • The customer has several LSPs defined and the customer traffic is using a different LSP than the router ID gives. • The customer has several LSPs defined and the customer traffic switches LSP based on a routemap. In the above bullet points you need to provide the correct BGP next hop. **Note** By specifying the LSP Endpoint Loopback IP Address, Diagnostics has the capability to test and detect core failures on multiple LSPs in the MPLS core. See the "Configuring the LSP Endpoint Loopback IP Address for a MPLS - PE to PE Test" section on page 11-34 for further information.
PE Access Circuit Interface	L3VPN - CE to CE L3VPN - PE to attached CE L3VPN - CE to PE across Core L3VPN - PE to PE in VRF	Enter the interface name of the PE Access Circuit Interface in the PE Access Circuit Interface field or select the PE Access Circuit Interface by clicking the Select button. **Note** Clicking the Select button opens the Select Device Interface window (see the "Selecting a PE Access Circuit Interface" section on page 11-24). You must specify a valid PE Device Name before selecting the PE Access Circuit Interface. The interface specified should be the access circuit interface attached to the site's CE. The interface name specified must match an interface on the device, but the interface does not necessarily need to be in the Prime Fulfillment device inventory.

Table 11-3 Field Descriptions for the L3VPN - CE to CE Diagnostics - Test Setup Window (continued)

Field	Valid for Test Type	Description
CE Access Circuit Interface IP Address	L3VPN - CE to CE L3VPN PE to attached CE L3VPN - CE to PE across Core	Enter the IP address of the CE access circuit interface for the local site. This should be the access circuit interface attached to the specified PE. When a PE Access Circuit Interface configured using IPv4 addressing and with a /30 subnet mask (255.255.255.252) or a /31 subnet mask (255.255.255.254) is selected, the CE Access Circuit Interface IP Address field is auto-completed with the remaining host address from that /30 or /31 subnet. When a PE Access Circuit Interface configured with a /31 mask (255.255.255.254) subnet mask has been manually entered, an attempt to derive the CE access circuit interface IP address is only made after the test is initiated. In this instance, the CE Access Circuit Interface IP Address field is not auto-completed before the OK button is clicked. It is not possible to derive the correct CE access circuit interface IP address in cases where the PE access circuit interface is using IP unnumbered or the CE access circuit interface is on a different subnet. The test supports managed and unmanaged Cisco CE devices, and non-Cisco CE devices.
Pings Ignored	L3VPN - CE to CE L3VPN - PE to attached CE L3VPN - CE to PE across Core	Check this check box to specify that there is an ACL configured on the CE that will ignore ping and trace route packets originating from the provider core network.
Customer Device IP Address	L3VPN - CE to CE L3VPN - PE to attached CE L3VPN - CE to PE across Core	Enter the IP address of a customer device on the local site customer network. Entering the customer device IP address causes the connectivity test to be performed to this device.
Find by Service	All	Click the Find by Service button to open the Populate using VPN/VRF window. The Populate using VPN/VRF window allows you to configure the test using customer VPN/VRF information (see the "Configuring Using Customer VPN/VRF Information" section on page 11-29.)
OK button	All	Click OK to run the test.
Clear button	All	Click Clear to reset all the fields in the window.

Step 4 Click **OK** to run your test after all the required fields are completed.

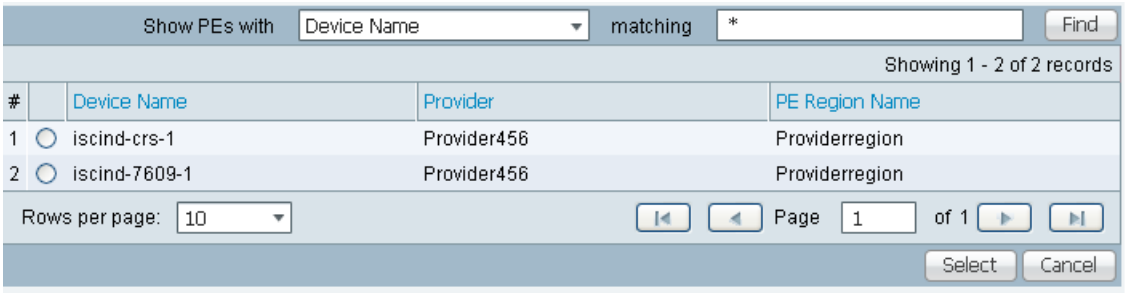
The Progress window appears. See the "Progress Window" section on page 11-37.

Selecting a PE Device

Click the **Select** button (for the Local/Remote PE Device Name) to open the Select PE Device window (see Figure 11-11) where you can choose the local/remote site PE. The Select PE Device window displays a table containing all the PE devices available in the inventory.

Note You can configure the default value of the Diagnostics device selector, as shown in Figure 11-11. Possible values are Device Name, Provider, and PE Region Name.

Figure 11-11 Select PE Device Window

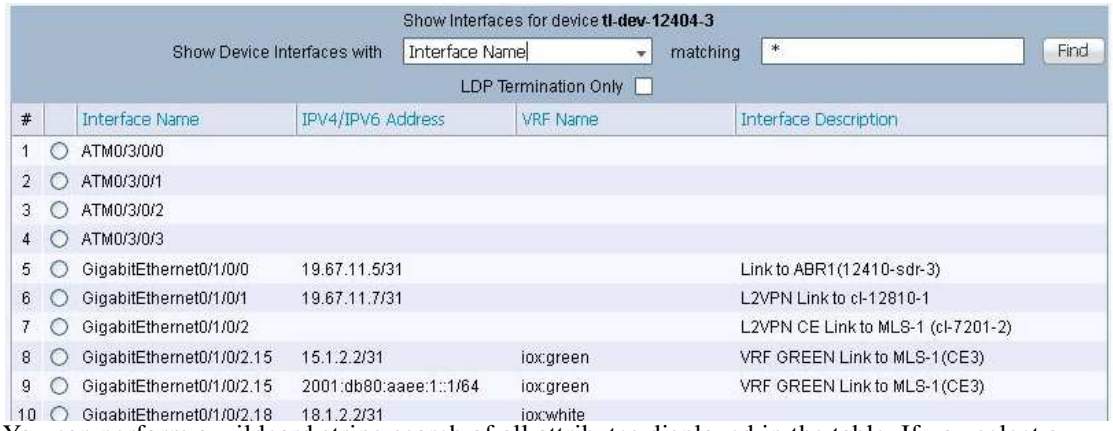


Note You can perform a wildcard string search of all PE attributes displayed in the PE table. If you select a local/remote site PE from the Prime Fulfillment inventory, this overrides anything entered in the Local/Remote PE Device Name field (see Figure 11-9.) This search feature is useful in large networks, where you have a large number of PEs.

Selecting a PE Access Circuit Interface

Click the **Select** button (for the Local/Remote PE Access Circuit Interface) to open the Select Device Interface window (see Figure 11-12) where you can choose the interface name. The Select Device Interface window displays a table containing all interfaces for the selected local/remote PE device.

Figure 11-12 Select Device Interface Window



You can perform a wildcard string search of all attributes displayed in the table. If you select a Local/Remote PE Access Circuit Interface from the Prime Fulfillment inventory, this overrides anything entered in the Local/Remote PE Access Circuit Interface field (see Figure 11-9).

Table 11-4 provides field descriptions for the Select Device Interface window.

Timesaver

Enter an appropriate search pattern first using the Show Device Interfaces with the drop-down box and the matching field (see Figure 11-12). This saves large, time-consuming, and unnecessary searches which could occur in large networks. Table 11-4 provides field descriptions for the Select Device Interface window.

Table 11-4 Field Descriptions for the Select Device Interface Window

Field	Description
Show Device Interfaces with	The Show Devices with drop-down box allows you to refine your search results. Select Interface Name, IPV4 Address, IPV6 Address, VRF Name or Interface Description from the drop-down menu to select the category to further refine the results of your search.
matching (optional field)	Enter information into the matching field to refine your search further within the category you selected in the Show Devices with drop-down box. You can enter text as a partial string; wildcards are also supported.
LDP Termination Only	The LDP Termination Only check box is used to filter for LDP terminating loopback interfaces in cases where selection of an LDP terminating loopback interface is required. This check box should be left unchecked.
Find	Click **Find** to run your search using the information you configured in the Select Device Interface window.
Interface Name	Displays the list of interfaces found after you have run your search. Click on the Interface Name column heading to sort your list of interface names.
IPV4/IPV6 Address	Displays the list of IPV4/IPV6 addresses found after you have run your search. Click on the IPV4/IPV6 Address column heading to sort your list of IPV4/IPV6 addresses. You can choose the IPV6 address either by selecting it from the existing list or by manually entering it.
VRF Name	Displays the list of VRF names found after you have run your search. Click on the VRF Name column heading to sort your list of VRF names.
Interface Description	Displays the list of interface descriptions found after you have run your search. Click on the Interface Description column heading to sort your list of interface descriptions.
Row per page	Displays the row number of the rows displayed in the table. Click the corresponding radio button to select a row in the table.
Select	Click Select to confirm your selection in the table. The L3VPN - CE to CE Diagnostics - Test Setup Window appears with the PE Access Circuit Interface fields populated with the values you selected in the table.
Cancel	Click **Cancel** to close the Select Device for VRF Search window.

Tip We recommend using the Interface Description to describe customer connection details. Diagnostics allows you to search on the Interface Description, for example, on a customer circuit ID. See the "Selecting, Configuring, and Running a L3VPN - CE to PE Across Core Test" section on page 11-31, and the "Selecting, Configuring, and Running a L3VPN - PE to PE Test" section on page 11-32 for information.

Testing Across Cisco IOS Multilink Access Circuit Interfaces

Diagnostics supports troubleshooting across Cisco IOS multilink access circuit interfaces. Troubleshooting is performed on the multilink bundle interface only. No troubleshooting of the individual bundle links or multilink specific troubleshooting is performed. The following multilink technologies are supported:

- Multilink PPP over Frame Relay (Multilink group interface configuration)
- Multilink PPP over Frame Relay (Virtual-Template interface configuration)
- Multilink PPP over ATM (Multilink group interface configuration)
- Multilink PPP over ATM (Virtual-Template interface configuration)
- Multilink PPP over Serial
- Multilink Frame Relay

Note Multilink is supported in Cisco IOS only and not Cisco IOS XR.

Note No Layer 2 Frame Relay, ATM, or Ethernet troubleshooting is performed for multilink access circuit interfaces.

Each multilink bundle has a number of interfaces associated with it. When configuring an MPLS VPN Connectivity Verification test over a multilink access circuit, you must ensure you enter the correct interface in the PE Access Circuit Interface field of the MPLS VPN Test Configuration window. The interface which you must enter varies depending on the multilink configuration used. Table 11-5 details the interface that must be entered in the PE Access Circuit Interface field for each multilink technology.

Table 11-5 Multilink Interfaces

Multilink Technology	PE Access Circuit Interface
ML-PPPoFR (Multilink Group)	Multilink interface representing the multilink bundle.
ML-PPPoFR (Virtual-Template)	Virtual-Access interface representing the multilink bundle.
ML-PPPoATM (Multilink Group)	Multilink interface representing the multilink bundle.
ML-PPPoATM (Virtual-Template)	Virtual-Access interface representing the multilink bundle.
ML-PPPoSerial	Multilink interface representing the multilink bundle.
ML-FR	Frame Relay interface on which the Virtual Circuit is configured. This might be the Multilink Frame Relay (MFR) interface or a Frame Relay subinterface on the MFR interface.

With the exception of Multilink Frame Relay (MFR), the interface that represents the multilink bundle must be entered in the PE Access Circuit Interface field. For Multilink Frame Relay, the Frame Relay interface, or subinterface against which the Virtual Circuit is configured must be entered. This might be the MFR interface or a subinterface of the MFR interface. In all cases the interface entered in the PE Access Circuit Interface field should have an IP address and VRF and be in the up/up state.

To determine the valid multilink bundle interfaces on a PE device, use the **show ppp multilink** or **show frame-relay multilink** IOS command. If there are no active multilink bundles on your PE device, then there might be none configured or all bundle links for any configured multilink bundles might be in the down/down state.

Note Virtual-Access interfaces are dynamically created and assigned. The multilink bundle to which a Virtual Access interface belongs and the role it plays can change as interface states change. As a result Virtual Access interfaces are not stored in the Prime Fulfillment/Diagnostics repository. When configuring a VPN Connectivity Verification Test using a Virtual Access interface, you must manually enter the interface name into the PE Access Circuit Interface field of the MPLS VPN Test Configuration window. It is not possible to select Virtual Access interfaces from the Interface Selection popup dialog box.

Configuring Using Customer VRF Information

You need to supply PE hostname or PE interface details when entering information into the MPLS VPN Connectivity Verification window. In certain instances, you might not know the PE hostname or PE interface details. However, this information can be identified through a corresponding and known VRF name. You can identify a corresponding VRF name using a VRF search.

Note To successfully find an interface by VRF Name, you must have previously run the Prime Fulfillment Task Manager Collect Configuration task to upload the VRF names into Prime Fulfillment. The VRF search is based on the information within the latest Collect Configuration task run. For details of how to perform a Task Manager Collect Configuration task, see Device Configuration Collection, page 11-11.

Step 1 Click the **Find by VRF** button in the MPLS VPN Connectivity Verification window.

The Select Device for VRF Search window appears.

Note The fields displayed in the Select Device for VRF Search window are initially empty, regardless of whether any PE data fields have been populated or not.

Step 2 Configure the fields displayed in the Select Device for VRF Search window.

Table 11-6 provides field descriptions for the Select Device for VRF Search window.

Timesaver Enter an appropriate search pattern first. This saves large, time-consuming, and unnecessary searches which could occur in large networks. Enter a VRF name pattern and click the Find button. For example, entering *t** and clicking Find provides a list of all VRFs starting with the letter *t*. You can further filter your list of results by selecting from the Show Devices with drop-down box, entering information into the matching field, and clicking Find. Table 11-6 provides field descriptions for the Select Device for VRF Search window.

Table 11-6 Field Descriptions for the Select Device for VRF Search Window

Field	Description
VRF Search String	Enter a VRF name string to search on. You can enter the VRF name string as a partial string; wildcards are also supported.
Show Devices with	The Show Devices with drop-down box allows you to refine your search results. Select Device Name, Interface Name, IPV4 Address, IPV6 Address or Interface Description from the drop-down menu to select the category to further refine the results of your search.
matching (optional field)	Enter information into the matching field to refine your search further within the category you selected in the Show Devices with drop-down box. You can enter text as a partial string; wildcards are also supported.
Find	Click **Find** to run your VRF search using the information you configured in the Select Device for VRF Search window.
Device Name	Displays the list of device names found after you have run your search. Click on the Device Name column heading to sort your list of device names.
Interface Name	Displays the list of interfaces found after you have run your search. Click on the Interface Name column heading to sort your list of interface names.
IPV4/IPV6 Address	Displays the list of IPV4/IPV6 addresses found after you have run your search. Click on the IPV4/IPV6 Address column heading to sort your list of IPV4/IPV6 addresses. You can choose the IPV6 address either by selecting it from the existing list or by manually entering it.
VRF Name	Displays the list of VRF names found after you have run your search. Click on the VRF Name column heading to sort your list of VRF names.
Interface Description	Displays the list of interface descriptions found after you have run your search. Click on the Interface Description column heading to sort your list of interface descriptions.
Rows per page	Displays the row number of the rows displayed in the table. Click the corresponding radio button to select a row in the table.
Select	Click **Select** to confirm your selection in the table. The L3VPN - CE to CE Diagnostics - Test Setup window appears with the PE Device Name and PE Access Circuit Interface fields populated with the values you selected in the table.
Cancel	Click **Cancel** to close the Select Device for VRF Search window.

Step 3 Click **Find** to start your search.

The table displayed in the Select Device for VRF Search window is populated with your search results.

Tip Click on the column headings to sort the information displayed in each column.

Tip The table automatically widens when required to display the information displayed in the VRF Name and Interface Description columns. When the table widens, use the horizontal scrollbar to scroll to the right side of the window.

Step 4 (Optional) Refine your search results by configuring the Show Devices with drop-down box and the matching field.

Click **Find** to refresh the table with the results of your search.

Step 5 Click the radio button to select the PE Device Name and corresponding Interface Name you require.

Step 6 Click **Select**.

The Select Device for VRF Search window closes. The L3VPN - CE to CE Diagnostics - Test Setup window appears with the PE Device Name and PE Access Circuit Interface fields populated with the values you selected.

Configuring Using Customer VPN/VRF Information

Diagnostics can be used standalone, without any dependency on other Prime Fulfillment functionality. However, if Prime Fulfillment VPN/VRF Provisioning functionality is used to provision VPN/VRFs within the network, this provisioning information, associated with the customer and VPN/VRF, can be used as an alternative means to configure an MPLS VPN Connectivity Verification test. Rather than specifying device-specific configuration, you can specify a customer, VPN/VRF, local site, and remote site. All required test configuration is then derived from this information.

Note The option to configure an MPLS VPN Connectivity Verification test using customer VPN/VRF information is only available if the Prime Fulfillment VPN/VRF Provisioning functionality is used to provision VPN/VRFs within the network.

Step 1 Click the **Find by Service** button in the L3VPN - CE to CE Diagnostics - Test Setup window.

The Populate using VPN/VRF window appears.

Step 2 Configure the fields displayed in the Populate using VPN/VRF window.

Table 11-1 provides field descriptions for the Populate using VPN/VRF window.

Table 11-7 Field Descriptions for the Populate using VPN/VRF Window

Field	Description
Customer Details	
Customer Name	Click the Select button to select a customer from the Select Customer pop-up window.
VPN/VRF Name	Click the Select button to select a VPN/VRF name from the VPN/VRF name pop-up window. **Note** You must select a Customer Name before you can select a VPN/VRF Name.
Site Details	

Table 11-7 Field Descriptions for the Populate using VPN/VRF Window (continued)

Field	Description
Local Site	Click the Select button to select a Local Site from the Local Site pop-up window. **Note** You must select a Customer Name and a VPN/VRF Name before you can select a local site.
Remote Site	Click the Select button to select a Remote Site from the Remote Site pop-up window. **Note** You must select a Customer Name and VPN/VRF Name before you can select a remote site. **Note** The Remote Site field is not available for the PE to attached CE test type.

Step 3 Click **OK**.

The L3VPN - CE to CE Diagnostics - Test Setup window reappears. The required fields are populated based on the customer VPN/VRF information you provided in the Populate using VPN/VRF window.

Note If you want to test to a customer device, you can enter the IP address in the Local and/or Remote Site Customer Device IP Addresses fields.

Note You can edit any of the fields in the L3VPN - CE to CE Diagnostics - Test Setup window that have been automatically populated.

Step 4 Click **OK** on the L3VPN - CE to CE Diagnostics - Test Setup window to run the test.

The Progress window appears (see the "Progress Window" section on page 11-37).

VPN Topologies

By default, an MPLS VPN Connectivity Verification test assumes that the local and remote sites are connected through a full mesh VPN topology and that these sites can communicate directly. If the sites being tested are connected through a VPN topology other than full mesh, the required configuration for an MPLS VPN Connectivity Verification test might differ. In this situation, the test might produce misleading results, so you must take care when interpreting the test results. See VPN Topologies, page 11-49 for details of the configuration required and how the test results should be interpreted for each supported VPN topology.

Selecting, Configuring, and Running a L3VPN - PE to Attached CE Test

This section details how to select, configure, and run a L3VPN - PE to attached CE test type.

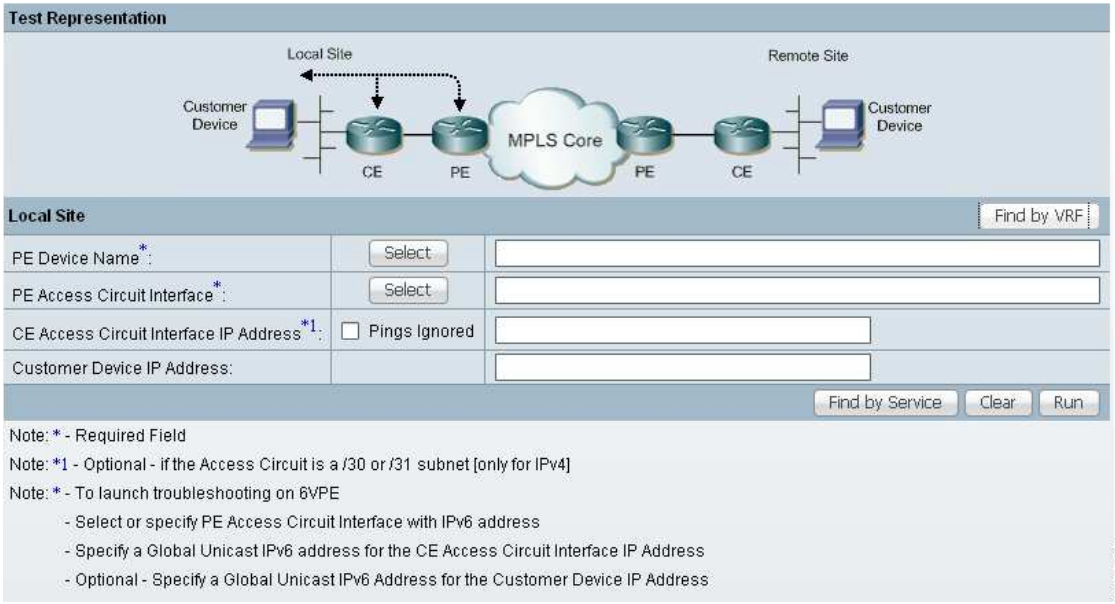
Step 1 From the Diagnostics menu, select the L3VPN - PE to Attached CE test type.

Step 2 Click on the L3VPN - PE to attached CE connectivity verification test type.

See the "L3VPN - PE to Attached CE Connectivity Test" section on page 11-15 for information on the PE to attached CE connectivity verification test type.

The MPLS VPN Connectivity Verification Configuration window appears (Figure 11-13) displaying the fields corresponding to the PE to attached CE test type. The MPLS VPN Connectivity Verification Configuration window allows you to configure the connectivity test you would like to perform.

Figure 11-13 L3VPN - PE to Attached CE Test Type



The L3VPN - PE to Attached CE window displays the following components:

- Network diagram
- Local Site configuration area

  These components and the test scope are described in further detail in the "Selecting, Configuring, and Running a L3VPN - CE to CE Test" section on page 11-19.

Step 3 Configure the fields in the L3VPN - PE to Attached CE window as required.

Table 11-3 on page 11-21 provides descriptions of the fields applicable to the L3VPN - PE to attached CE test type.

Step 4 Click **OK** to run your test after all the required fields are completed. The Progress window appears. See the "Progress Window" section on page 11-37.

Selecting, Configuring, and Running a L3VPN - CE to PE Across Core Test

This section details how to select, configure, and run a L3VPN - CE to PE across core test type.

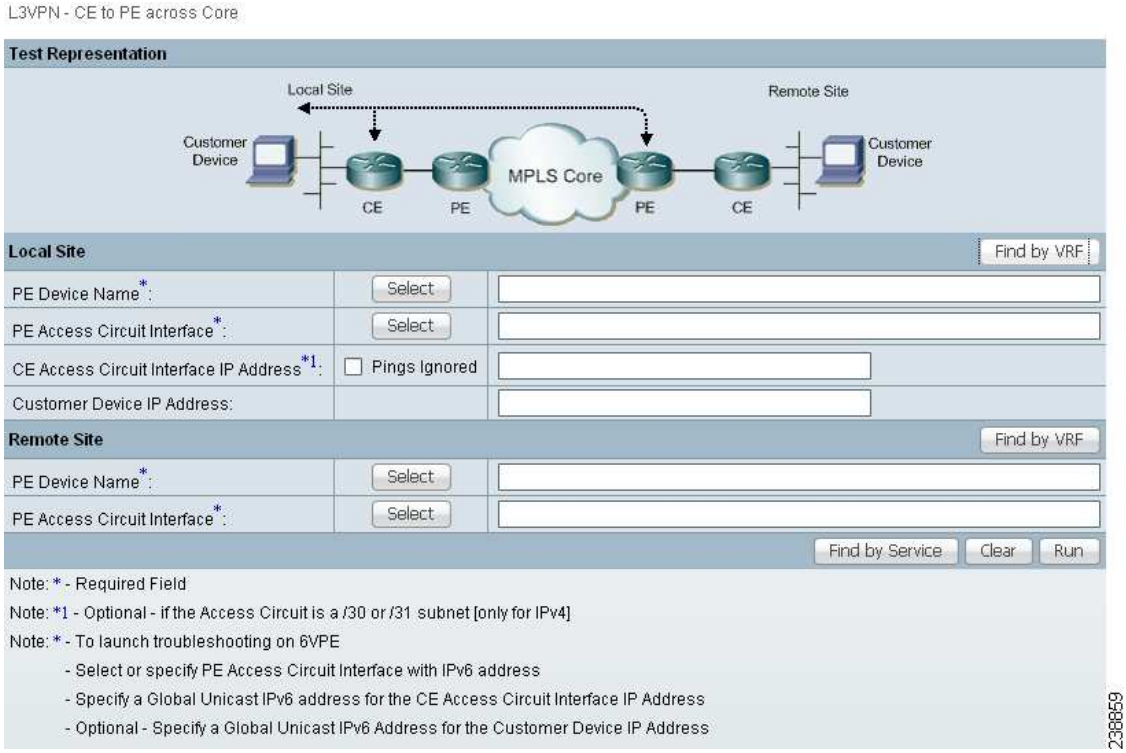
Step 1 From the Diagnostics menu, select the L3VPN - CE to PE across Core test type.

Step 2 Click on the L3VPN - CE to PE across Core connectivity verification test type.

See the "L3VPN - CE to PE Across Core Connectivity Test" section on page 11-16 for information on the L3VPN - CE to PE across core connectivity verification test type.

The L3VPN - CE to PE Across MPLS Core Diagnostics - Test Setup window appears (Figure 11-14) displaying the fields corresponding to the L3VPN - CE to PE across core test type. The L3VPN - CE to PE Across MPLS Core Diagnostics - Test Setup window allows you to configure the connectivity test you would like to perform.

Figure 11-14 L3VPN - CE to PE Across Core Test Type



The L3VPN - CE to PE Across Core window displays the following components:

- Network diagram
- Local Site configuration area
- Remote Site configuration area

  These components and the test scope are described in further detail in the "Selecting, Configuring, and Running a L3VPN - CE to CE Test" section on page 11-19.

Step 3 Configure the fields in the L3VPN - CE to PE Across Core window as required.

Table 11-3 on page 11-21 provides descriptions of the fields applicable to the L3VPN - CE to PE across core test type.

Step 4 Click **OK** to run your test after all the required fields are completed.

The Progress window appears. See the "Progress Window" section on page 11-37.

Selecting, Configuring, and Running a L3VPN - PE to PE Test

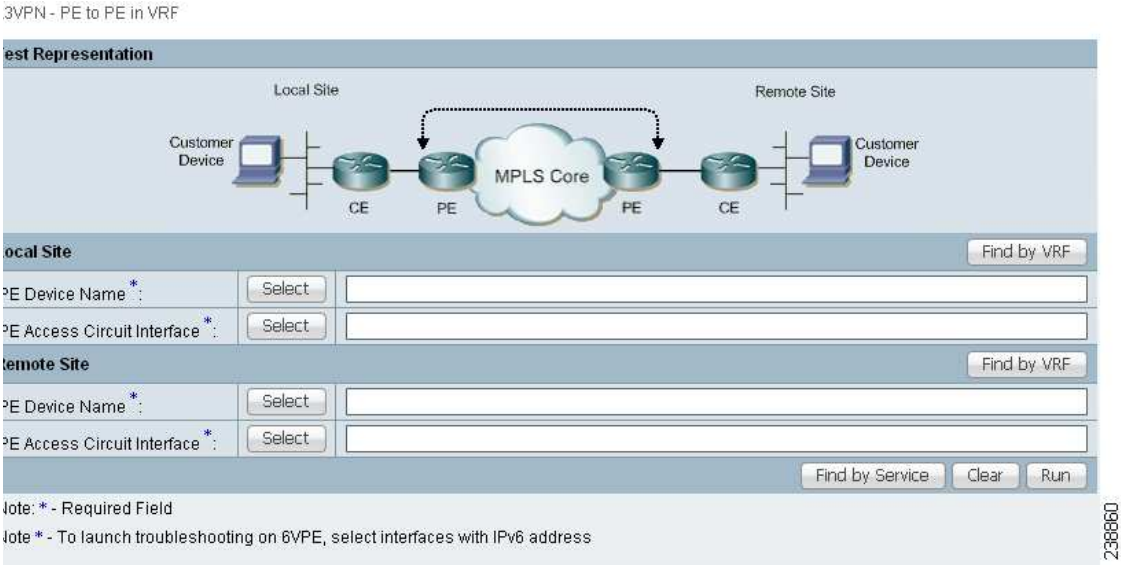
This section details how to select, configure, and run a L3VPN - PE to PE test type.

Step 1 From the Diagnostics menu, select the L3VPN - PE to PE test type.

See the "L3VPN - PE to PE in VRF Connectivity Test" section on page 11-16 for information on the L3VPN- PE to PE (in VRF) connectivity verification test type.

The L3VPN- PE to PE in VRF Diagnostics - Test Setup window appears (Figure 11-15) displaying the fields corresponding to the L3VPN - PE to PE in VRF test type. The L3VPN- PE to PE in VRF Diagnostics - Test Setup window allows you to configure the connectivity test you would like to perform.

Figure 11-15 L3VPN - PE to PE Test Type



The L3VPN - PE to PE in VRF Diagnostics - Test Setup window displays the following components:

- Network diagram
- Local Site configuration area
- Remote Site configuration area

  These components and the test scope are described in further detail in the "Selecting, Configuring, and Running a L3VPN - CE to CE Test" section on page 11-19.

Step 2 Configure the fields in the L3VPN - PE to PE in VRF Diagnostics - Test Setup window as required.

Table 11-3 on page 11-21 provides descriptions of the fields applicable to the L3VPN - PE to PE in VRF test type.

Step 3 Click **OK** to run your test after all the required fields are completed.

The Progress window appears. See the "Progress Window" section on page 11-37.

Selecting, Configuring, and Running a MPLS - PE to PE Test

This section details how to select, configure, and run a L3VPN - PE to PE (Core) test type.

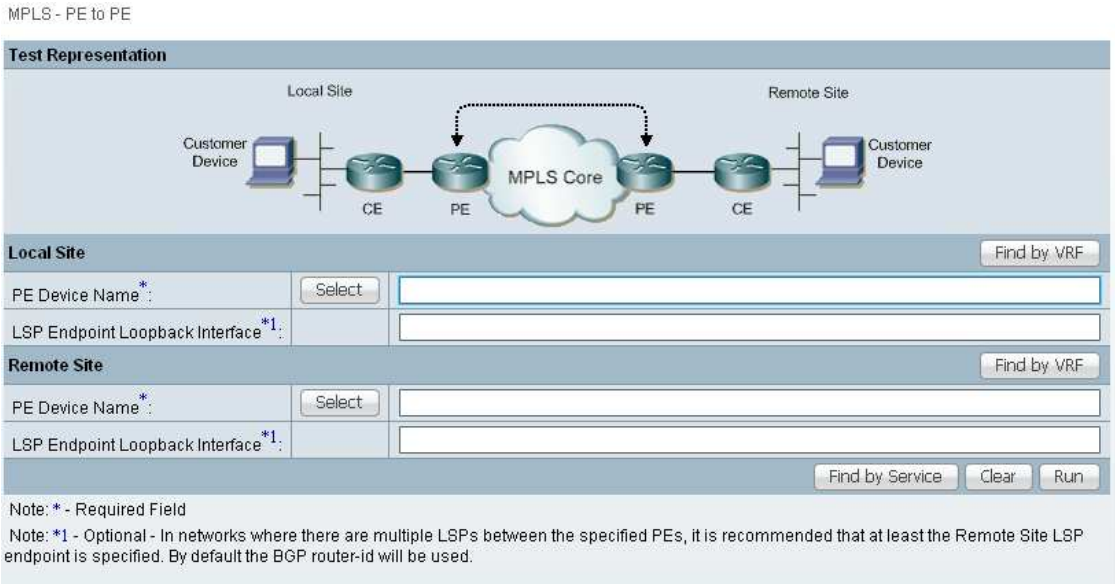
Step 1 From the Diagnostics menu, select the MPLS - PE to PE test type.

Step 2 Click on the MPLS - PE to PE connectivity verification test type.

See the "L3VPN - PE to PE Connectivity Test" section on page 11-17 for information on the PE to PE connectivity verification test type.

The MPLS - PE to PE window appears (Figure 11-16) displaying the fields corresponding to the MPLS - PE to PE test type. The MPLS - PE to PE window allows you to configure the connectivity test you would like to perform.

Figure 11-16 MPLS - PE to PE Test Type



The MPLS - PE to PE window displays the following components:

- Network diagram
- Local Site configuration area
- Remote Site configuration area

  These components and the test scope are described in further detail in the "Selecting, Configuring, and Running a L3VPN - CE to CE Test" section on page 11-19.

Step 3 Configure the fields in the MPLS - PE to PE window as required.

Table 11-3 on page 11-21 provides descriptions of the fields applicable to the L3VPN - PE to PE test type.

Step 4 Click **OK** to run your test after all the required fields are completed.

The Progress window appears. See the "Progress Window" section on page 11-37.

Configuring the LSP Endpoint Loopback IP Address for a MPLS - PE to PE Test

This section details how to configure the LSP endpoint loopback interface and IP address for the MPLS - PE to PE test type.

Remote LSP Endpoint Loopback IP Address

L3 VPN Customer traffic uses the BGP next hop address of the customer route to select the LSP. When testing the core, an MPLS OAM ping and trace is performed from the local PE to the remote PE. To ensure that Diagnostics tests the same LSP as your traffic traverses, the IP prefix Diagnostics tests to is the BGP next hop address of the customer route.

Diagnostics does not have customer route information for the PE to PE core test type. Diagnostics therefore has no way to determine the BGP next hop. By default, Diagnostics chooses the ping and trace destination, not based on the next hop, but on the BGP router ID on the remote PE. In some network configurations, such as those with multiple cores, or with multiple loopback addresses used for control and data plane traffic, this BGP router ID might not match the next hop used by the customer traffic and the incorrect (or no) LSP is tested.

Local LSP Endpoint Loopback IP Address

The MPLS - PE to PE test type allows you to perform the test in the reverse direction when running the test in the forward direction fails to find the problem. Configuring the local LSP endpoint loopback IP address ensures that the test selects the correct LSP when the test is run in the reverse direction.

When Should I Specify the LSP Endpoint Loopback IP Address?

Specify the LSP endpoint loopback IP address when:

- The BGP router ID is the address of a loopback that has no LSP assigned to it.
- The BGP router ID is not the address of a loopback.
- Several LSPs are defined and the traffic is using a different LSP than the router ID provides.
- Several LSPs are defined and the traffic switches LSP based on a routemap.

Note You must provide the correct BGP next hop when specifying the remote LSP endpoint.

Figure 11-17 displays an example network topology that illustrates the LSP Endpoint Loopback IP Address field usage. This example network topology has three logical MPLS cores and some of the PE BGP router-ids are not associated with a loopback interface. In addition, two of the CEs are dual homed to different cores.

Figure 11-17 Example Network Topology

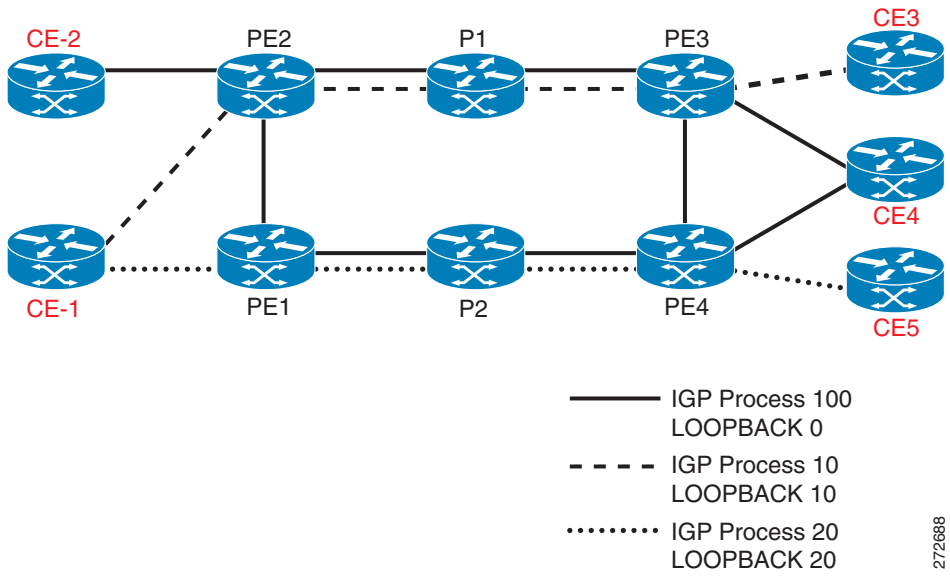


Table 11-8 provides IP addressing information relating to the example network topology displayed in Figure 11-17.

Table 11-8 IP Addressing

PE	BGP Router ID	Loopback 0	Loopback 10	Loopback 20
PE2	1.1.1.1	1.1.1.1	N/A	20.20.20.1
PE3	1.1.1.3	1.1.1.3	N/A	20.20.20.3
PE1	50.50.50.1	1.1.1.6	10. 10.10.1	N/A
PE4	50.50.50.3	1.1.1.8	10. 10. 10.3	N/A

Table 11-9 specifies the IP addresses that can be used as the remote LSP Endpoint IP Address to test each LSP.

Table 11-9 Inputs Required to Test Each LSP

LSP Under Test	For CE	Remote Site PE	Remote Endpoint
Solid line	CE-2	PE2	Not required as next hop is the BGP router-id.
Solid line	CE-4	PE4	1.1.1.8 (Loopback 0)
Solid line	CE-4	PE3	Not required as next hop is the BGP router-id.
Dotted line	CE-1	PE2	20.20.20.1 (Loopback 20)
Dotted line	CE-3	PE3	20.20.20.3 (Loopback 20)
Dashed line	CE-1	PE1	10. 10.10.1 (Loopback 10)
Dashed line	CE-5	PE4	10. 10.10.3 (Loopback 10)

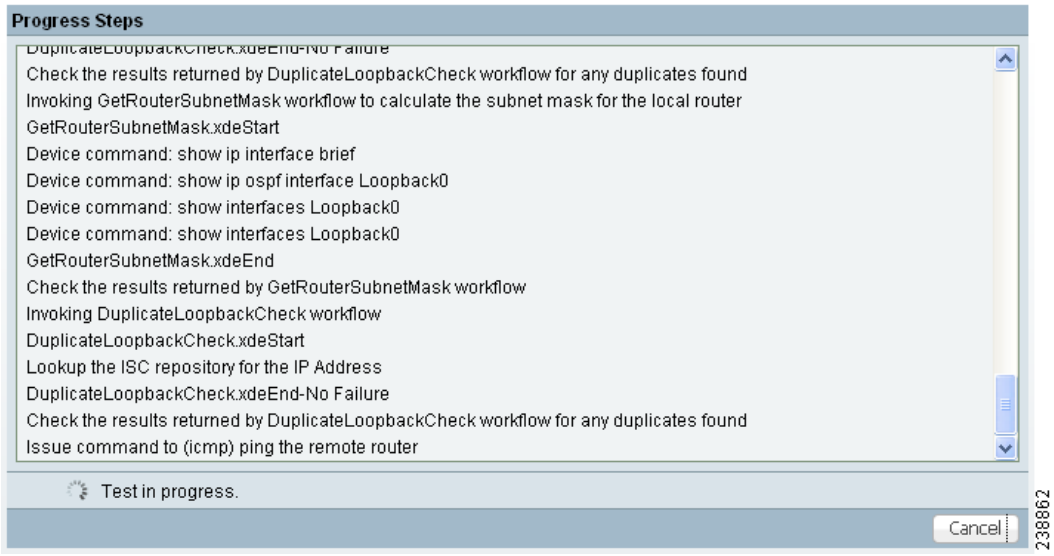
Progress Window

The Progress window appears (see Figure 11-18) while the test is being performed.

Note The time taken to perform an MPLS VPN Connectivity Verification test varies. A test could take some time to complete, depending on the size of your network, the test type selected, whether a connectivity problem is identified, and the nature of this connectivity problem.

The Progress window displays a one-line textual summary of each step that has been completed and the step that is currently executing.

Figure 11-18 Progress Window



Click the Cancel button to cancel the test if required. If you click Cancel, you are asked to confirm that you want to cancel the test. If you confirm, the test is cancelled when the current step has completed. If the current step involves device interaction, this completes before the test is cancelled. Upon cancellation, the Test Results window appears indicating that you cancelled the test. All completed steps are displayed in the test log.

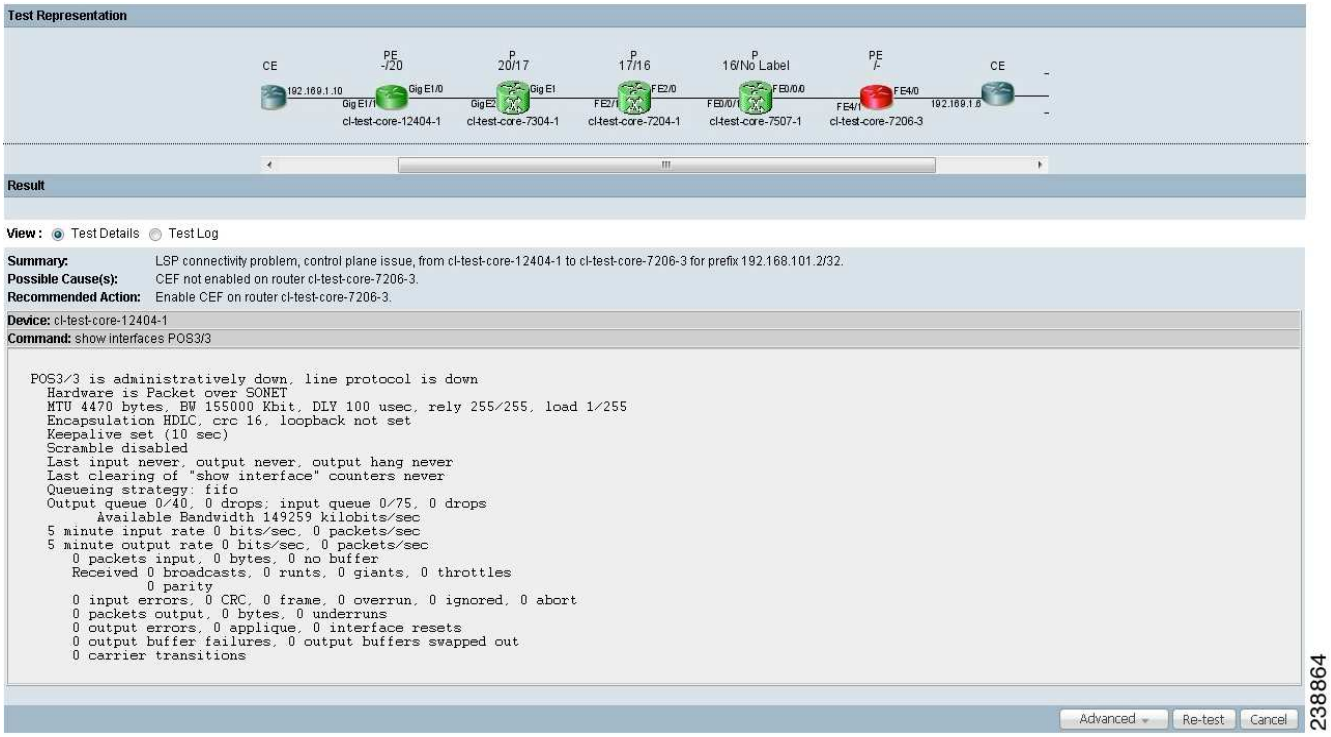
When the test is complete, the Test Results window appears. See the "Interpreting the Test Results" section on page 11-37, for further details.

Interpreting the Test Results

This section describes how to interpret your test results. This section contains the following information:

Upon completion of a MPLS VPN Connectivity Verification test, the Test Results window appears (see Figure 11-19).

Figure 11-19 Test Results Window with Failure Specific Additional Information Displayed



The Test Result window displays the location and cause of the problem found, recommended actions, observations, and details of the automated troubleshooting and diagnostics steps performed. The Test Result window also allows you to invoke advanced troubleshooting options where appropriate (see Table 11-10).

The Test Results window consists of the following components:

Table 11-10 Field Descriptions for the Test Results Window

Field/Button	Description
Data path	See the "Data Path" section on page 11-39
Test Details	See the "Test Details" section on page 11-41
Test Log	See the "Test Log" section on page 11-42
Export button	The Export button appears when the Test Log radio button is selected. See the "Export" section on page 11-43.
Advanced button	Click the Advanced button to launch advanced troubleshooting. See the "Advanced Troubleshooting Options" section on page 11-43. The options available on this button are dynamically configured depending on the test result and the test type.

Table 11-10 Field Descriptions for the Test Results Window (continued)

Field/Button	Description
Re-test button	Click the Re-test button to rerun the connectivity test using the existing configuration. This can be used to verify the fix implemented.
Cancel button	Click the Cancel button to cancel the current test and return to the Test Configuration window. You will not be asked to confirm the cancellation.

If multiple failures exist in the tested path, the failure reported is determined by the order in which Diagnostics performs troubleshooting. For the CE to CE connectivity test type, Diagnostics troubleshooting is performed in the following order:

1. Access circuit (local and remote).
2. MPLS Traffic Engineered (TE) tunnels.
3. MPLS core.
4. MPLS VPN edge.

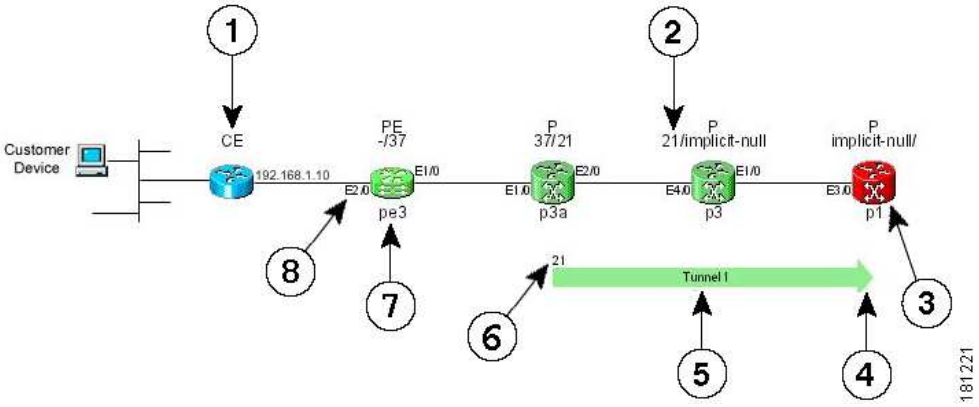
The other test types troubleshoot in the same order, but do not perform all of the steps.

Note The Test Result window displays details of the first failure found. If multiple failures exist, subsequent failures are not reported until the current failure is fixed and the test is rerun.

Data Path

The Data Path (see Figure 11-20) shows a graphical representation of the path between the two sites that have been tested. If a failure is found on an MPLS Traffic Engineered tunnel, the tunnel is displayed in the Data Path. Any non-overlapping P-P, PE-P, or P-PE MPLS TE Tunnels found in the path before the point of failure will also be displayed in the datapath.

Figure 11-20 Data Path



1. Device Role (CE, PE, or P).
2. MPLS labels (ingress/egress).
3. Failed device.

4. Tunnel direction arrow.
5. Tunnel name.
6. Tunnel label.
7. Device hostname.
8. Interface name.

Where present, MPLS TE tunnels are displayed below the device path.

If a Customer Device IP address is specified, this IP address will appear beside the text "Customer Device."

Note An MPLS TE tunnel is displayed, only when it is found to be the cause of the connectivity failure.

If a failure is found, the data path highlights the failed device or link. The device colors used in the data path are described in Table 11-11.

Table 11-11 Data Path Device Color Codes

Color	Icon	Description
Green		Device has been tested and is functioning normally.
Blue		Device has not been tested or status is unknown.
Red		Device failure.
Yellow		Possible device failure.
Grey		Device access failure.

The link color used in the data path is described in Table 11-12.

Table 11-12 Data Path Link Color Code

Color	Icon	Description
Red		A connectivity failure has been found. This failure might be due to a problem on one or both attached devices.

For each core PE and P device, the following information is displayed:

- Role (PE or P)
- Device name
- Interface names
- Ingress and egress MPLS labels (MPLS core failures only)

The information displayed for CE devices and customer devices is minimal. Typically only the information provided during test configuration is displayed for these devices.

The following information is displayed for an MPLS Traffic Engineered tunnel:

- Tunnel name
- Tunnel direction (direction arrow)
- Tunnel label

Note It is not possible to Telnet to a device from the Data Path in the Test Result window.

Test Details

The Test Details section of the Test Results window (see Figure 11-19 on page 11-38) displays a summary of the automated troubleshooting and diagnostics results, observations made during troubleshooting, additional failure-specific information, and recommended action. See Failure Scenarios, page 11-57 for details of failures and observations reported by Diagnostics, and for a list of all IOS and IOS XR commands executed by Diagnostics as part of the troubleshooting.

The Test Details summary is displayed in all cases. The test details summary consists of three fields that detail:

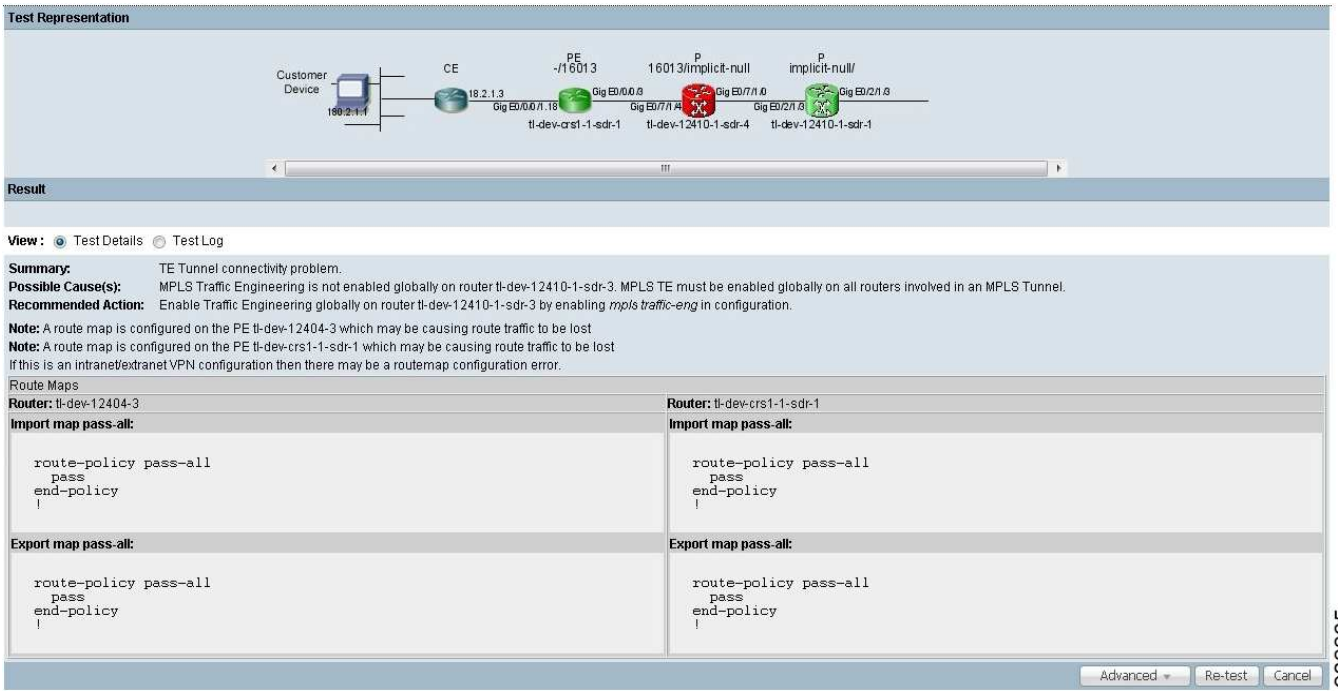
- Summary—Displays a brief summary of the failure found.
- Possible Cause(s)—Possible causes of the failure.
- Recommended Action—Recommended actions to resolve the problem.

Failure-specific additional information is displayed below the summary as required. When displayed, this provides additional information on the problem found. For example, Forwarding Information Base (FIB), Label Forwarding Information Base (LFIB), Border Gateway Protocol (BGP) table entries, and route target import/exports. This additional failure specific information helps highlight problems such as FIB, LFIB, BGP inconsistencies, and route target import/export mismatches. For some failures no additional information is displayed.

Figure 11-19 on page 11-38 shows an example Test Results window with failure specific information below the Test Details summary. The Test Details radio button is selected by default.

Observations made during troubleshooting are displayed as notes below the Test Details summary. Observation notes detail observations made during troubleshooting which could be related to the failure. They should be considered as additional troubleshooting information. Figure 11-21 shows an example Test Results window with two observation notes. In some cases no observation notes are displayed, while in other cases multiple notes might be displayed.

Figure 11-21 Test Results Window with Observation Notes

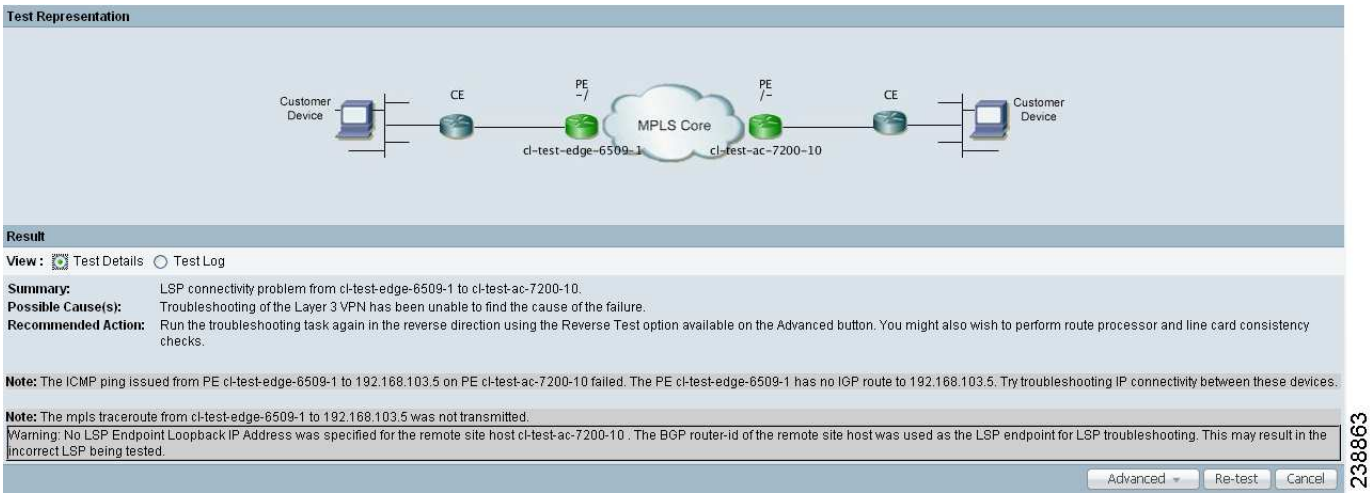


238865

Test Log

Click the Test Log (see Figure 11-22) radio button to display details of all troubleshooting and diagnostics steps in the order in which they were performed.

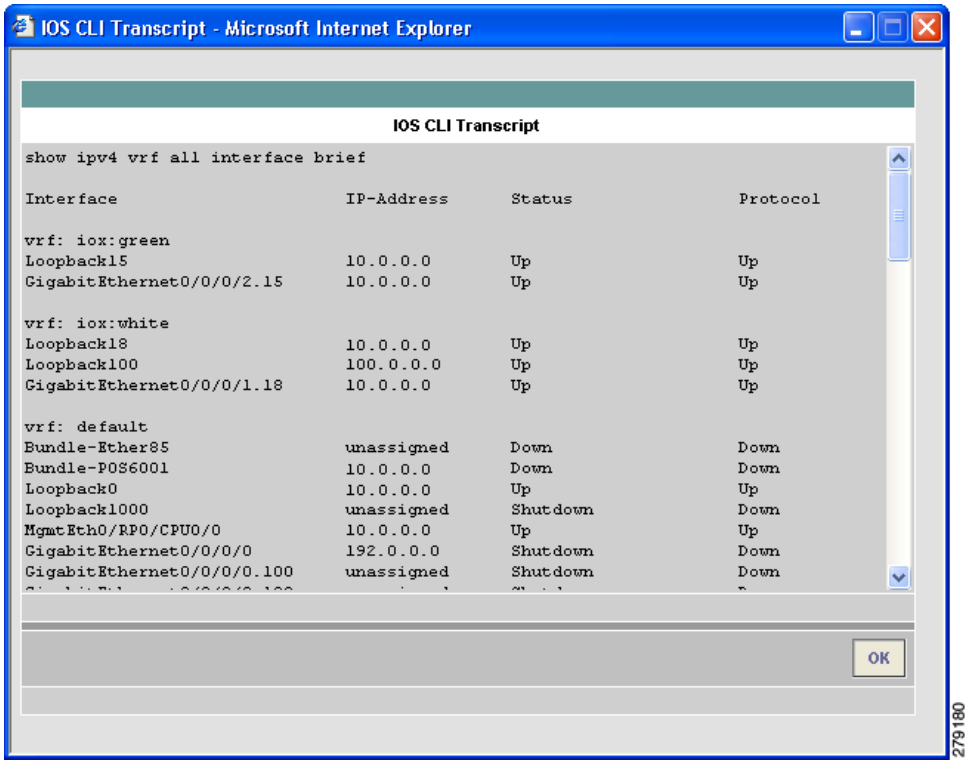
Figure 11-22 Test Results Window—Test Log



238863

Some steps require device interaction involving the execution of IOS or IOS XR CLI commands. These steps appear in the Test Log as hyperlinks. Clicking a hyperlink opens a pop-up window that displays the IOS or IOS XR CLI transcript for the step (see Figure 11-23). This transcript includes the IOS or IOS XR commands run and all resulting output.

Figure 11-23 IOS CLI Transcript Window



Export

You might want to export the test log to include it in a trouble ticket, problem escalation, or when contacting Cisco TAC. The test log can be exported to file through the Export button located at the bottom of the Test Log (see Figure 11-22 on page 11-42). All steps displayed in the test log, including IOS and IOS XR CLI transcripts, are exported in text format.

Step 1 Click the **Export** button.

The standard browser file download window appears with a default filename of *export.rtf*.

Step 2 Save the file.

Advanced Troubleshooting Options

This section describes advanced troubleshooting options, as follows:

- Reverse Path Testing, page 11-44

- LSP Visualization, page 11-44

Advanced troubleshooting provides further options that you can use to troubleshoot your network.

The advanced troubleshooting options supported are detailed in Table 11-13.

Table 11-13 Advanced Troubleshooting Options

Advanced Troubleshooting Option	Description
Reverse path test	Available when a failure is found.
LSP Visualization	Available when no failure is found.
LSP Troubleshooting	Available when an IP failure is found.

The appropriate advanced troubleshooting options are made available through the Advanced drop-down button at the bottom of the Test Results window.

Reverse Path Testing

Note The reverse path testing option is available for all test types except for the PE to attached CE test type.

In some cases, the MPLS VPN Connectivity Verification test detects a connectivity failure but is unable to identify the cause of this failure. By repeating the test in the reverse direction (that is, reversing the local and remote site configuration), it might be possible to identify the cause of the problem. In other cases, repeating the test in the reverse direction can result in a more precise diagnosis of the problem found. For example, while performing a connectivity test in the forward direction, an LSP connectivity problem might be identified on a device. However, this problem could be caused by an LDP misconfiguration on the downstream LSP neighbor. By repeating the test in the reverse direction, the misconfigured downstream router is encountered first and the LDP misconfiguration is diagnosed. When this situation occurs, the Test Details displayed in the Test Results window advises you to perform the test in the reverse direction. The Reverse Test option is available on the Advanced drop-down button in the Test Results window.

Selecting the Reverse Test advanced troubleshooting option invokes the MPLS VPN Connectivity Verification test in the reverse direction. No further configuration is required.

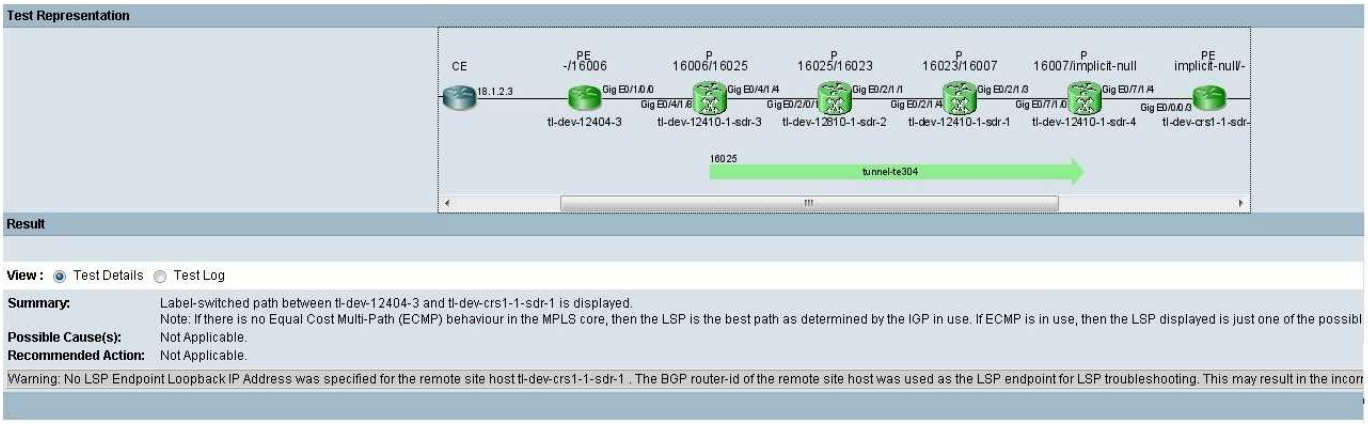
The results of the reverse path testing are displayed in the Test Results window.

LSP Visualization

Note LSP visualization is available for all test types except for the PE to attached CE test type.

When no failure is found, the Test Results window data path displays a summary of the test performed. This does not show details of the path through the core that has been tested. LSP Visualization displays a hop-by-hop Data Path illustration of the MPLS label switched path (LSP) between the local and remote sites (see Figure 11-24). The LSP Visualization displays all intermediate non-overlapping PE to P, P to P and P to PE tunnels found in the forward path. The path shown is the path tested during the MPLS VPN Connectivity Verification test.

Figure 11-24 Test Results Window—LSP Visualization



The Data Path displays the following for each PE and P device in the tested path:

- Role (PE or P)
- Device name
- Interface name
- Ingress and egress labels

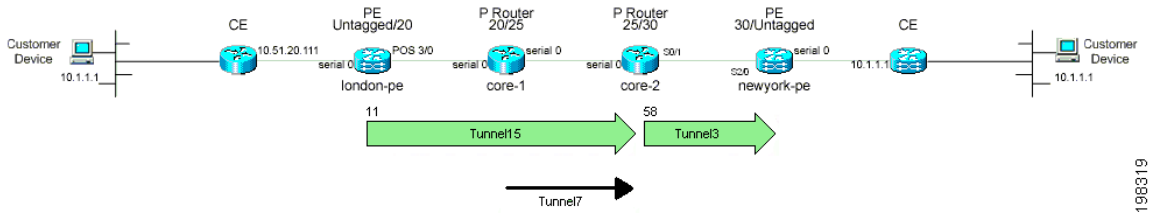
The Data Path displays the following for each PE to PE MPLS Traffic Engineered tunnel:

- Tunnel name
- Tunnel direction (direction arrow)
- Tunnel label
- Tunnel Type

Note In cases where there are multiple MPLS TE Tunnels configured, the only tunnel that is displayed in the datapath will be the one that is actually carrying the traffic.

In the example below, Tunnel 7 is not displayed because it is overlapping, i.e. its headend is configured at the midpoint of Tunnel 15 which is configured on the upstream router.

Figure 11-25 Multiple MPLS TE Tunnel Configuration



For more details of what is displayed in the Data Path, see the "Data Path" section on page 11-39.

LSP Visualization is only offered when an MPLS VPN Connectivity Verification test does not detect a connectivity problem.

Note When using an MPLS VPN Connectivity Verification test for post-provisioning verification, LSP Visualization provides an additional level of verification by displaying the LSP path taken across the MPLS core.

Switching Tunnel Checking Off—For Networks with Non-Cisco P Routers

During tunnel diagnostics, Diagnostics might be required to visit every device to determine if a tunnel is present at that point. Since Diagnostics does not log in to non-Cisco devices, this can result in a misdiagnosis of a fault occurring at the non-Cisco device (even thought it might not be the actual source of the fault) as the troubleshooting workflow is unable to proceed. As a result, it is useful to disable tunnel diagnostics for networks that contain non-Cisco devices.

Tunnel diagnostics is enabled as default. The default value can be changed by an Admin user, within the the Prime Fulfillment Control Center (**Administration tab > Control Center > Hosts**). Tunnel diagnostics can be enabled or disabled within the Command Flow Runner (cfr) component (parameter disableTunnelDiagnostics). When the appropriate disableTunnelDiagnostics parameter is set to true, Diagnostics does not perform tunnel diagnostics.

The Test Results window displays an observation message stating that Diagnostics tunnel diagnostics are disabled. The error message indicating a device is not in the inventory mentions that a possible cause is a non-Cisco device on the path, and that the error might be on this device or a near neighbor.

How Does Diagnostics Work?

This chapter describes how the Diagnostics application works.

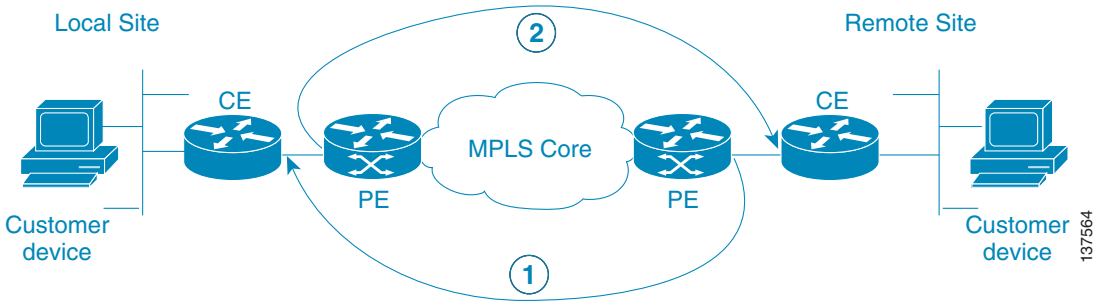
The MPLS VPN Connectivity Verification test consists of connectivity testing, troubleshooting, and diagnostics steps. The exact steps performed for each test depend upon the nature of the failure found and the location of the failure within the network. Due to the simple test configuration and result presentation, you have little need to understand the troubleshooting and diagnostics logic. However, in some cases - particularly when examining the test log - you might want an understanding of the troubleshooting and diagnostics process. This chapter provides a high-level overview of the connectivity testing, troubleshooting, and diagnostics logic.

Note The steps detailed in this chapter are illustrative of the types of tests that Diagnostics performs. However, this list of tests is not exhaustive; Diagnostics performs many more tests.

The test scope is determined by the test configuration you enter. For example, for each site, testing could be performed to a customer device within the site or to the CE access circuit interface. For simplicity, this chapter assumes that testing for all sites is to the CE access circuit interface.

The first step tests VPN connectivity between the two sites to determine if a problem exists. This is achieved using the Cisco IOS VRF ping functionality. Ideally, this test should be initiated from a device in the local site subnet to a destination IP address in the remote site subnet. However, Prime Fulfillment supports managed and unmanaged Cisco CE devices, and non-Cisco CE devices. The troubleshooting and diagnostics functionality works for all cases. As a result, it is only possible to initiate tests from PE and P devices within your core network. To work around this limitation, it is necessary to perform the connectivity test in two stages (see Figure 11-26).

Figure 11-26 IOS VRF Ping Connectivity Tests



1. The first stage (see Figure 11-26) tests connectivity from the remote site PE to the local site CE. This is achieved using a Cisco IOS **ping vrf** command, specifying the local site CE access circuit interface IP address as the destination and the remote site PE access circuit interface as the source IP address.
2. The second stage (see Figure 11-26) tests connectivity from the local site PE to the remote site CE. The second stage is performed only when the **ping vrf** command in the first stage indicates successful connectivity. This is also achieved using a Cisco IOS **ping vrf** command, specifying the remote site CE access circuit interface IP address as the destination and the local site PE access circuit interface as the source IP address.

Performing the connectivity test from the remote site PE to the local site CE first ensures that any problems with the local access circuit are found first. This means that any reverse-path MPLS VPN, MPLS core, and MPLS TE Tunnel problems will be found before forward-path problems.

By testing connectivity in two stages, the troubleshooting and diagnostics functionality is able to simulate an end-to-end test from the local site CE to the remote site CE, and thus identify any VPN connectivity problems between the sites. This connectivity test exercises VPN, MPLS, and IP connectivity between the two sites.

If a VPN connectivity problem is not detected, then no troubleshooting and diagnostics are performed. If a VRF connectivity problem is detected, then a further series of connectivity tests are performed in an attempt to isolate the connectivity problem. These tests are initiated on the PE device and performed in the direction for which a VPN failure was detected. They include:

- VRF ping across core to PE access circuit interface. This determines if the failure lies on the access circuit, between the CE and PE or in the core.
- ICMP ping across core to PE loopback—This confirms that IP connectivity is working across the core.
- LSP ping across core to PE loopback—This confirms that the MPLS LSP path across the core is working.

Testing might stop at any point if the fault is isolated. A sequence of automated troubleshooting and diagnostics steps is then performed to diagnose the cause of the fault. The steps performed depend upon the nature and location of the fault. Troubleshooting is performed in the following order:

1. Access circuit (local and remote).
   a. L3 connectivity (VRF pings and trace to CE and Customer Device) and route checks.
   b. L2 (ATM, Ethernet, Frame Relay, Serial) connectivity and status checks.
   c. PE-CE routing protocol determination and status checks.
   d. PE-CE routing protocol and MP-BGP redistribution checks.

2. MPLS VPN edge.
   a. MP-BGP neighbor and VPN route checks.
   b. VRF route limit and checks.
   c. Route map presence checks.
   d. PE–PE VRF (VRF pings and trace across MPLS core) connectivity checks.
   e. PE MPLS OAM capability checks.
3. MPLS Traffic Engineered (TE) tunnels.
   a. Tunnel connectivity (TE aware ping and trace) and status checks.
4. MPLS core.
   a. IP connectivity (ICMP ping) checks.
   b. LSP connectivity (LSP ping and trace) and status checks.
   c. LSP datapath generation.
   d. LSP fault localization.
   e. LDP session and neighbor checks.
   f. Label checks.
   g. MPLS VPN edge.
   h. VPN label checks.
   i. VRF route target checks.

Note Core troubleshooting will only be performed for PE devices which support the Cisco IOS MPLS LSP Ping and Traceroute feature. For details of supported device types and Cisco IOS versions with MPLS OAM support, see Supported Hardware, IOS, and IOS XR Versions, page 11-3.

Note Diagnostics troubleshoots the primary tunnel if it is configured with FRR protection and reports the possible failures found in the primary tunnel and backup tunnels (providing FRR protection to the primary tunnel). The backup tunnel troubleshooting is limited to tunnels that are configured to protect the FRR enabled primary tunnel configured between ABRs.

After a fault diagnosis has been made, the result is displayed in the Test Results window with appropriate recommended actions to resolve the fault. The exact connectivity testing and automated troubleshooting and diagnostics steps performed can be viewed in the Test Log section of the Test Results window.

Frequently Asked Questions

Q. When I perform an MPLS VPN Connectivity Verification Test, the Progress window appears to hang and performs the same step for up to five minutes. After five minutes the Test Results window displays the following message.

```
Summary: Cannot connect or login to device router1.
Possible Cause(s): Device could be down, there could be problems with network
connectivity, or the login details in the repository might be incorrect
Recommended Action: Restore connectivity to the device before attempting the test.
```

```
If in-band network management is in use then you might want to consider performing
a Traceroute from the management station to device router1 to find where IP
connectivity fails.
```

A. The device has not responded when an attempt has been made to log on to it. Ensure that the device is not down. Ensure that you have IP connectivity from the Prime Fulfillment server to the device. Ensure that the device login details configured in the Prime Fulfillment Repository match those configured on the physical device. Ensure that all available VTY sessions on the device are not in use.

Q. When I perform an MPLS VPN Connectivity Verification Test, sometimes the devices I configured as the local site are displayed on the left-hand side of the Data Path, in the Test Results window. In other instances, these local site devices are displayed at the right-hand side of the Data Path, in the Test Results window. Why is this?

A. Connectivity problems in an MPLS VPN can often only be detected in a particular direction. The MPLS VPN Connectivity Verification Test tests in both directions (from local site to remote site and vice-versa). Depending on the direction of test when the problem is found, the local site devices might be displayed on either the left-hand side, or right-hand side of the Data Path in the Test Results window.

Q. When I perform two or more MPLS VPN Connectivity Verification tests in parallel on the same client machine, the test results for one of these tests is displayed in the Result Screens for all tests. The test results for the other tests are lost. How can I avoid this?

A. When performing parallel MPLS VPN Connectivity Verification tests on the same client machine, you must ensure each test is performed using a different HTTP session. To do so, run each test in a separate browser launched from the command line or by clicking on the browser icon on the desktop or Start menu. Do not run parallel tests in tabs within the same browser window or in browser windows launched from existing browser windows.

VPN Topologies

This appendix details how to perform an MPLS VPN Connectivity Verification test for the supported VPN topologies. This appendix contains the following sections:

Testing with Full Mesh VPN Topology

By default, an MPLS VPN Connectivity Verification test assumes that the local and remote sites are connected through a full mesh VPN topology and that these sites can communicate directly. For details of how to configure an MPLS VPN Connectivity Verification test for a full mesh VPN topology, see Performing an MPLS VPN Connectivity Verification Test, page 11-18.

Testing with Hub and Spoke VPN Topology

Customer sites connected through a hub and spoke VPN, cannot communicate directly. The customer sites (Spokes) communicate through a Hub router. When testing connectivity between two sites connected through a hub and spoke VPN you should perform the test using the following steps:

Step 1 MPLS VPN Connectivity Verification test between the local and the remote sites.

Step 2 MPLS VPN Connectivity Verification test between the local site and the hub CE interface that is attached to the hub PE interface importing routes.

Step 3 MPLS VPN Connectivity Verification test between the remote site and the hub CE interface that is attached to the hub PE interface importing routes.

Step 4 MPLS VPN Connectivity Verification test between the local site and the hub CE interface that is attached to the hub PE interface exporting routes.

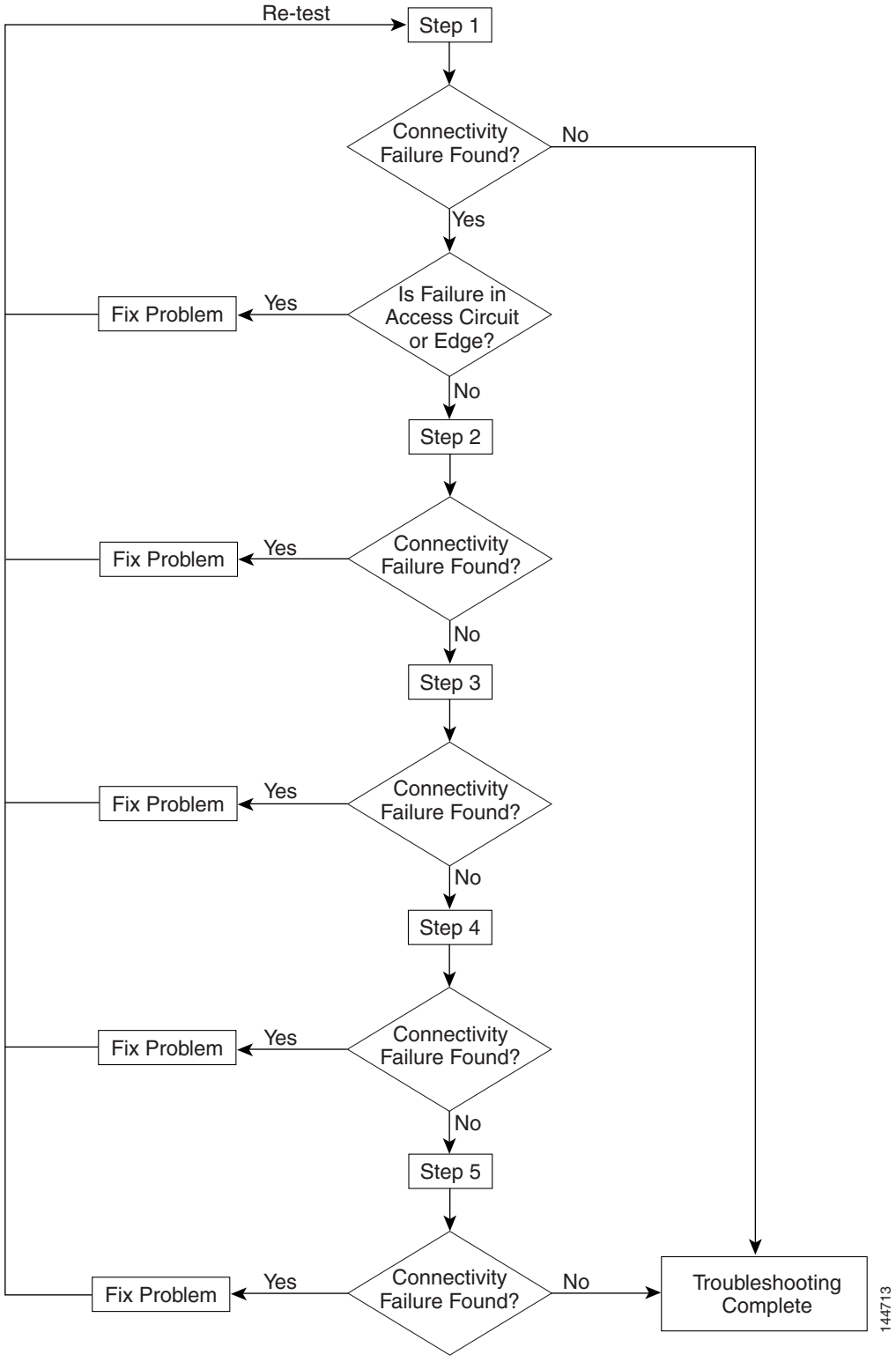
Step 5 MPLS VPN Connectivity Verification test between the remote site and the hub CE interface that is attached to the hub PE interface exporting routes.

Each step involves performing an MPLS VPN Connectivity Verification test between different points. Depending on whether a connectivity failure exists and the location of this failure, it might not be necessary to perform all five steps. Figure 11-27 shows the workflow for testing a hub and spoke VPN.

After fixing a problem reported in Step 1 through Step 5, you should repeat Step 1 to verify that connectivity between the sites has been restored.

Note If a connectivity failure is detected in Step 1 due to an access circuit or VPN edge problem, then the problem will be correctly diagnosed by the MPLS VPN Connectivity Verification test performed in Step 1. You should rectify the problem as described by the text results. If the connectivity failure is due to a problem within the core of the hub and spoke MPLS VPN, then the result reported by Step 1 might be incorrect and should be ignored. Step 2 through Step 5 should be performed until the problem is diagnosed correctly.

Figure 11-27 Workflow

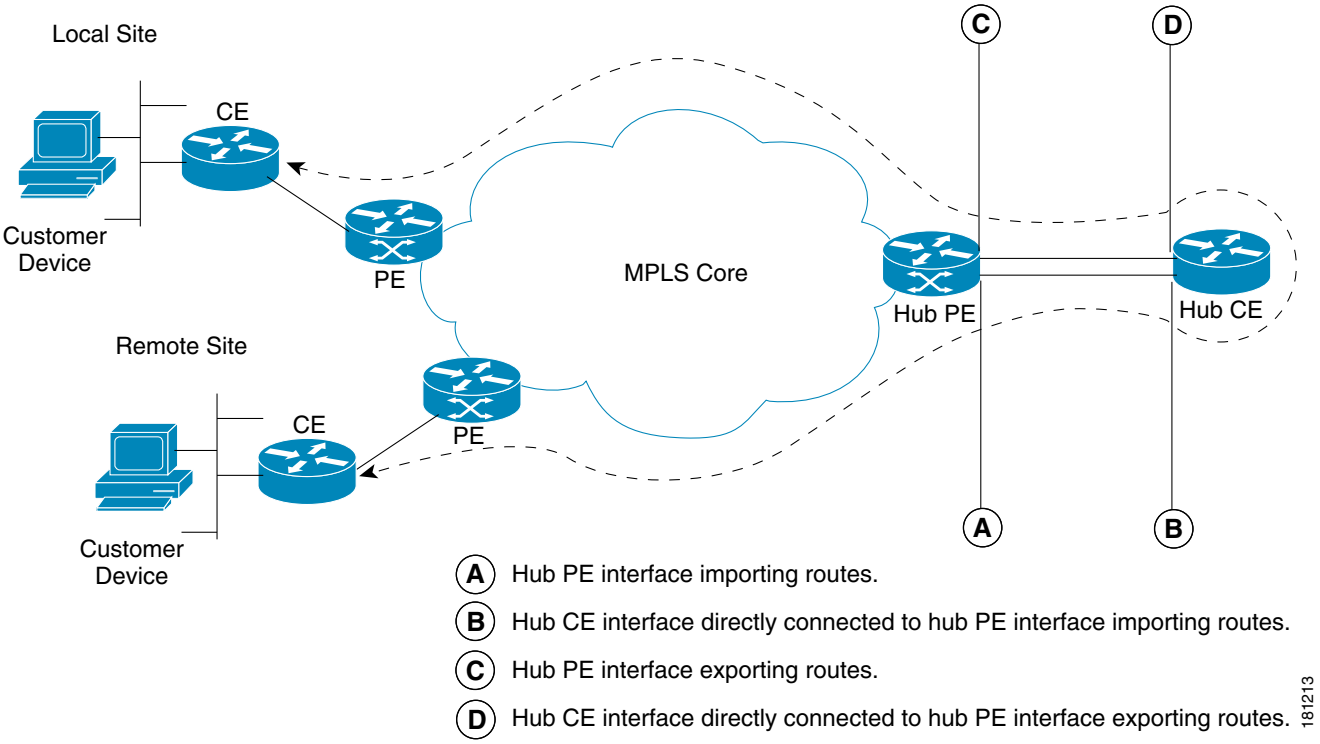


1. You should perform an MPLS VPN Connectivity Verification test between the local and remote sites. If this test finds no connectivity problems, then no further troubleshooting is required. If this test reports a connectivity failure caused by an MPLS problem, you should ignore the test result and move to 2.. As an MPLS VPN Connectivity Verification test assumes a full mesh VPN topology, the problem reported will be incorrect. You must perform further MPLS VPN Connectivity Verification tests to identify the problem on a hub and spoke VPN. If this test reports a connectivity failure caused by a non-MPLS problem (for example, access circuit or VPN edge failure), then you should fix the problem as reported and retest.

Note If a connectivity failure is found in the core, the MPLS VPN Connectivity Verification test performed in 1. might detect that a hub and spoke VPN topology is being tested and advise you to perform hub and spoke specific troubleshooting as described in the following steps. The MPLS VPN Connectivity Verification test detects a hub and spoke VPN topology by checking the Route Target imports and exports. If the same Route Target is imported and exported by one or both PE routers, then a hub and spoke VPN is assumed.

Figure 11-28 illustrates the MPLS VPN Connectivity Verification tests required to test connectivity between two sites in a hub and spoke VPN.

Figure 11-28 Testing a Hub and Spoke VPN Topology—Step 1

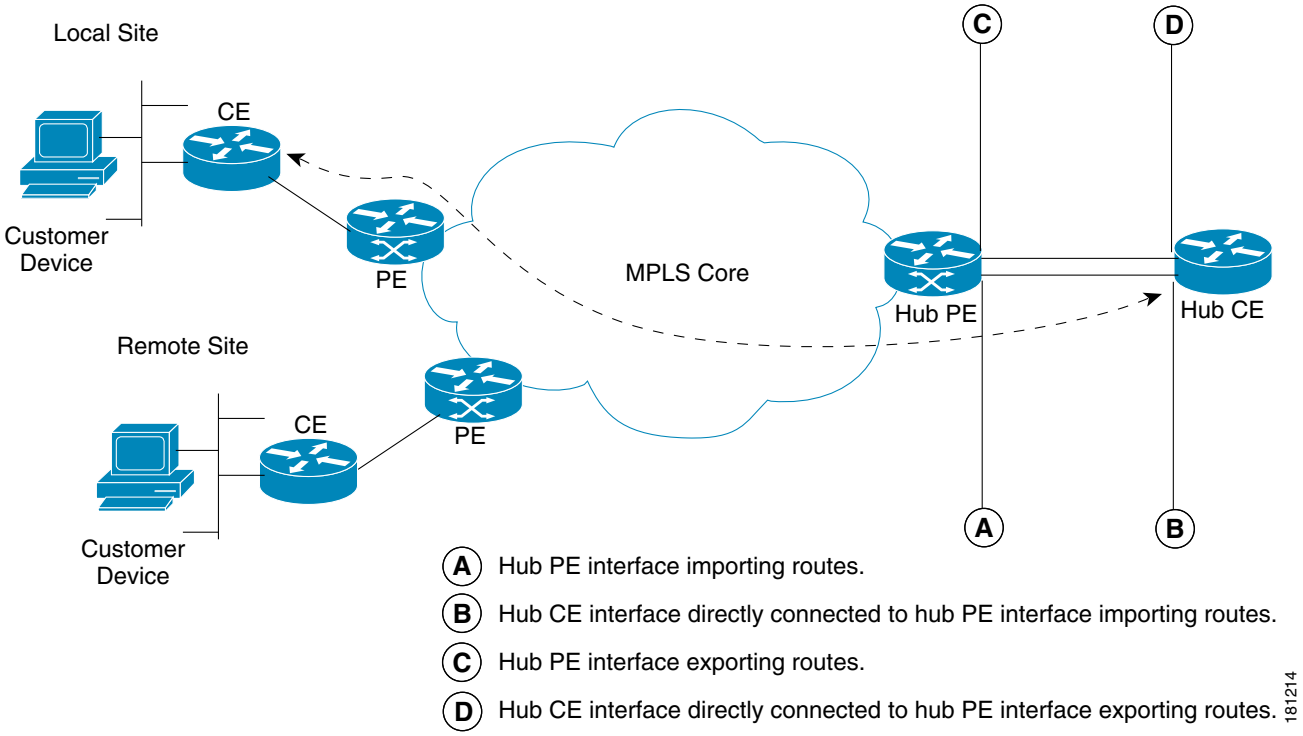


2. You should perform an MPLS VPN Connectivity Verification test between the local site (Spoke) and the hub CE interface that is attached to the hub PE interface which imports routes (shown in Figure 11-29 as *B*). When configuring the MPLS VPN Connectivity Verification test, the Local Site fields on the MPLS VPN Connectivity Verification Configuration window should be configured with details of the local customer site. The Remote Site fields should be configured with details of the Hub PE/CE interfaces that import routes (shown in Figure 11-29 as *A* and *B*), as shown in Table 11-14.

Table 11-14 Test Configuration—Hub Route Import Interface Tests

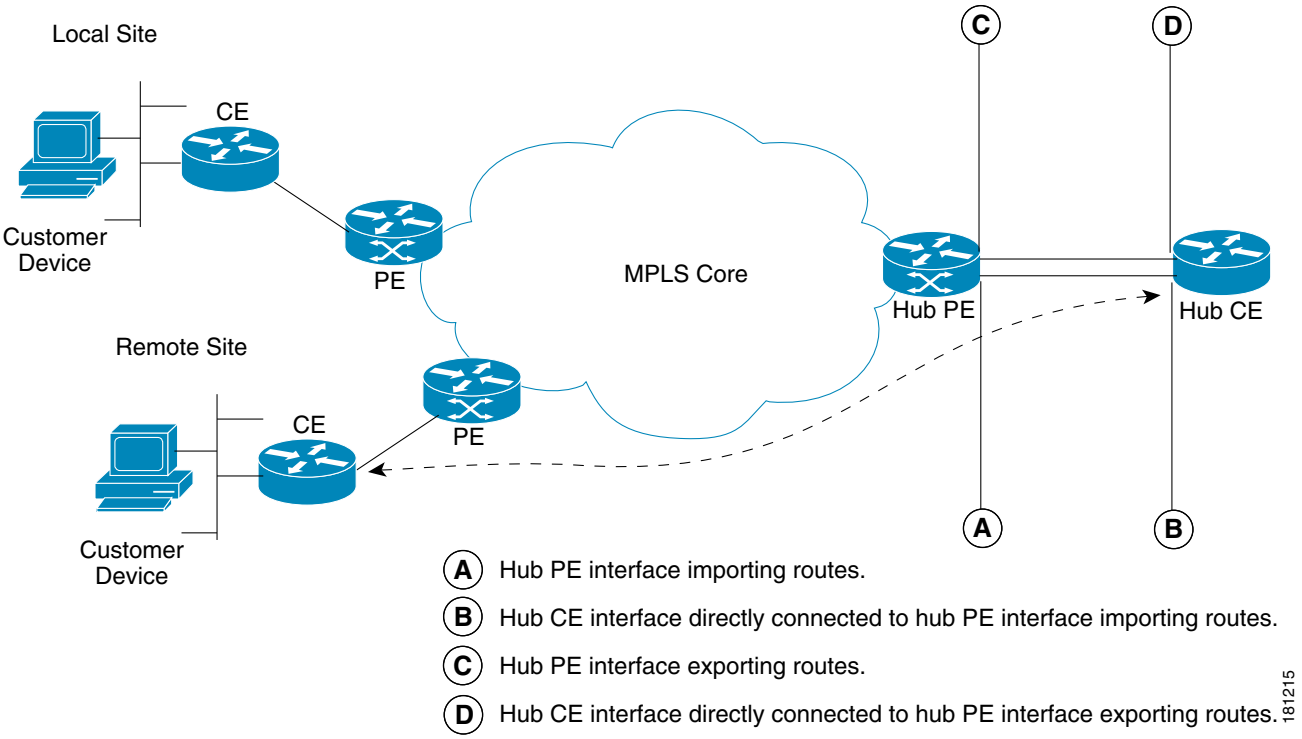
Field Name	Hub Detail
PE Device Name	Hub PE device name.
PE Access Circuit Interface	Hub PE interface which imports routes.
CE Access Circuit Interface IP Address	IP address of hub CE interface directly connected to PE interface which imports routes.
Customer Device IP Address	Leave blank.

Figure 11-29 Testing a Hub and Spoke VPN Topology—Step 2



3. You should perform an MPLS VPN Connectivity Verification test between the remote site (Spoke) and the hub CE interface that is attached to the hub PE interface which imports routes (shown in Figure 11-30 as *B*). When configuring the MPLS VPN Connectivity Verification test, the Local Site fields should be configured with details of the Hub PE/CE interfaces that import routes (shown in Figure 11-30 as *A* and *B*), as shown in Table 11-14. The Remote Site fields on the MPLS VPN Connectivity Verification Configuration window should be configured with details of the remote customer site.

Figure 11-30 Testing a Hub and Spoke VPN Topology—Step 3

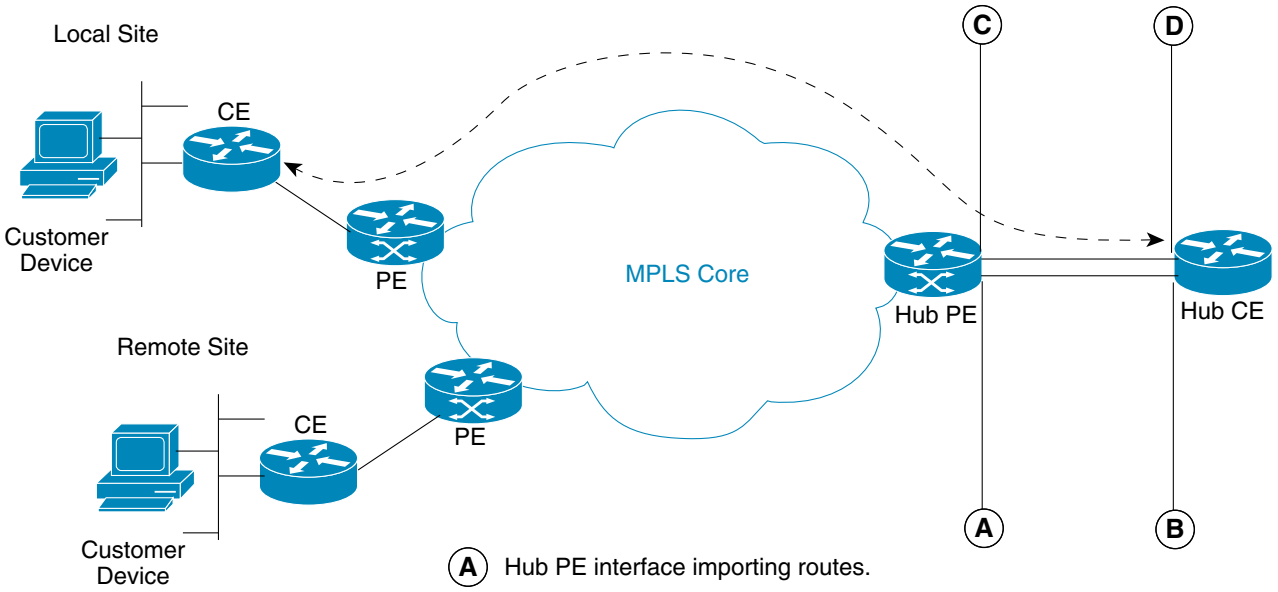


4. You should perform an MPLS VPN Connectivity Verification test between the local site (Spoke) and the hub CE interface that is attached to the hub PE interface which exports routes (shown in Figure 11-31 as *D*). When configuring the MPLS VPN Connectivity Verification test, the Local Site fields on the MPLS VPN Connectivity Verification Configuration window should be configured with details of the local customer site. The Remote Site fields should be configured with details of the Hub PE/CE interfaces that export routes (shown in Figure 11-31 as *C* and *D*), as shown in Table 11-15.

Table 11-15 Test Configuration —Hub Route Export Interface Tests

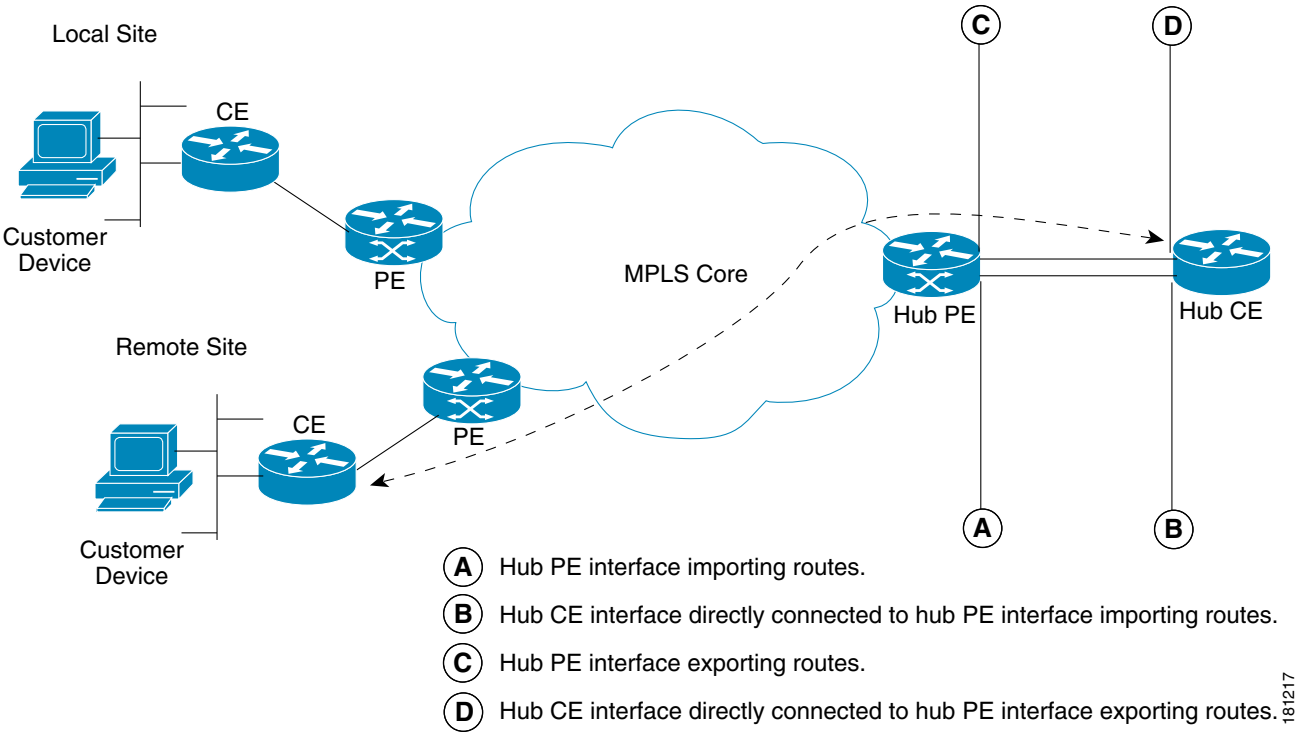
Field Name	Hub Detail
PE Device Name	Hub PE device name.
PE Access Circuit Interface	Hub PE interface which exports routes.
CE Access Circuit Interface IP Address	IP address of hub CE interface directly connected to PE interface which exports routes.
Customer Device IP Address	Leave blank.

Figure 11-31 Testing a Hub and Spoke VPN Topology—Step 4



5. You should perform an MPLS VPN Connectivity Verification test between the remote site (Spoke) and the hub CE interface that is attached to the hub PE interface which exports routes (shown in Figure 11-32 as *D*). When configuring the MPLS VPN Connectivity Verification test, the Local Site fields should be configured with details of the Hub PE/CE interfaces that export routes (shown in Figure 11-32 as *C* and *D*), as shown in Table 11-15. The Remote Site fields on the MPLS VPN Connectivity Verification Configuration window should be configured with details of the remote customer site.

Figure 11-32 Testing a Hub and Spoke VPN Topology—Step 5



Testing with Intranet/Extranet VPN Topology

Sites connected through an Intranet/Extranet VPN topology can communicate directly, similar to a full mesh VPN topology. When configuring an MPLS VPN Connectivity Verification test between two sites connected through an Intranet/Extranet VPN, you should configure the test as normal.

When testing connectivity between sites connected through an Intranet/Extranet VPN, Diagnostics will troubleshoot MPLS VPN connectivity issues including access circuit, VPN edge, and MPLS core problems. Diagnostics does not troubleshoot Intranet/Extranet VPN specific problems, such as missing or miss-configured route maps.

If an MPLS VPN Connectivity Verification test detects a connectivity failure but that failure cannot be attributed to MPLS VPN connectivity issues, including access circuit, VPN edge, and MPLS core problems, then the Test Results window recommends you troubleshoot the Intranet/Extranet configuration.

Note Diagnostics assumes a possible Intranet/Extranet VPN topology if it finds Route Maps configured on either PE.

Testing with Central Services VPN Topology

With a Central Services VPN topology the client sites can communicate directly with one or more central sites, but they cannot communicate with each other. When configuring an MPLS VPN Connectivity Verification test between a client site and central site, connected through a Central Services VPN topology, you should configure the test as normal by entering the client site and central site, as the local and remote site respectively.

It is not possible to perform an MPLS VPN Connectivity Verification test between two client sites in a Central Services VPN.

Failure Scenarios

This chapter provides details of all failure scenarios reported by the Diagnostics application. It also details IOS XR support caveats.

For more information, e-mail: mpls-diagnostics-expert@cisco.com

Note Diagnostics only supports L3 services implemented on sub-interfaces/interfaces.

Failure Scenarios

This section lists the failure scenarios reported by Diagnostics, as follows:

- Access Circuit, page 11-57
- MPLS Edge, page 11-68
- MPLS Core, page 11-74
- Customer Site, page 11-83

Each failure scenario provides a table that lists whether the failure scenario is supported by each of the five Diagnostics test types. This table details whether the failure scenario is supported on IOS and IOS XR. The table also details if the failure scenario is supported for IPv4 and IPv6.

Note In the following tables, NA stands for Not Applicable and NS stands for Not Supported.

Access Circuit

Access Circuit Blocking IP Connectivity

There is a blocking access list preventing IP connectivity from an access circuit interface on the provider (PE) router to the destination.

CE to CE	PE to Attached CE	CE to PE Across Core	PE to PE (in VRF)	PE to PE Core	IOS	IOS XR	IPv4	IPv6
Yes	Yes	Yes	NA	NA	Yes	Yes	Yes	Yes

An Invalid PE Interface has been Specified

Interface does not exist on the PE router.

CE to CE	PE to Attached CE	CE to PE Across Core	PE to PE (in VRF)	PE to PE Core	IOS	IOS XR	IPv4	IPv6
Yes	Yes	Yes	Yes	NA	Yes	Yes	Yes	Yes

ATM Interface Has No VPI/VCI

An asynchronous transfer mode (ATM) access circuit interface on a PE router has no virtual path identifier (VPI), or virtual channel identifier (VCI) assigned to it, or no VPI/VCI maps to the destination IP address.

CE to CE	PE to Attached CE	CE to PE Across Core	PE to PE (in VRF)	PE to PE Core	IOS	IOS XR	IPv4	IPv6
Yes	Yes	Yes	NA	NA	Yes	Yes	Yes	NS

ATM Interface Is Protocol Down

An ATM access circuit interface on a PE router is protocol down. See the "IOS XR Support" section on page 11-83.

CE to CE	PE to Attached CE	CE to PE Across Core	PE to PE (in VRF)	PE to PE Core	IOS	IOS XR	IPv4	IPv6
Yes	Yes	Yes	NA	NA	Yes	Yes	Yes	NS

ATM Sub-interface is Protocol Down

An ATM access circuit subinterface on a PE router is protocol down. This might be caused by incorrect subinterface parameters or by ATM Operation, Administration, and Maintenance (OAM) detecting a fault and bringing the interface down automatically. See the "IOS XR Support" section on page 11-83.

CE to CE	PE to Attached CE	CE to PE Across Core	PE to PE (in VRF)	PE to PE Core	IOS	IOS XR	IPv4	IPv6
Yes	Yes	Yes	NA	NA	Yes	Yes	Yes	NS

Calculation of the CE Access Circuit Interface IP Address is only possible if the PE interface is not Unnumbered and has a /30 Subnet mask Interface on PE

Unable to calculate the customer edge (CE) access circuit interface IP address for the PE.

CE to CE	PE to Attached CE	CE to PE Across Core	PE to PE (in VRF)	PE to PE Core	IOS	IOS XR	IPv4	IPv6
Yes	Yes	Yes	NA	NA	Yes	Yes	Yes	NA

For more details see the "IPv6 Support" section on page 11-85.

eBGP Max Prefix Exceeded for Peer

Exterior border gateway protocol (eBGP) is running between the PE and CE, however the border gateway protocol (BGP) neighbor is not established. The peer has exceeded the configured maximum number of prefixes.

CE to CE	PE to Attached CE	CE to PE Across Core	PE to PE (in VRF)	PE to PE Core	IOS	IOS XR	IPv4	IPv6
Yes	Yes	Yes	NA	NA	Yes	Yes	Yes	Yes

eBGP Neighbor Not Established, No Route Present

eBGP is running between the PE and CE, but the BGP neighbor on the PE is not established. The BGP neighbor is on a different subnet from the PE and there is no route to the neighbor in the VPN routing/forwarding (VRF).

CE to CE	PE to Attached CE	CE to PE Across Core	PE to PE (in VRF)	PE to PE Core	IOS	IOS XR	IPv4	IPv6
Yes	Yes	Yes	NA	NA	Yes	Yes	Yes	Yes

eBGP Neighbor Not Established, Possible Misconfiguration

eBGP is running between the PE and CE, but the BGP neighbor on the PE is not established. There is a route to the BGP neighbor in the VRF and it is reachable via ping. Possible CE or PE BGP configuration problem.

CE to CE	PE to Attached CE	CE to PE Across Core	PE to PE (in VRF)	PE to PE Core	IOS	IOS XR	IPv4	IPv6
Yes	Yes	Yes	NA	NA	Yes	Yes	Yes	Yes

eBGP Neighbor Not Established, Route Present

eBGP is running between the PE and CE, but the BGP neighbor on the PE is not established. The BGP neighbor is on a different subnet from the PE and there is a route to the neighbor in the VRF. However, the BGP neighbor is unreachable via ping.

CE to CE	PE to Attached CE	CE to PE Across Core	PE to PE (in VRF)	PE to PE Core	IOS	IOS XR	IPv4	IPv6
Yes	Yes	Yes	NA	NA	Yes	Yes	Yes	Yes

eBGP sites use the same AS number

The local and remote sites use eBGP and share the same AS number and neither "allowas-in" nor "as-override" is configured for the BGP neighbor within the vrf on the local PE router.

CE to CE	PE to Attached CE	CE to PE Across Core	PE to PE (in VRF)	PE to PE Core	IOS	IOS XR	IPv4	IPv6
Yes	NA	NA	NA	NA	NA	Yes	Yes	Yes

EIGRP Not Exchanging Routes

The enhanced interior gateway routing protocol (EIGRP) is running between the PE and CE and a peer relationship has been established. However, no routes have been received from EIGRP on the CE.

CE to CE	PE to Attached CE	CE to PE Across Core	PE to PE (in VRF)	PE to PE Core	IOS	IOS XR	IPv4	IPv6
Yes	Yes	Yes	NA	NA	Yes	Yes	Yes	NS

Ethernet Interface Protocol Down

An Ethernet access circuit interface on the PE router protocol is down.

CE to CE	PE to Attached CE	CE to PE Across Core	PE to PE (in VRF)	PE to PE Core	IOS	IOS XR	IPv4	IPv6
Yes	Yes	Yes	NA	NA	Yes	Yes	Yes	Yes

Ethernet Sub-Interface Protocol Down

An Ethernet access circuit subinterface on the PE router protocol is down.

CE to CE	PE to Attached CE	CE to PE Across Core	PE to PE (in VRF)	PE to PE Core	IOS	IOS XR	IPv4	IPv6
Yes	Yes	Yes	NA	NA	Yes	Yes	Yes	Yes

Frame Relay Interface Has No DLCI

A Frame Relay access circuit interface on the PE router has no data-link connection identifier (DLCI) assigned to it, or no DLCI maps to the destination IP address.

CE to CE	PE to Attached CE	CE to PE Across Core	PE to PE (in VRF)	PE to PE Core	IOS	IOS XR	IPv4	IPv6
Yes	Yes	Yes	NA	NA	Yes	Yes	Yes	NS

Frame Relay Interface Protocol Down

A Frame Relay access circuit interface on the PE router protocol is down. This might be because of line parameters or cabling faults.

CE to CE	PE to Attached CE	CE to PE Across Core	PE to PE (in VRF)	PE to PE Core	IOS	IOS XR	IPv4	IPv6
Yes	Yes	Yes	NA	NA	Yes	Yes	Yes	NS

Frame Relay Interface Has No DLCI

A multipoint Frame Relay permanent virtual circuit (PVC) on an access circuit interface on the PE router has no DLCI that maps to the destination IP address.

CE to CE	PE to Attached CE	CE to PE Across Core	PE to PE (in VRF)	PE to PE Core	IOS	IOS XR	IPv4	IPv6
Yes	Yes	Yes	NA	NA	Yes	Yes	Yes	NS

Frame Relay Interface Has No DLCI

A point-to-point Frame Relay PVC on an access circuit interface on the PE router has no DLCI assigned to it.

CE to CE	PE to Attached CE	CE to PE Across Core	PE to PE (in VRF)	PE to PE Core	IOS	IOS XR	IPv4	IPv6
Yes	Yes	Yes	NA	NA	Yes	Yes	Yes	NS

Frame Relay PVC Marked as Deleted

A Frame Relay PVC on an access circuit interface on the PE router is marked as deleted.

CE to CE	PE to Attached CE	CE to PE Across Core	PE to PE (in VRF)	PE to PE Core	IOS	IOS XR	IPv4	IPv6
Yes	Yes	Yes	NA	NA	Yes	NA	Yes	NS

Frame Relay PVC Marked as Down

A multipoint Frame Relay PVC on an access circuit interface on the PE router is marked as down.

CE to CE	PE to Attached CE	CE to PE Across Core	PE to PE (in VRF)	PE to PE Core	IOS	IOS XR	IPv4	IPv6
Yes	Yes	Yes	NA	NA	Yes	NA	Yes	NS

Frame Relay PVC Marked as Down

A point-to-point Frame Relay PVC on an access circuit interface on the PE router is marked as down.

CE to CE	PE to Attached CE	CE to PE Across Core	PE to PE (in VRF)	PE to PE Core	IOS	IOS XR	IPv4	IPv6
Yes	Yes	Yes	NA	NA	Yes	Yes	Yes	NS

Incomplete Carrier on Serial Interface

A serial access circuit interface on the PE router has an incomplete carrier.

CE to CE	PE to Attached CE	CE to PE Across Core	PE to PE (in VRF)	PE to PE Core	IOS	IOS XR	IPv4	IPv6
Yes	Yes	Yes	NA	NA	Yes	Yes	Yes	NS

Interface Administratively Down

An access circuit interface (or subinterface) on the PE router is administratively down.

CE to CE	PE to Attached CE	CE to PE Across Core	PE to PE (in VRF)	PE to PE Core	IOS	IOS XR	IPv4	IPv6
Yes	Yes	Yes	NA	NA	Yes	Yes	Yes	Yes

Interface Administratively Down

An access circuit subinterface on the PE router is administratively down.

CE to CE	PE to Attached CE	CE to PE Across Core	PE to PE (in VRF)	PE to PE Core	IOS	IOS XR	IPv4	IPv6
Yes	Yes	Yes	NA	NA	Yes	Yes	Yes	Yes

Interface in Protocol Down State

An access circuit interface on the PE router protocol is down.

CE to CE	PE to Attached CE	CE to PE Across Core	PE to PE (in VRF)	PE to PE Core	IOS	IOS XR	IPv4	IPv6
Yes	Yes	Yes	NA	NA	Yes	Yes	Yes	Yes

Interface on the PE is a Bundle Link Virtual-Access Interface

The interface on the PE is not a valid access circuit interface.

CE to CE	PE to Attached CE	CE to PE Across Core	PE to PE (in VRF)	PE to PE Core	IOS	IOS XR	IPv4	IPv6
Yes	Yes	Yes	Yes	NA	Yes	NA	Yes	NS

Interface Operationally Down

An access circuit interface on the PE router is operationally down.

CE to CE	PE to Attached CE	CE to PE Across Core	PE to PE (in VRF)	PE to PE Core	IOS	IOS XR	IPv4	IPv6
Yes	Yes	Yes	NA	NA	Yes	Yes	Yes	Yes

Intermittent ATM Failure (to the ATM Next Hop)

An ATM access circuit has intermittent ATM access to the ATM next hop. See the "IOS XR Support" section on page 11-83.

CE to CE	PE to Attached CE	CE to PE Across Core	PE to PE (in VRF)	PE to PE Core	IOS	IOS XR	IPv4	IPv6
Yes	Yes	Yes	NA	NA	Yes	Yes	Yes	NS

Intermittent ATM Failure (to the Destination)

An ATM access circuit has intermittent ATM access to the destination. See the "IOS XR Support" section on page 11-83.

CE to CE	PE to Attached CE	CE to PE Across Core	PE to PE (in VRF)	PE to PE Core	IOS	IOS XR	IPv4	IPv6
Yes	Yes	Yes	NA	NA	Yes	Yes	Yes	NS

Invalid Access Circuit IP Address Configuration

The CE router access circuit interface IP address is not in the same subnet as the attached PE access circuit interface IP address.

CE to CE	PE to Attached CE	CE to PE Across Core	PE to PE (in VRF)	PE to PE Core	IOS	IOS XR	IPv4	IPv6
Yes	Yes	Yes	NA	NA	Yes	Yes	Yes	Yes

Invalid Access Circuit IP Address Configuration

The CE access circuit interface IP address is a network address.

CE to CE	PE to Attached CE	CE to PE Across Core	PE to PE (in VRF)	PE to PE Core	IOS	IOS XR	IPv4	IPv6
Yes	Yes	Yes	NA	NA	Yes	Yes	Yes	NA

For more details see the "IPv6 Support" section on page 11-85.

Invalid Access Circuit IP Address Configuration

The CE access circuit interface IP address is a network broadcast address.

CE to CE	PE to Attached CE	CE to PE Across Core	PE to PE (in VRF)	PE to PE Core	IOS	IOS XR	IPv4	IPv6
Yes	Yes	Yes	NA	NA	Yes	Yes	Yes	NA

For more details see the "IPv6 Support" section on page 11-85.

Invalid Access Circuit IP Address Configuration

The CE access circuit interface IP address is the same as the attached PE access circuit interface IP address.

CE to CE	PE to Attached CE	CE to PE Across Core	PE to PE (in VRF)	PE to PE Core	IOS	IOS XR	IPv4	IPv6
Yes	Yes	Yes	NA	NA	Yes	Yes	Yes	Yes

IP Connectivity Problem

Unknown IP connectivity issue. An access circuit connectivity problem in the VRF instance from the PE interface to the CE.

CE to CE	PE to Attached CE	CE to PE Across Core	PE to PE (in VRF)	PE to PE Core	IOS	IOS XR	IPv4	IPv6
NA	Yes	NA	NA	NA	Yes	Yes	Yes	Yes

Missing Route

There is no route from the access circuit interface on the PE router to the destination.

CE to CE	PE to Attached CE	CE to PE Across Core	PE to PE (in VRF)	PE to PE Core	IOS	IOS XR	IPv4	IPv6
Yes	Yes	Yes	NA	NA	Yes	Yes	Yes	Yes

Missing Route

There is no route from the access circuit interface on the PE router to the customer destination.

CE to CE	PE to Attached CE	CE to PE Across Core	PE to PE (in VRF)	PE to PE Core	IOS	IOS XR	IPv4	IPv6
Yes	Yes	Yes	NA	NA	Yes	Yes	Yes	Yes

No EIGRP Peer Relationship Established

The routing protocol EIGRP is running between the PE and CE, however no peer relationship has been established.

CE to CE	PE to Attached CE	CE to PE Across Core	PE to PE (in VRF)	PE to PE Core	IOS	IOS XR	IPv4	IPv6
Yes	Yes	Yes	NA	NA	Yes	Yes	Yes	NS

No EIGRP Peer Relationship Established

The routing protocol EIGRP is running between the PE and CE. The PE and CE interfaces are on different subnets and are not using IP unnumbered. No peer relationship has been established as the PE and CE are on different subnets.

CE to CE	PE to Attached CE	CE to PE Across Core	PE to PE (in VRF)	PE to PE Core	IOS	IOS XR	IPv4	IPv6
Yes	Yes	Yes	NA	NA	Yes	Yes	Yes	NS

Route redistribution does not specify a route-policy

The routing protocol EIGRP is running between the PE and CE and is redistributing routes into MP-BGP. However no outbound route policy has been specified, which means all routes will be dropped rather than advertised.

CE to CE	PE to Attached CE	CE to PE Across Core	PE to PE (in VRF)	PE to PE Core	IOS	IOS XR	IPv4	IPv6
Yes	Yes	Yes	NA	NA	NA	Yes	Yes	NS

No OSPF Peers

Open shortest path first (OSPF) is running between the PE and CE, but no peer exists on the PE.

CE to CE	PE to Attached CE	CE to PE Across Core	PE to PE (in VRF)	PE to PE Core	IOS	IOS XR	IPv4	IPv6
Yes	Yes	Yes	NA	NA	Yes	Yes	Yes	NS

OSPF not enabled on peer interface

The interface on the router does not have OPSF enabled. OSPF must be enabled on both neighboring interface.

CE to CE	PE to Attached CE	CE to PE Across Core	PE to PE (in VRF)	PE to PE Core	IOS	IOS XR	IPv4	IPv6
Yes	NA	Yes	Yes	Yes	NS	Yes	Yes	NS

OSPF in passive mode on peer interface

OSPF on interface on the router is in passive mode.

CE to CE	PE to Attached CE	CE to PE Across Core	PE to PE (in VRF)	PE to PE Core	IOS	IOS XR	IPv4	IPv6
Yes	NA	Yes	Yes	Yes	NS	Yes	Yes	NS

OSPF Area Mismatch

OSPF is enabled on the neighboring interfaces; however the interfaces are configured in different areas.

CE to CE	PE to Attached CE	CE to PE Across Core	PE to PE (in VRF)	PE to PE Core	IOS	IOS XR	IPv4	IPv6
Yes	NA	Yes	Yes	Yes	NS	Yes	Yes	NS

OSPF Area Type Mismatch

OSPF is enabled on the neighboring interfaces; however the interfaces are configured as different area types.

CE to CE	PE to Attached CE	CE to PE Across Core	PE to PE (in VRF)	PE to PE Core	IOS	IOS XR	IPv4	IPv6
Yes	NA	Yes	Yes	Yes	NS	Yes	Yes	NS

No Routing Protocol has been Determined Running between the PE and CE and no Static Route is Present

An access circuit connectivity problem in VRF.

CE to CE	PE to Attached CE	CE to PE Across Core	PE to PE (in VRF)	PE to PE Core	IOS	IOS XR	IPv4	IPv6
Yes	Yes	Yes	NA	NA	Yes	Yes	Yes	Yes

OSPF Not Exchanging Routes

OSPF is running between the PE and CE, but a peer relationship has been established. However, no routes have been received from OSPF on the CE.

CE to CE	PE to Attached CE	CE to PE Across Core	PE to PE (in VRF)	PE to PE Core	IOS	IOS XR	IPv4	IPv6
Yes	Yes	Yes	NA	NA	Yes	Yes	Yes	NS

OSPF Peers Not Established

OSPF is running between the PE and CE, but a peer relationship has not been established.

CE to CE	PE to Attached CE	CE to PE Across Core	PE to PE (in VRF)	PE to PE Core	IOS	IOS XR	IPv4	IPv6
Yes	Yes	Yes	NA	NA	Yes	Yes	Yes	NA

OSPF Timer Mismatch

OSPF is enabled on the neighboring interfaces; however the interfaces have different values configured for their [hello|dead] timers.

CE to CE	PE to Attached CE	CE to PE Across Core	PE to PE (in VRF)	PE to PE Core	IOS	IOS XR	IPv4	IPv6
Yes	NA	Yes	Yes	Yes	NS	Yes	Yes	NS

OSPF Peers Not Established

OSPF is running between the PE and CE. However, a peer relationship has not been established as PE and CE interfaces are on different subnets and are not using IP unnumbered.

CE to CE	PE to Attached CE	CE to PE Across Core	PE to PE (in VRF)	PE to PE Core	IOS	IOS XR	IPv4	IPv6
Yes	Yes	Yes	NA	NA	Yes	Yes	Yes	NS

Route redistribution does not specify a route-policy

The routing protocol OSPF is running between the PE and CE and is redistributing routes into MP-BGP. However no outbound route policy has been specified, which means all routes will be dropped rather than advertised.

CE to CE	PE to Attached CE	CE to PE Across Core	PE to PE (in VRF)	PE to PE Core	IOS	IOS XR	IPv4	IPv6
Yes	Yes	Yes	NA	NA	NS	Yes	Yes	NS

PE has No Route to CE

Connected PE and CE interfaces are on different subnets. No routing protocol has been determined running between the PE and CE and no static route to the CE is present.

CE to CE	PE to Attached CE	CE to PE Across Core	PE to PE (in VRF)	PE to PE Core	IOS	IOS XR	IPv4	IPv6
Yes	Yes	Yes	NA	NA	Yes	NA	Yes	NS

RIB Failure

A route from the PE to a destination in a VRF has not been installed in the VRF routing table. This has been identified as a Routing Information Base (RIB) failure.

CE to CE	PE to Attached CE	CE to PE Across Core	PE to PE (in VRF)	PE to PE Core	IOS	IOS XR	IPv4	IPv6
Yes	Yes	Yes	Yes	NA	Yes	NA	Yes	NS

RIP Misconfiguration

Routing information protocol (RIP) is running between the PE and CE, but no routes have been received from RIP on the CE.

CE to CE	PE to Attached CE	CE to PE Across Core	PE to PE (in VRF)	PE to PE Core	IOS	IOS XR	IPv4	IPv6
Yes	Yes	Yes	NA	NA	Yes	Yes	Yes	NS

RIP Not Exchanging Routes

RIP is running between the PE and CE, but no routes have been received from RIP on the CE as the PE and CE interfaces are on different subnets and are not using IP unnumbered.

CE to CE	PE to Attached CE	CE to PE Across Core	PE to PE (in VRF)	PE to PE Core	IOS	IOS XR	IPv4	IPv6
Yes	Yes	Yes	NA	NA	Yes	Yes	Yes	NS

Route redistribution does not specify a route-policy

The routing protocol RIP is running between the PE and CE and is redistributing routes into MP-BGP. However no outbound route policy has been specified, which means all routes will be dropped rather than advertised.

CE to CE	PE to Attached CE	CE to PE Across Core	PE to PE (in VRF)	PE to PE Core	IOS	IOS XR	IPv4	IPv6
Yes	Yes	Yes	NA	NA	NA	Yes	Yes	NS

Serial Interface in Loopback Mode

A serial access circuit interface on the PE router is configured in loopback mode.

CE to CE	PE to Attached CE	CE to PE Across Core	PE to PE (in VRF)	PE to PE Core	IOS	IOS XR	IPv4	IPv6
Yes	Yes	Yes	NA	NA	Yes	Yes	Yes	NS

Serial Interface Operationally Down

A serial access circuit interface on the PE router is operationally down.

CE to CE	PE to Attached CE	CE to PE Across Core	PE to PE (in VRF)	PE to PE Core	IOS	IOS XR	IPv4	IPv6
Yes	Yes	Yes	NA	NA	Yes	Yes	Yes	NS

Static IP address on ATM Point-to-Point Interface

An ATM access circuit has a static IP address mapping on an ATM point-to-point subinterface. Neither a static mapping nor an address resolution protocol (ARP) are required on a point-to-point subinterface because there is a single VC and a single path for the traffic. See the "IOS XR Support" section on page 11-83.

CE to CE	PE to Attached CE	CE to PE Across Core	PE to PE (in VRF)	PE to PE Core	IOS	IOS XR	IPv4	IPv6
Yes	Yes	Yes	NA	NA	Yes	Yes	Yes	NS

Sub-interface in Protocol Down State

An access circuit subinterface on the PE router protocol is down.

CE to CE	PE to Attached CE	CE to PE Across Core	PE to PE (in VRF)	PE to PE Core	IOS	IOS XR	IPv4	IPv6
Yes	Yes	Yes	NA	NA	Yes	Yes	Yes	Yes

Undiagnosed ATM Failure (ATM Pings Failed but the ATM Segment Ping Succeeded)

An ATM access circuit connection is broken. The end-to-end ATM ping failed, but the ATM segment ping succeeded. This might be caused by various issues such as incorrect ATM line parameters, misconfigured ATM routing, CE or ATM cloud interfaces being down, or devices being down. See the "IOS XR Support" section on page 11-83.

CE to CE	PE to Attached CE	CE to PE Across Core	PE to PE (in VRF)	PE to PE Core	IOS	IOS XR	IPv4	IPv6
Yes	Yes	Yes	NA	NA	Yes	Yes	Yes	NS

Undiagnosed ATM Failure (End to End and Segment ATM Pings Failed)

An ATM access circuit connection is broken. Both the end-to-end and segment ATM pings failed. This might be caused by various issues such as incorrect ATM line parameters, misconfigured ATM routing on the ATM next hop, next hop interfaces being down, or devices being down. See the IOS XR Support, page 11-83.

CE to CE	PE to Attached CE	CE to PE Across Core	PE to PE (in VRF)	PE to PE Core	IOS	IOS XR	IPv4	IPv6
Yes	Yes	Yes	NA	NA	Yes	Yes	Yes	NS

Virtual Template Interface has been Specified for the PE Access Circuit Interface

The PE interface on the PE is not a valid access circuit interface.

CE to CE	PE to Attached CE	CE to PE Across Core	PE to PE (in VRF)	PE to PE Core	IOS	IOS XR	IPv4	IPv6
Yes	Yes	Yes	Yes	NA	Yes	NA	Yes	NS

MPLS Edge

BGP Next Hop Interface Admin Down

BGP next hop on PE is assigned to a loopback interface. However, that interface is administratively down.

CE to CE	PE to Attached CE	CE to PE Across Core	PE to PE (in VRF)	PE to PE Core	IOS	IOS XR	IPv4	IPv6
Yes	NA	Yes	Yes	Yes	Yes	Yes	Yes	Yes

BGP Next Hop is Not Assigned to an Interface

The BGP next hop for routes to the remote site PE is not assigned to an interface on the remote PE.

CE to CE	PE to Attached CE	CE to PE Across Core	PE to PE (in VRF)	PE to PE Core	IOS	IOS XR	IPv4	IPv6
Yes	NA	Yes	Yes	Yes	Yes	Yes	Yes	Yes

BGP Not Active

BGP is not active on the PE router.

CE to CE	PE to Attached CE	CE to PE Across Core	PE to PE (in VRF)	PE to PE Core	IOS	IOS XR	IPv4	IPv6
Yes	Yes	Yes	Yes	NA	Yes	Yes	Yes	Yes

BGP Peers Using Same BGP Next Hop

BGP VPNv4/VPNv6 peers use the same BGP next hop. This prevents correct route distribution of PE routes. Other routing problems might also be encountered.

CE to CE	PE to Attached CE	CE to PE Across Core	PE to PE (in VRF)	PE to PE Core	IOS	IOS XR	IPv4	IPv6
Yes	NA	Yes	Yes	NA	Yes	Yes	Yes	Yes

BGP Peers Using Same Router Identifier (RID)

BGP VPNv4/VPNv6 peers use the same router identifier. This prevents correct route distribution of PE routes. Other routing problems might also be encountered.

CE to CE	PE to Attached CE	CE to PE Across Core	PE to PE (in VRF)	PE to PE Core	IOS	IOS XR	IPv4	IPv6
Yes	NA	Yes	Yes	Yes	Yes	Yes	Yes	Yes

Multiple Access Circuits in the same subnet

LSP Connectivity problem, control plane issue. The next-hop for the current BGP selected route(s) to the remote PE router is not assigned to an interface on the local PE router. There are multiple VPNv4/VPNv6 routes found within the vrf on the locale PE router to the remote prefix.

CE to CE	PE to Attached CE	CE to PE Across Core	PE to PE (in VRF)	PE to PE Core	IOS	IOS XR	IPv4	IPv6
Yes	NA	NA	NA	NA	Yes	Yes	Yes	Yes

BGP to LFIB Mismatch

Untagged entry has mismatch between BGP and label forwarding information base (LFIB) tables.

CE to CE	PE to Attached CE	CE to PE Across Core	PE to PE (in VRF)	PE to PE Core	IOS	IOS XR	IPv4	IPv6
Yes	Yes	Yes	Yes	NA	Yes	Yes	Yes	Yes

BGP to LFIB Mismatch

The forwarding information base (FIB) and BGP entries are inconsistent.

CE to CE	PE to Attached CE	CE to PE Across Core	PE to PE (in VRF)	PE to PE Core	IOS	IOS XR	IPv4	IPv6
Yes	Yes	Yes	Yes	NA	Yes	Yes	Yes	Yes

Duplicate BGP Next Hop

Duplicate IP address found in the network for the BGP next hop on PE.

CE to CE	PE to Attached CE	CE to PE Across Core	PE to PE (in VRF)	PE to PE Core	IOS	IOS XR	IPv4	IPv6
Yes	NA	Yes	Yes	Yes	Yes	Yes	Yes	Yes

Duplicate IP Address

There is a duplicate IP address configured on the PE router that conflicts with an access circuit interface.

CE to CE	PE to Attached CE	CE to PE Across Core	PE to PE (in VRF)	PE to PE Core	IOS	IOS XR	IPv4	IPv6
Yes	Yes	Yes	NA	NA	Yes	Yes	Yes	Yes

Duplicate IP Address

Duplicate IP address found in the network for the BGP router identifier on PE.

CE to CE	PE to Attached CE	CE to PE Across Core	PE to PE (in VRF)	PE to PE Core	IOS	IOS XR	IPv4	IPv6
Yes	NA	Yes	Yes	Yes	Yes	Yes	Yes	Yes

eBGP not Redistributing Connected Routes into MP-BGP

The routing protocol eBGP is running between the PE and CE, but eBGP on the PE is not redistributing connected routes into Multi Protocol (MP)-BGP, also there is no explicit network statement.

CE to CE	PE to Attached CE	CE to PE Across Core	PE to PE (in VRF)	PE to PE Core	IOS	IOS XR	IPv4	IPv6
Yes	Yes	Yes	NA	NA	Yes	Yes	Yes	Yes

FIB to LFIB Mismatch

Aggregate entry has mismatch between FIB and LFIB tables.

CE to CE	PE to Attached CE	CE to PE Across Core	PE to PE (in VRF)	PE to PE Core	IOS	IOS XR	IPv4	IPv6
Yes	Yes	Yes	NA	NA	Yes	Yes	Yes	Yes

Ingress FIB to Egress LFIB Mismatch

Egress LFIB and Ingress FIB inconsistency.

CE to CE	PE to Attached CE	CE to PE Across Core	PE to PE (in VRF)	PE to PE Core	IOS	IOS XR	IPv4	IPv6
Yes	Yes	Yes	Yes	NA	Yes	Yes	Yes	Yes

Inconsistent BGP Entries

BGP entries are inconsistent for the VRF.

CE to CE	PE to Attached CE	CE to PE Across Core	PE to PE (in VRF)	PE to PE Core	IOS	IOS XR	IPv4	IPv6
Yes	Yes	Yes	Yes	NA	Yes	Yes	Yes	Yes

Label mismatch or interfaces on different VPNs Down

VPN connectivity problem in VRF from PE to destination. BGP VPNv4/VPNv6 label for prefix do not match. This might indicate a label mismatch or interfaces on different VPNs. Check that the interfaces selected are on the same VRF.

CE to CE	PE to Attached CE	CE to PE Across Core	PE to PE (in VRF)	PE to PE Core	IOS	IOS XR	IPv4	IPv6
NA	NA	NA	Yes	NA	Yes	Yes	Yes	Yes

PE interface is administratively down

The access circuit interface on the PE router is administratively down.

CE to CE	PE to Attached CE	CE to PE Across Core	PE to PE (in VRF)	PE to PE Core	IOS	IOS XR	IPv4	IPv6
Yes	Yes	Yes	NA	NA	Yes	Yes	Yes	NS

Missing Router Identifier (RID)

Unable to determine the local router identifier on the PE.

CE to CE	PE to Attached CE	CE to PE Across Core	PE to PE (in VRF)	PE to PE Core	IOS	IOS XR	IPv4	IPv6
NA	NA	NA	NA	Yes	Yes	Yes	Yes	Yes

Missing VPNv4 Address Family Configuration

VPNv4 configuration missing; Virtual private network (VPN) label exchange problem. VPNv4 address family configuration missing from BGP router configuration on the PE router. This results in routes being dropped.

CE to CE	PE to Attached CE	CE to PE Across Core	PE to PE (in VRF)	PE to PE Core	IOS	IOS XR	IPv4	IPv6
Yes	NA	Yes	Yes	NA	Yes	Yes	Yes	Yes

Missing VPNv6 Address Family Configuration

VPNv6 configuration missing; Virtual private network (VPN) label exchange problem. VPNv6 address family configuration missing from BGP router configuration on the PE router. This results in routes being dropped.

CE to CE	PE to Attached CE	CE to PE Across Core	PE to PE (in VRF)	PE to PE Core	IOS	IOS XR	IPv4	IPv6
Yes	NA	Yes	Yes	NA	NA	Yes	NA	Yes

MPLS LDP Package not enabled on IOS XR Router

The MPLS LDP package is not enabled on the IOS XR router.

CE to CE	PE to Attached CE	CE to PE Across Core	PE to PE (in VRF)	PE to PE Core	IOS	IOS XR	IPv4	IPv6
Yes	NA	Yes	Yes	Yes	NA	Yes	Yes	Yes

MPLS Package installed but not active on IOS XR Router

The MPLS package is installed, but it is not active on the IOS XR router.

CE to CE	PE to Attached CE	CE to PE Across Core	PE to PE (in VRF)	PE to PE Core	IOS	IOS XR	IPv4	IPv6
Yes	NA	Yes	Yes	Yes	NA	Yes	Yes	Yes

MPLS Package not installed on IOS XR Router

The MPLS package is not installed on the IOS XR router.

CE to CE	PE to Attached CE	CE to PE Across Core	PE to PE (in VRF)	PE to PE Core	IOS	IOS XR	IPv4	IPv6
Yes	NA	Yes	Yes	Yes	NA	Yes	Yes	Yes

No MP-BGP Neighbors

There are no MP-BGP neighbors defined on the PE router.

CE to CE	PE to Attached CE	CE to PE Across Core	PE to PE (in VRF)	PE to PE Core	IOS	IOS XR	IPv4	IPv6
Yes	NA	Yes	Yes	NA	Yes	Yes	Yes	Yes

No MP-BGP Neighbor Session Established

No MP-BGP neighbor session established on the PE router.

CE to CE	PE to Attached CE	CE to PE Across Core	PE to PE (in VRF)	PE to PE Core	IOS	IOS XR	IPv4	IPv6
Yes	NA	Yes	Yes	NA	Yes	Yes	Yes	Yes

No VPN Label For Prefix

No VPN label has been allocated for the prefix.

CE to CE	PE to Attached CE	CE to PE Across Core	PE to PE (in VRF)	PE to PE Core	IOS	IOS XR	IPv4	IPv6
Yes	NA	Yes	Yes	NA	Yes	Yes	Yes	Yes

No VRF Associated with PE Interface

An interface on the PE router has no VRF associated.

CE to CE	PE to Attached CE	CE to PE Across Core	PE to PE (in VRF)	PE to PE Core	IOS	IOS XR	IPv4	IPv6
Yes	NA	Yes	Yes	NA	Yes	Yes	Yes	Yes

OSPF Loopback Interface Uses A Non /32 Netmask

VPNv4 routes are being advertised to IBGP neighbors by the PE. The address of the next hop is a loopback interface that does not have a /32 mask defined. OSPF is being used on this loopback interface, and the OSPF network type of this interface is loopback. OSPF advertises this IP address as a host route (with mask /32), regardless of what mask is configured. This advertising conflicts with TDP/LDP, that uses configured masks, so the TDP/LDP neighbors might not receive a label for the routes advertised by this router. This condition could break connectivity between sites that belong to the same VPN.

CE to CE	PE to Attached CE	CE to PE Across Core	PE to PE (in VRF)	PE to PE Core	IOS	IOS XR	IPv4	IPv6
Yes	NA	Yes	Yes	Yes	Yes	NA	Yes	NA

PE Interface Has No IP Address

An interface on the PE router has no IP address.

CE to CE	PE to Attached CE	CE to PE Across Core	PE to PE (in VRF)	PE to PE Core	IOS	IOS XR	IPv4	IPv6
Yes	NA	Yes	Yes	NA	Yes	Yes	Yes	Yes

Router ID Loopback Interface Down

The loopback interface used to assign the local router ID on the PE is administratively down.

CE to CE	PE to Attached CE	CE to PE Across Core	PE to PE (in VRF)	PE to PE Core	IOS	IOS XR	IPv4	IPv6
Yes	Yes	Yes	NA	NA	Yes	Yes	Yes	Yes

Routes not Redistributed to or from MP-BGP

Routes are not being redistributed to or from MP-BGP on the PE.

CE to CE	PE to Attached CE	CE to PE Across Core	PE to PE (in VRF)	PE to PE Core	IOS	IOS XR	IPv4	IPv6
Yes	Yes	Yes	NA	NA	Yes	NA	Yes	NA

Static Route to Remote Prefix

A static route to a remote prefix has been configured within a VRF on the PE.

CE to CE	PE to Attached CE	CE to PE Across Core	PE to PE (in VRF)	PE to PE Core	IOS	IOS XR	IPv4	IPv6
Yes	NA	Yes	Yes	NA	Yes	Yes	Yes	Yes

Traffic Administratively Blocked

VPN connectivity problem from the PE to the destination due to traffic being administratively blocked.

CE to CE	PE to Attached CE	CE to PE Across Core	PE to PE (in VRF)	PE to PE Core	IOS	IOS XR	IPv4	IPv6
Yes	NA	Yes	NA	NA	Yes	Yes	Yes	Yes

Troubleshooting of the Layer 3 VPN has been Unable to Find the Cause of the Failure

LSP connectivity problem.

CE to CE	PE to Attached CE	CE to PE Across Core	PE to PE (in VRF)	PE to PE Core	IOS	IOS XR	IPv4	IPv6
Yes	NA	Yes	Yes	Yes	Yes	Yes	Yes	Yes

Troubleshooting of the Layer 3 VPN has been Unable to Find the Cause of the Failure

VPN connectivity problem in VRF from PE to destination.

CE to CE	PE to Attached CE	CE to PE Across Core	PE to PE (in VRF)	PE to PE Core	IOS	IOS XR	IPv4	IPv6
Yes	NA	Yes	Yes	NA	Yes	Yes	Yes	Yes

VRF Route Target Import/Export Mismatch

VRF route target import/export mismatch between the PE devices.

CE to CE	PE to Attached CE	CE to PE Across Core	PE to PE (in VRF)	PE to PE Core	IOS	IOS XR	IPv4	IPv6
Yes	NA	Yes	Yes	NA	Yes	Yes	Yes	Yes

MPLS Core

An Invalid PE Interface has been Specified

Interface supplied as the LSP Endpoint does not exist on the PE router.

CE to CE	PE to Attached CE	CE to PE Across Core	PE to PE (in VRF)	PE to PE Core	IOS	IOS XR	IPv4	IPv6
NA	NA	NA	NA	Yes	Yes	Yes	Yes	NA

Broken LDP Neighbor Session

LDP session with downstream neighbor broken. Route processor/line card forwarding discrepancy on the router. See the "IOS XR Support" section on page 11-83.

CE to CE	PE to Attached CE	CE to PE Across Core	PE to PE (in VRF)	PE to PE Core	IOS	IOS XR	IPv4	IPv6
Yes	NA	Yes	Yes	Yes	Yes	NA	Yes	Yes

CEF Not Enabled On Router

CEF has not been enabled on a router. See the "IOS XR Support" section on page 11-83.

CE to CE	PE to Attached CE	CE to PE Across Core	PE to PE (in VRF)	PE to PE Core	IOS	IOS XR	IPv4	IPv6
Yes	NA	Yes	Yes	Yes	Yes	NA	Yes	Yes

Distributed LFIB Table Discrepancy

There is a discrepancy in the LFIB table between the route processor and line cards.

CE to CE	PE to Attached CE	CE to PE Across Core	PE to PE (in VRF)	PE to PE Core	IOS	IOS XR	IPv4	IPv6
Yes	NA	Yes	Yes	NA	Yes	Yes	Yes	Yes

Distributed FIB Table Discrepancy

There is a discrepancy in the FIB table between the route processor and line cards.

CE to CE	PE to Attached CE	CE to PE Across Core	PE to PE (in VRF)	PE to PE Core	IOS	IOS XR	IPv4	IPv6
Yes	NA	Yes	Yes	NA	Yes	Yes	Yes	Yes

Label Inconsistency

LFIB local tag, received packet, and LDP local binding label inconsistent on the router. See the "IOS XR Support" section on page 11-83.

CE to CE	PE to Attached CE	CE to PE Across Core	PE to PE (in VRF)	PE to PE Core	IOS	IOS XR	IPv4	IPv6
Yes	NA	Yes	Yes	Yes	Yes	NA	Yes	Yes

LDP Host Not Reachable

The host is not reachable from the label switch router (LSR). This could be caused by no LDP session on router for the downstream router, LDP ID not reachable because of IGP problem, ACL configured that is blocking LDP packets, or an authentication problem.

CE to CE	PE to Attached CE	CE to PE Across Core	PE to PE (in VRF)	PE to PE Core	IOS	IOS XR	IPv4	IPv6
Yes	NA	Yes	Yes	Yes	Yes	Yes	Yes	Yes

LDP Label Mismatch

Label received for a prefix did not match the label sent. See the "IOS XR Support" section on page 11-83.

CE to CE	PE to Attached CE	CE to PE Across Core	PE to PE (in VRF)	PE to PE Core	IOS	IOS XR	IPv4	IPv6
Yes	NA	Yes	Yes	Yes	Yes	NA	Yes	Yes

LDP Neighbors Not Discovered

LDP neighbors have not been discovered. Generic LDP discovery problem found on an interface of the device with its downstream neighbor. See the "IOS XR Support" section on page 11-83.

CE to CE	PE to Attached CE	CE to PE Across Core	PE to PE (in VRF)	PE to PE Core	IOS	IOS XR	IPv4	IPv6
Yes	NA	Yes	Yes	Yes	Yes	NA	Yes	Yes

LDP Neighbors Not Discovered

LDP neighbors have not been discovered. An interface on a router has an ACL configured that could be preventing LDP neighbor discovery.

CE to CE	PE to Attached CE	CE to PE Across Core	PE to PE (in VRF)	PE to PE Core	IOS	IOS XR	IPv4	IPv6
Yes	NA	Yes	Yes	Yes	Yes	Yes	Yes	Yes

LDP Neighbors Not Established

LSP connectivity problem, control plane issue. The LDP session is not established. The interface has an ACL configured that could be blocking LDP session establishment on port 646.

CE to CE	PE to Attached CE	CE to PE Across Core	PE to PE (in VRF)	PE to PE Core	IOS	IOS XR	IPv4	IPv6
Yes	NA	Yes	Yes	Yes	Yes	Yes	Yes	Yes

LDP/TDP Mismatch

LDP and TDP have been enabled on opposite ends of a link. See the "IOS XR Support" section on page 11-83.

CE to CE	PE to Attached CE	CE to PE Across Core	PE to PE (in VRF)	PE to PE Core	IOS	IOS XR	IPv4	IPv6
Yes	NA	Yes	Yes	Yes	Yes	Yes	Yes	Yes

LSP Reply Path Problem

LSP connectivity problem with the reply path. See the "IOS XR Support" section on page 11-83.

CE to CE	PE to Attached CE	CE to PE Across Core	PE to PE (in VRF)	PE to PE Core	IOS	IOS XR	IPv4	IPv6
Yes	NA	Yes	Yes	Yes	Yes	NA	Yes	Yes

Missing LFIB Entry

LFIB entry missing. Could be because of misrouting in an earlier router, or due to duplicate loopbacks. See the "IOS XR Support" section on page 11-83.

CE to CE	PE to Attached CE	CE to PE Across Core	PE to PE (in VRF)	PE to PE Core	IOS	IOS XR	IPv4	IPv6
Yes	NA	Yes	Yes	Yes	Yes	NA	Yes	Yes

Missing or Untagged Return Path

Return path from the core router absent or untagged for the prefix. See the "IOS XR Support" section on page 11-83.

CE to CE	PE to Attached CE	CE to PE Across Core	PE to PE (in VRF)	PE to PE Core	IOS	IOS XR	IPv4	IPv6
Yes	NA	Yes	Yes	Yes	Yes	NA	Yes	Yes

MPLS Label Space Exhausted

MPLS label space exhausted on a router.

CE to CE	PE to Attached CE	CE to PE Across Core	PE to PE (in VRF)	PE to PE Core	IOS	IOS XR	IPv4	IPv6
Yes	NA	Yes	Yes	Yes	Yes	Yes	Yes	Yes

MPLS Not Enabled Globally

MPLS has not been enabled globally on router.

CE to CE	PE to Attached CE	CE to PE Across Core	PE to PE (in VRF)	PE to PE Core	IOS	IOS XR	IPv4	IPv6
Yes	NA	Yes	Yes	Yes	Yes	NA	Yes	Yes

MPLS Not Enabled On Interface

MPLS has not been enabled on an interface. See the "IOS XR Support" section on page 11-83.

CE to CE	PE to Attached CE	CE to PE Across Core	PE to PE (in VRF)	PE to PE Core	IOS	IOS XR	IPv4	IPv6
Yes	NA	Yes	Yes	Yes	Yes	Yes	Yes	Yes

No Entry for Label

No entry in LFIB for incoming label going to the destination prefix. See the "IOS XR Support" section on page 11-83.

CE to CE	PE to Attached CE	CE to PE Across Core	PE to PE (in VRF)	PE to PE Core	IOS	IOS XR	IPv4	IPv6
Yes	NA	Yes	Yes	Yes	Yes	NA	Yes	Yes

No LDP Session with Neighbor

No LDP session on router exists with neighbor.

CE to CE	PE to Attached CE	CE to PE Across Core	PE to PE (in VRF)	PE to PE Core	IOS	IOS XR	IPv4	IPv6
Yes	NA	Yes	Yes	Yes	Yes	Yes	Yes	Yes

No Valid Next-Hop Entry

No valid entry can be found for the next-hop from the current device. See the "IOS XR Support" section on page 11-83.

CE to CE	PE to Attached CE	CE to PE Across Core	PE to PE (in VRF)	PE to PE Core	IOS	IOS XR	IPv4	IPv6
Yes	NA	Yes	Yes	Yes	Yes	NA	Yes	Yes

Routing Loop In Forwarding Path

A routing loop is present in the forwarding path.

CE to CE	PE to Attached CE	CE to PE Across Core	PE to PE (in VRF)	PE to PE Core	IOS	IOS XR	IPv4	IPv6
Yes	NA	Yes	Yes	Yes	Yes	Yes	Yes	Yes

All MPLS enabled core facing interfaces are down

LSP connectivity problem, control plane issue. No MPLS enabled interface is operationally up.

CE to CE	PE to Attached CE	CE to PE Across Core	PE to PE (in VRF)	PE to PE Core	IOS	IOS XR	IPv4	IPv6
Yes	NA	Yes	Yes	Yes	Yes	Yes	Yes	Yes

Label Advertising Not Enabled

Label advertisement is disabled on router.

CE to CE	PE to Attached CE	CE to PE Across Core	PE to PE (in VRF)	PE to PE Core	IOS	IOS XR	IPv4	IPv6
Yes	NA	Yes	Yes	Yes	Yes	Yes	Yes	Yes

Label Advertising Possibly Denied by ACL

Label advertisement has been globally disabled, but selectively enabled for one or more access lists. The ACL(s) might be denying the advertising of labels to the destination prefix.

CE to CE	PE to Attached CE	CE to PE Across Core	PE to PE (in VRF)	PE to PE Core	IOS	IOS XR	IPv4	IPv6
Yes	NA	Yes	Yes	Yes	Yes	Yes	Yes	Yes

TTL Propagation Disabled

Could not troubleshoot or detect failure point because the device is not propagating the Time To Live (TTL).

CE to CE	PE to Attached CE	CE to PE Across Core	PE to PE (in VRF)	PE to PE Core	IOS	IOS XR	IPv4	IPv6
Yes	NA	Yes	Yes	Yes	Yes	Yes	Yes	Yes

Check tunnel traffic admission policy

No traffic admission policy (such as via autoroute announce, or Policy Based Tunnel Selection (PBTS) or a static route) configured on the TE Tunnel.

CE to CE	PE to Attached CE	CE to PE Across Core	PE to PE (in VRF)	PE to PE Core	IOS	IOS XR	IPv4	IPv6
Yes	NA	Yes	Yes	Yes	NA	Yes	Yes	Yes

Check if MPLS is enabled on the tunnel interface

Incomplete configuration detected for the MPLS TE Tunnel interface.

CE to CE	PE to Attached CE	CE to PE Across Core	PE to PE (in VRF)	PE to PE Core	IOS	IOS XR	IPv4	IPv6
Yes	NA	Yes	NA	Yes	NS	Yes	Yes	Yes

Check that the primary and backup tunnel's interfaces are up

The tunnel interface on the router is administratively down.

CE to CE	PE to Attached CE	CE to PE Across Core	PE to PE (in VRF)	PE to PE Core	IOS	IOS XR	IPv4	IPv6
Yes	NA	Yes	Yes	Yes	NS	Yes	Yes	Yes

Tunnel config not present at headend

The TE Tunnel configuration is not present at the headend router.

CE to CE	PE to Attached CE	CE to PE Across Core	PE to PE (in VRF)	PE to PE Core	IOS	IOS XR	IPv4	IPv6
Yes	NA	Yes	Yes	Yes	NS	Yes	Yes	Yes

Check that the tunnels outgoing interface is operational

The outgoing interface of FRR primary tunnel configured on the router is operationally down.

CE to CE	PE to Attached CE	CE to PE Across Core	PE to PE (in VRF)	PE to PE Core	IOS	IOS XR	IPv4	IPv6
Yes	NA	Yes	Yes	Yes	NS	Yes	Yes	Yes

TE not enabled globally on router

MPLS Traffic Engineering is not enabled globally on the router.

CE to CE	PE to Attached CE	CE to PE Across Core	PE to PE (in VRF)	PE to PE Core	IOS	IOS XR	IPv4	IPv6
Yes	NA	Yes	Yes	Yes	NS	Yes	Yes	Yes

TE not enabled globally on the interface

MPLS Traffic Engineering is not enabled on the interface of the router.

CE to CE	PE to Attached CE	CE to PE Across Core	PE to PE (in VRF)	PE to PE Core	IOS	IOS XR	IPv4	IPv6
Yes	NA	Yes	Yes	Yes	NS	Yes	Yes	Yes

No IP Address assigned for tunnel

MPLS Traffic Engineering Tunnel has no IP address assigned.

CE to CE	PE to Attached CE	CE to PE Across Core	PE to PE (in VRF)	PE to PE Core	IOS	IOS XR	IPv4	IPv6
Yes	NA	Yes	Yes	Yes	NS	Yes	Yes	Yes

Tunnel destination invalid

The destination address configured for the tunnel is unreachable.

CE to CE	PE to Attached CE	CE to PE Across Core	PE to PE (in VRF)	PE to PE Core	IOS	IOS XR	IPv4	IPv6
Yes	NA	Yes	Yes	Yes	NS	Yes	Yes	Yes

Tunnel is administratively shut down

Tunnel has been admin shut down on the router.

CE to CE	PE to Attached CE	CE to PE Across Core	PE to PE (in VRF)	PE to PE Core	IOS	IOS XR	IPv4	IPv6
Yes	NA	Yes	Yes	Yes	NS	Yes	Yes	Yes

Missing OSPF configuration for TE

Router has not configured OSPF for MPLS TE.

CE to CE	PE to Attached CE	CE to PE Across Core	PE to PE (in VRF)	PE to PE Core	IOS	IOS XR	IPv4	IPv6
Yes	NA	Yes	Yes	Yes	NS	Yes	Yes	Yes

Node is not advertising MPLS TE links

Router is not advertising itself through OSPF as an MPLS TE link.

CE to CE	PE to Attached CE	CE to PE Across Core	PE to PE (in VRF)	PE to PE Core	IOS	IOS XR	IPv4	IPv6
Yes	NA	Yes	Yes	Yes	NS	Yes	Yes	Yes

No targeted LDP session exists between peer PE's

Remote Site PE router is not accepting targeted LDP session requests.

CE to CE	PE to Attached CE	CE to PE Across Core	PE to PE (in VRF)	PE to PE Core	IOS	IOS XR	IPv4	IPv6
Yes	NA	Yes	NA	Yes	NS	Yes	Yes	Yes

Blocking ACL causing targeted LDP session setup problem

Router has an access control list which is blocking LDP messages (on TCP port 646).

CE to CE	PE to Attached CE	CE to PE Across Core	PE to PE (in VRF)	PE to PE Core	IOS	IOS XR	IPv4	IPv6
Yes	NA	Yes	Yes	Yes	NS	Yes	Yes	Yes

Targeted LDP not established/operational between PE's

LDP has been unable to establish a targeted session between the devices due to peer PE router being unreachable.

CE to CE	PE to Attached CE	CE to PE Across Core	PE to PE (in VRF)	PE to PE Core	IOS	IOS XR	IPv4	IPv6
Yes	NA	Yes	Yes	Yes	NS	Yes	Yes	Yes

Tunnel Connectivity Failure No Targeted LDP Configuration

LDP discovery failure for the neighbor devices. The tunnel tail end device is not accepting LDP targeted hellos.

CE to CE	PE to Attached CE	CE to PE Across Core	PE to PE (in VRF)	PE to PE Core	IOS	IOS XR	IPv4	IPv6
Yes	NA	Yes	Yes	Yes	Yes	Yes	Yes	Yes

For link protection, check that the backup tunnel does not pass through the protected interface

There is a FRR backup tunnel configured on the router. It is configured to protect a link on router (as NHOP), which is on the path the primary tunnel takes. However the configured backup tunnel is configured to traverse this link.

CE to CE	PE to Attached CE	CE to PE Across Core	PE to PE (in VRF)	PE to PE Core	IOS	IOS XR	IPv4	IPv6
Yes	NA	Yes	Yes	Yes	NS	Yes	Yes	Yes

Check the primary tunnel has FRR enabled with 'fast-reroute'

A tunnel has been detected on router. It appears that this tunnel is intended to be a primary tunnel, but it does not have the fast-reroute configuration as required.

CE to CE	PE to Attached CE	CE to PE Across Core	PE to PE (in VRF)	PE to PE Core	IOS	IOS XR	IPv4	IPv6
Yes	NA	Yes	NA	Yes	NS	Yes	Yes	Yes

Node protection - Check if backup tunnel path is explicit and does not contain the protected node interface in path

There is a FRR backup tunnel configured on router. It is configured to protect a router (as NNHOP), which is on the path the primary tunnel takes. However the configured backup tunnel is configured to traverse a link on this router.

CE to CE	PE to Attached CE	CE to PE Across Core	PE to PE (in VRF)	PE to PE Core	IOS	IOS XR	IPv4	IPv6
Yes	NA	Yes	Yes	Yes	NS	Yes	Yes	Yes

Check if the primary & backup tunnel merge point is reachable

The merge point router for FRR primary tunnel and FRR backup tunnel is unreachable.

CE to CE	PE to Attached CE	CE to PE Across Core	PE to PE (in VRF)	PE to PE Core	IOS	IOS XR	IPv4	IPv6
Yes	NA	Yes	Yes	Yes	NS	Yes	Yes	Yes

TE ping failure

The *ping mpls traffic-eng tunnel* command returned failure.

CE to CE	PE to Attached CE	CE to PE Across Core	PE to PE (in VRF)	PE to PE Core	IOS	IOS XR	IPv4	IPv6
Yes	NA	Yes	NA	Yes	NS	Yes	Yes	Yes

Tunnel operationally down

The *show mpls traffic-eng tunnels* command has indicated that the TE Tunnel is down.

CE to CE	PE to Attached CE	CE to PE Across Core	PE to PE (in VRF)	PE to PE Core	IOS	IOS XR	IPv4	IPv6
Yes	NA	Yes	NA	Yes	NS	Yes	Yes	Yes

Tunnel Connectivity Failure

Unknown MPLS TE connectivity problem.

CE to CE	PE to Attached CE	CE to PE Across Core	PE to PE (in VRF)	PE to PE Core	IOS	IOS XR	IPv4	IPv6
Yes	NA	Yes	Yes	Yes	NA	Yes	Yes	Yes

Unable to Troubleshoot MPLS TE Connectivity Problem

Router is running a non-OAM Cisco IOS version. The connectivity of the tunnel cannot be tested.

CE to CE	PE to Attached CE	CE to PE Across Core	PE to PE (in VRF)	PE to PE Core	IOS	IOS XR	IPv4	IPv6
Yes	NA	Yes	Yes	Yes	Yes	Yes	Yes	Yes

Unknown LSP Connectivity Problem

LSP connectivity problem, data plane issue, or unknown cause.

CE to CE	PE to Attached CE	CE to PE Across Core	PE to PE (in VRF)	PE to PE Core	IOS	IOS XR	IPv4	IPv6
Yes	NA	Yes	Yes	Yes	Yes	Yes	Yes	Yes

Unsupported IOS Version

Core router running an unsupported IOS version. This version of IOS does not support the required OAM functionality.

CE to CE	PE to Attached CE	CE to PE Across Core	PE to PE (in VRF)	PE to PE Core	IOS	IOS XR	IPv4	IPv6
Yes	NA	Yes	Yes	Yes	Yes	NA	Yes	NS

VPN Connectivity Tests have been Exercised and No Failures Found (However as Pings are Blocked to the CE, it is Not Possible to Verify VPN Connectivity)

Unable to verify VPN connectivity in VRF from PE to destination.

CE to CE	PE to Attached CE	CE to PE Across Core	PE to PE (in VRF)	PE to PE Core	IOS	IOS XR	IPv4	IPv6
Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes

Customer Site

Possible Customer Routing Issue

The PE has some routes from the CE, but the CE is unable to respond to pings.

CE to CE	PE to Attached CE	CE to PE Across Core	PE to PE (in VRF)	PE to PE Core	IOS	IOS XR	IPv4	IPv6
Yes	Yes	Yes	NA	NA	Yes	Yes	Yes	Yes

IOS XR Support

This section details the IOS XR support caveats:

1. MPLS Not Enabled On Interface, page 11-77

   IOS XR has the concept of packages. One of the packages relevant to Diagnostics is the MPLS package. When Diagnostics troubleshooting focuses on an IOS XR router in the core, various preparation checks are carried out to ensure that the router is configured sufficiently; that is, that the MPLS package is installed and active, and then to check for the required features being enabled (MPLS OAM and MPLS LDP). If any of these checks fail, a failure scenario is reported. MPLS Not Enabled On Interface, page 11-77 remains valid for IOS devices, or when MPLS is disabled on an interface on an IOS XR device.

2. CEF Not Enabled On Router, page 11-75

It is still possible for Diagnostics to determine cases where CEF is disabled on IOS routers in the core, for example, CEF could be disabled via CLI configuration, or when the router is overloaded and shuts itself down. In such cases, the IOS CLI command **show cef state** reports that CEF is either *enabled/running* or *disabled/not running*, and Diagnostics can determine that CEF is disabled.

CEF cannot be disabled on an IOS XR router. When the CEF switching feature on an IOS XR router becomes overloaded, it does not shut down. Instead, it applies back pressure to the queue of outstanding requests in an effort to reduce the load on the CEF switching process. Therefore, in an IOS XR router, the relevant CLI **show** command does not report that CEF is non-operational, and thus this failure scenario is not valid on an IOS XR router.

3. LDP/TDP Mismatch, page 11-76

   This is not a valid scenario where directly connected IOS XR routers are present, because IOS XR only supports a single label distribution protocol – that is LDP. It is possible for the application to find LDP-TDP mismatch if it is configured in IOS-IOS XR, IOS XR-IOS, or IOS-IOS configurations.

4. Broken LDP Neighbor Session, page 11-74

   Label Inconsistency, page 11-75

   LDP Label Mismatch, page 11-75

   LDP Neighbors Not Discovered, page 11-76

   LSP Reply Path Problem, page 11-76

Missing LFIB Entry, page 11-76

Missing or Untagged Return Path, page 11-77

No Entry for Label, page 11-77

No Valid Next-Hop Entry, page 11-78

These failure scenarios are due to specific bugs in IOS versions, and are not applicable to IOS XR.

5. ATM Interface Is Protocol Down, page 11-58

   ATM Sub-interface is Protocol Down, page 11-58

   Intermittent ATM Failure (to the ATM Next Hop), page 11-62

   Intermittent ATM Failure (to the Destination), page 11-62

   Static IP address on ATM Point-to-Point Interface, page 11-67

   Undiagnosed ATM Failure (ATM Pings Failed but the ATM Segment Ping Succeeded), page 11-68

   Undiagnosed ATM Failure (End to End and Segment ATM Pings Failed), page 11-68

   These failures are not supported in case of ATM interfaces on the CRS-1 platform. However, they continue to be applicable on IOS devices and for IOS XR on the Cisco 12000 XR Series.

IPv6 Support

This section details the IPv6 support caveats:

- In addition to IPv4 troubleshooting for both IOS and IOS XR devices, troubleshooting is extended for the IOS XR devices where IPv6 addressing is used on the PE-CE link. IPv6 is not supported on IOS devices.
- Ethernet is the only Access Circuit interface technology on which Diagnostics can troubleshoot, when IPv6 addressing is used on IOS XR devices.
- IPv6 support is extended only to support eBGP as the PE-CE routing protocol.
- Since the scope of the IPv6 address is only between the attachment circuit links, it is assumed that both ends of the LSP to have either IPv6 or IPv4 and not IPv6 at one end and IPv4 at the other and vice-versa.
- IPv6 support can troubleshoot PE-CE links configuration with Global Unicast IPv6 address alone.
- For IPv6 support, IPv4 router-id will be used for identification of Peer Routers, for protocols including BGP and LDP.
- Unlike in case of IPv4, the CE Access Circuit Interface IP address (for applicable test types) will not be auto populated as IPv6 unnumbered is not supported on IOS XR devices and there is no concept of /30 and /31 address in IPv6.
- The below failure scenarios although applicable on IOS XR, not applicable in IPv6 context as these validations are performed during initial data validations. The failure scenario to report that the CE access circuit interface IP address is a network address is performed during initial data validations. There is no concept of Broadcast address in IPv6.
  - Broadcast Address
  - Network Address

Observations

Observations are conditions that could lead to connectivity problems. Because Diagnostics cannot categorically conclude the cause of the connectivity problem, these conditions are reported as observations.

For more information, e-mail: mpls-diagnostics-expert@cisco.com

ACL Configured on PE

There is an access control list (ACL) configured on the provider edge (PE) router. It might be causing failure of the VPN routing/forwarding instance (VRF) ping from this PE to the remote PE, however we have not analyzed the ACL to confirm its usage. This is not causing the connectivity failure from the PE to the local customer edge (CE) router, or customer device.

BGP Neighbor Session Problem

Possible border gateway protocol (BGP) neighbor session problem detected. Displays table with columns BGP Neighbor (Neighbor IP Address) and BGP State (BGP Neighbor State).

BGP Router ID is Not a Loopback Interface

The local BGP router ID on the PE is not assigned to a loopback interface. It is recommended that the router ID is taken from a loopback interface to both reduce the chance of duplication and enhance stability.

Connected Routes Not Redistributed into MP-BGP

Directly connected routes might not be redistributed into MP-BGP.

Core Troubleshooting Could Not be Performed. The VPN Route is External.

Core troubleshooting could not be performed. Diagnostics is unable to determine the label-switched path (LSP) to test, because the PE <*PE Name*> has no valid VPN route to the remote prefix <*IP address*> within the VRF <*VRF name*>. The route is not learned through an internal Border Gateway Protocol (BGP) VPNv4 neighbor. It is known through the <*Routing Protocol Name*>. The next-hop for this external route is <*IP address*>. Traffic does not flow through the MPLS core, as expected. This might be an intentional back door link, however, it is often a symptom of PE - CE misrouting. To test LSP connectivity, you might want to run a PE to PE Core test that allows you to specify the LSP endpoints manually.

Core Troubleshooting Could Not be Performed. The VPN Route is External and the Next-Hop Is Inaccessible.

Core troubleshooting could not be performed. Diagnostics is unable to determine the LSP to test, because the PE <*PE Name*> has no valid virtual private network (VPN) route to the remote prefix <*IP address*> within the VRF <*IP address*>. The route is not learned through an internal BGP VPNv4 neighbor. It is known through the <*Routing Protocol Name*>. The next-hop for this external route is inaccessible. This might be an intentional back door link, however, it is often a symptom of PE - CE misrouting. To test LSP connectivity, you might want to run a PE to PE Core test that allows you to specify the LSP endpoints manually.

Core Troubleshooting Could Note be Performed. The VPN Route Next-Hop is Inaccessible.

Core troubleshooting could not be performed. Diagnostics is unable to determine the LSP to test, because the PE <*PE Name*> has no valid VPN route to the remote prefix <*IP address*> within the VRF <*VRF name*>. The next-hop is inaccessible. This might be due to a problem within the Core Interior Gateway Protocol (IGP) or IP connectivity failure. To test LSP connectivity, you might want to run a PE to PE Core test that allows you to specify the LSP endpoints manually.

Duplicate BGP Router ID

BGP Router Identifier on the PE is found to be duplicated on one or more interfaces of the listed devices.

eBGP Maximum Prefixes

The exterior border gateway protocol (eBGP) session between the PE and an eBGP neighbor has a maximum prefix count configured on the PE. There are currently *X* prefixes in the VRF from this neighbor.

eBGP Neighbor Not Established

It appears that you are running eBGP as your PE-CE routing protocol. The PE and CE interfaces are on different subnets and there is no route to the CE on the PE. Until there is a route to the CE, this eBGP session is not established.

eBGP Neighbors Not Established

eBGP neighbors have been specified in a VRF but are not established and are unreachable.

EIGRP Peer Relationship Not Established

The PE interface is configured with IP unnumbered. The CE interface must either also be using IP unnumbered or be on the same subnet in order for the enhanced interior gateway routing protocol (EIGRP) to establish a peer relationship.

Full-Mesh VPN Topology

These routers appear to be connected via a fully meshed VPN configuration. If this is not correct, there is an issue with the route target configuration.

Hub and Spoke VPN Topology

These routers appear to be connected via a hub and spoke VPN configuration. If this is not correct, there is an issue with the route target configuration.

Hub To Hub, Hub and Spoke VPN Topology

These routers appear to be connected via a hub to hub, hub and spoke VPN configuration. If this is not correct, there is an issue with the route target configuration.

Incomplete CEF Adjacencies

Incomplete Cisco express forwarding (CEF) adjacencies on the access circuit interface.

Incorrect Multilink Virtual-Access Interface Specified

If you are specifying a multilink access circuit interface for the PE ensure that the virtual access interface specified is an active multilink bundle interface and that it has active bundle links.

Interface Not In VLAN

Warning: Ethernet access circuit interface is not associated with a virtual LAN (VLAN).

Intermittent Ping Success

The ping showed only intermittent connectivity.

Inverse ARP Disabled on FR Interface

The Frame Relay interface is dynamically configured but has inverse address resolution protocol (ARP) explicitly disabled.

Inverse ARP Implicitly Disabled on FR Interface

There is a Frame Relay static map on the interface. This interface is configured dynamically but the presence of the static map will, as a side effect, disable inverse ARP.

LMI Disabled on Frame Relay Interface

Warning: Frame Relay permanent virtual circuit (PVC) status cannot be checked on interface because the local management interface (LMI) is disabled.

LSP Endpoint is Not a Loopback Interface

The VPNv4 route is being sent to IBGP neighbor(s). However, the next hop address is one of the directly connected physical interfaces. It is recommended to use loopback interfaces as the next hops for VPNv4 IBGP neighbors. If the address is not available at the correct hop via the IGP, it could break connectivity between VPN sites because no forwarding label information is available.

MPLS OAM Package is not enabled on IOS XR Router

MPLS OAM package is not enabled on the IOS XR router.

MPLS TE Package is not Enabled on IOS XR Router

MPLS TE package is not enabled on the IOS XR router.

Multiple Equal Cost Paths

Equal cost multiple paths (ECMP) were found.

Non-compliant IOS Version on PE Router

Core troubleshooting could not be performed because the provider edge (PE) router is running a non MPLS OAM compliant Cisco IOS version.

No Routes Received from eBGP

It appears that you are running eBGP as your PE-CE routing protocol. However, no routes have been received from the neighbor.

No Route to Remote Prefix Received from eBGP

It appears that you are running eBGP as your PE-CE routing protocol. However, the route to a remote prefix has not been received from the neighbor. Check PE and customer edge (CE) BGP configuration.

No VPN Label in VRF for Prefix

No virtual private network (VPN) label was found for the address in the VPN routing/forwarding (VRF) on the device.

OSPF Peer Relationship Not Established

The PE interface is configured with IP unnumbered. The CE interface must either also be using IP unnumbered or be on the same subnet in order for open shortest path first (OSPF) to establish a peer relationship.

PE-PE Core Only Test Performed and the Optional Loopback IP Address Parameters Have Not Been Supplied

The LSP under test was selected based on the BGP router-id of the remote site PE. If the network has multiple LSPs between the two PEs, the reported result might not accurately reflect the state of the LSP used for customer traffic. To ensure the correct LSP is tested, you can supply the LSP endpoints on the test input window.

Possible Backup Link

The ping from the PE to the destination prefix succeeded, however the route from the PE to the destination prefix has not been learned via the expected PE interface. There might be a backup link in operation, or you might have input the incorrect parameters.

Possible Blocking Route Map

A route map is configured on the PE which might be causing route traffic to be lost. If this is an intranet/extranet VPN configuration, then there might be a route map configuration error.

Possible Core IP Failure

The internet control message protocol (ICMP) ping issued from the local PE to the remote PE failed. There is no route to the remote PE in the Interior Gateway Protocol (IGP) route table of the local PE. Try troubleshooting IP connectivity between these devices.

Possible Ethernet Duplex Mismatch

Warning: Access circuit interface has late collisions. This might be caused by an Ethernet duplex mismatch.

Route Limit Reached

The route count on the device has reached the route limit.

Traceroute Not Transmitted

The MPLS traceroute was not transmitted.

This chapter provides details of all IOS and IOS XR commands executed by the troubleshooting workflow in the Diagnostics application for the Cisco Prime Fulfillment 6.2 release.

IOS Commands

This section lists the IOS commands used by Diagnostics. If TACACS+ (or another authentication/authorization system) is used, ensure that these are all allowed for Diagnostics.

Note This list might be updated when Diagnostics releases or patches are made available, e-mail: mpls-diagnostics-expert@cisco.com for the latest list.

1. attach <*slot*> show version
2. execute-on slot <*slot*> 'show mpls forwarding-table <*destinationPrefix*> <*subnetMask*>'
3. execute-on slot <*slot*> 'show mpls forwarding-table <*destinationPrefix*>'
4. execute-on slot <*slot*> 'show mpls forwarding-table vrf <*vrfName*> <*destinationPrefix*> <*subnetMask*>'
5. execute-on slot <*slot*> 'show mpls forwarding-table vrf <*vrfName*> <*destinationPrefix*>'
6. execute-on slot <*slot*> 'show mpls forwarding-table vrf <*vrfName*>'
7. execute-on slot <*slot*> 'show mpls forwarding-table'
8. execute-on slot <*slot*> show ip cef vrf <*vrfName*> <*networkPrefix*>
9. execute-on slot <*slot*> show ip cef vrf <*vrfName*> <*networkPrefix*> <*subnetMask*>
10. execute-on slot <*slot*> show version

11. ping (interactive)
12. ping <*targetIp*>
13. ping mpls ipv4 <*targetIp*>/<*targetIpSubnetMask*> source <*source*> sweep <*minSweepSize*> <*maxSweepSize*> <*sweepInterval*> <*repeatCount*> timeout <*timeout*> replyMode <*replyMode*>
14. ping mpls traffic-eng Tunnel <*tunnelNumber*>
15. ping vrf <*vrfName*> (interactive)
16. show access-lists <*listName*>
17. show atm map
18. show atm pvc <*interface*>
19. show cef drop
20. show cef drop | include ^<*slot*>
21. show frame-relay lmi
22. show frame-relay lmi interface <*interface*>
23. show frame-relay map
24. show frame-relay pvc <*interface*> dlci <*dlci*>
25. show interfaces <*interface*>
26. show ip bgp summary
27. show ip bgp vpnv4 <*vrfName*> rib-failure
28. show ip bgp vpnv4 all neighbors
29. show ip bgp vpnv4 all neighbors <*destIp*>
30. show ip bgp vpnv4 all | include local router
31. show ip bgp vpnv4 vrf <*vrfName*> <*networkPrefix*>
32. show ip bgp vpnv4 vrf <*vrfName*> neighbors <*destIp*>
33. show ip bgp vpnv4 vrf <*vrfName*> <*prefix*> <*subnetMask*>
34. show ip bgp vpnv4 vrf <*vrfName*> labels | include <*networkPrefix*>/<*subnetMask*> | [0-9]+\.[0-9]+\.[0-9]+\.[0-9]+
35. show ip bgp vpnv4 vrf <*vrfName*> labels | include <*classfulPrefix*>
36. show ip bgp vpnv4 vrf <*vrfName*> labels | include <*networkPrefix*>/<*subnetMask*>
37. show ip cef <*destinationPrefix*>
38. show ip cef summary
39. show ip cef vrf <*vrfName*> <*networkPrefix*> <*subnetMask*> detail
40. show ip cef vrf <*vrfName*> <*networkPrefix*> detail
41. show ip cef vrf <*vrfName*> adjacency <*interface*> <*destip*> detail
42. show ip eigrp <*vrfName*> interfaces <*vrfInterface*>
43. show ip interface <*interface*>
44. show ip interface <*interface*> | include access list is
45. show ip interface brief <*interface*>
46. show ip interface brief | include <*ip-address*>

47. show ip ospf <*processId*> <*area*> interface <*intName*>
48. show ip ospf mpls traffic-eng link
49. show ip protocols <*vrfName*>
50. show ip route <*targetIp*>
51. show ip route vrf <*vrfName*> <*targetIp*>
52. show ip traffic
53. show ip vrf detail <*vrfName*>
54. show ip vrf interfaces <*vrfName*>
55. show mpls forwarding-table <*destinationPrefix*>
56. show mpls forwarding-table <*destinationPrefix*> <*subnetMask*>
57. show mpls forwarding-table <*destinationPrefix*> detail
58. show mpls forwarding-table labels <*label*>
59. show mpls forwarding-table labels <*label*> detail
60. show mpls forwarding-table vrf <*vrfName*>
61. show mpls forwarding-table vrf <*vrfName*> <*destinationPrefix*>
62. show mpls forwarding-table
63. show mpls interfaces <*interface*>
64. show mpls interfaces all
65. show mpls ip binding <*destinationPrefix*> <*destinationMask*>
66. show mpls ip binding local
67. show mpls ip binding summary
68. show mpls label range
69. show mpls ldp bindings <*ip*> <*subnetMask*>
70. show mpls ldp bindings neighbor <*neighbor ip*> <*subnetMask*>
71. show mpls ldp discovery
72. show mpls ldp neighbor
73. show mpls ldp neighbor <*interface*>
74. show mpls traffic-eng tunnels
75. show mpls traffic-eng tunnels <*status*>
76. show mpls traffic-eng tunnels <*tunnelId*>
77. show mpls traffic-eng tunnels destination <*destination*> <*status*>
78. show mpls traffic-eng tunnels destination <*destination*>
79. show mpls traffic-eng tunnels role <*role*>
80. show mpls traffic-eng tunnels role <*role*> <*status*>
81. show mpls traffic-eng tunnels role <*role*> destination <*destination*> <*status*>
82. show mpls traffic-eng tunnels role <*role*> destination <*destination*> up
83. show mpls traffic-eng tunnels role head brief
84. show ppp multilink interface <*interface*>

85. show route-map <*mapName*>
86. show running-config
87. show running-config interface <*interface*>
88. show running-config interface <*interface*> | include frame-relay interface-dlci
89. show running-config interface <*interface*> | include map-group
90. show running-config interface <*interface*> | include no frame-relay inverse-arp
91. show running-config | begin router bgp
92. show running-config | include advertise-
93. show running-config | include ldp password
94. show running-config | include mpls label protocol
95. show running-config | include no
96. show version
97. show vlans
98. traceroute mpls ipv4 <*ipAddress*>/<*subnetMask*> source <*source*> destination <*destination*> ttl 15
99. traceroute mpls traffic-eng Tunnel <*tunnelNumber*>
100. traceroute vrf <*vrfName*> (interactive)

IOS XR Commands

This section lists the IOS XR commands used by Diagnostics. If TACACS+ (or another authentication/authorization system) is used, ensure that these are all allowed for Diagnostics.

Note This list might be updated when Diagnostics releases or patches are made available, e-mail: mpls-diagnostics-expert@cisco.com for the latest list.

1. ping <*targetIp*>
2. ping atm interface <*interface*> <*vpi*>/<*vci*>
3. ping atm interface <*interface*> <*vpi*>/<*vci*> end-loopback
4. ping atm interface <*interface*> <*vpi*>/<*vci*> seg-loopback
5. ping mpls ipv4 <*destination*>/<*subnetMask*>
6. ping mpls ipv4 <*destination*>/<*subnetMask*> reply mode router-alert
7. ping mpls ipv4 <*destination*>/<*subnetMask*> source <*source*>
8. ping mpls traffic-eng Tunnel <*tunnelId*>
9. ping vrf <*vrfName*>
10. ping vrf <*vrfName*> <*targetIp*> <*sourceInterface*> <*minSweepSize*> <*maxSweepSize*> <*sweepInterval*>
11. show access-lists ipv4 <*listName*>
12. show bgp ipv4 all summary

13. show bgp vpnv4 unicast neighbors
14. show bgp vpnv4 unicast summary
15. 14.show bgp vpnv4 unicast vrf <*vrfName*> <*networkPrefix*>
16. show bgp vpnv4 unicast vrf <*vrfName*> <*prefix*> <*mask*>
17. show bgp vpnv4 unicast vrf <*vrfName*> labels
18. show bgp vrf <*vrfName*> advertised neighbor <*neighboreId*> summary | include <*ceDeviceIpAddr*>
19. show bgp vrf <*vrfName*> ipv4 unicast
20. show bgp vrf <*vrfName*> neighbors
21. show bgp vrf <*vrfName*> vpnv4 unicast neighbors
22. show cef ipv4 <*destinationPrefix*>
23. show cef ipv4 drops
24. show cef ipv4 drops location <*slot*>
25. show cef ipv4 summary
26. show cef vrf <*vrfName*> ipv4 <*networkPrefix*> detail
27. show cef vrf <*vrfName*> ipv4 <*networkPrefix*> <*subnetMask*> detail
28. show cef vrf <*vrfName*> <*networkPrefix*> <*subnetMask*> location <*location*>
29. show eigrp <*vrfName*> interfaces <*vrfInterface*>
30. show frame-relay lmi
31. show frame-relay lmi interface <*interface*>
32. show install active summary
33. show install inactive summary
34. show install location <*slot*>
35. show interfaces <*interface*>
36. show ip cef vrf <*vrfName*> adjacency <*interface*> <*destip*> detail
37. show ip ospf <*processId*> <*area*> interface <*intName*>
38. show ipv4 interface <*interface*>
39. show ipv4 interface brief <*interface*>
40. show ipv4 interface brief | include <*ip-address*>
41. show ipv4 traffic
42. show ipv4 vrf <*vrfName*> interface brief
43. show ipv4 vrf <*vrfName*> interface <*interface*>
44. show ipv4 vrf all interface brief
45. show mpls forwarding
46. show mpls forwarding labels <*label*>
47. show mpls forwarding prefix <*destinationPrefix*>/<*subnetMask*>
48. show mpls forwarding prefix <*destinationPrefix*>/<*subnetMask*> detail
49. show mpls forwarding vrf <*vrf*>

50. show mpls forwarding vrf <*vrf*> prefix <*destinationPrefix*>/<*subnetMask*>
51. show mpls forwarding vrf <*vrfName*> prefix <*destinationPrefix*>/<*subnetMask*> labels <*label*> location <*location*>
52. show mpls forwarding vrf <*vrfName*> prefix <*destinationPrefix*>/<*subnetMask*> location <*location*>
53. show mpls interfaces
54. show mpls interfaces <*interface*>
55. show mpls label range
56. show mpls label table summary
57. show mpls ldp bindings <*ip*> <*mask*>
58. show mpls ldp bindings neighbor <*neighbor*> <*ip*> <*mask*>
59. show mpls ldp discovery
60. show mpls ldp neighbor
61. show mpls ldp neighbor <*interface*>
62. show mpls traffic-eng tunnels
63. show mpls traffic-eng tunnels backup <*tunnelId*>
64. show mpls traffic-eng tunnels brief role head
65. show mpls traffic-eng tunnels <*status*> detail
66. show mpls traffic-eng tunnels <*tunnel-id*>
67. show mpls traffic-eng tunnels <*tunnelNumber*> detail
68. show mpls traffic-eng tunnels destination <*destination*>
69. show mpls traffic-eng tunnels name <*name*>
70. show mpls traffic-eng tunnels destination <*destination*> <*status*> detail
71. show mpls traffic-eng tunnels destination <*destination*> detail
72. show mpls traffic-eng tunnels detail
73. show mpls traffic-eng tunnels role <*role*> <*status*> detail
74. show mpls traffic-eng tunnels role <*role*> destination <*destination*> <*status*> detail
75. show mpls traffic-eng tunnels role <*role*> destination <*destination*> up detail
76. show mpls traffic-eng tunnels role <*role*> detail
77. show ospf
78. show ospf vrf <*vrf*>
79. show ospf border-routers | include ABR
80. show ospf | include ID
81. show ospf mpls traffic-eng link
82. show ospf vrf <*vrfName*> interface brief
83. show ospf vrf <*vrfName*> interface <*interfaceName*>
84. show protocols | include OSPF
85. show rib ipv4 tables

86. show rib vrf <*vrf*> ipv4 unicast statistics <*protocolName*>
87. show rib vrf <*vrf*> protocols
88. show rip vrf <*vrf*>
89. show route ipv4 <*targetIp*>
90. show route vrf <*vrfName*> ipv4 <*targetIp*>
91. show rpl route-policy <*mapName*>
92. show rsvp neighbors
93. show running-config
94. show running-config explicit-path name <*explicitPathName*>
95. show running-config interface <*interface*>
96. show running-config mpls ldp
97. show running-config mpls ldp label advertise
98. show running-config mpls traffic-eng
99. show running-config router bgp
100. show running-config router bgp <*asNumber*> vrf <*vrfName*> neighbor <*neighborIpAddr*>
101. show running-config router bgp <*asNumber*> neighbor-group <*neighborGroupName*>
102. show running-config router bgp | include redistribute <*protocol*>
103. show running-config router ospf
104. show running-config router <*protocol ID*> vrf <*vrf*>
105. show running-config rsvp interface <*interface-name*>
106. show vlan interface
107. show version
108. show vrf <*vrfName*> ipv4 detail
109. traceroute mpls ipv4 <*destination*>/<*subnetMask*>
110. traceroute mpls traffic-eng Tunnel <*tunnelId*>
111. traceroute vrf <*vrf*>
112. ping vrf <*vrfName*> <*targetIpv6Address*> <*sourceInterface*> <*minSweepSize*> <*maxSweepSize*> <*sweepInterval*>
113. show bgp vpnv6 unicast neighbors
114. show bgp vpnv6 unicast neighbors <*destIp*>
115. show bgp vpnv6 unicast vrf <*vrfName*> <*networkPrefix*>
116. show bgp vpnv6 unicast vrf <*vrfName*> <*networkPrefix*>/<*subnetMask*>
117. show bgp vpnv6 unicast vrf <*vrfName*> labels | include <*networkPrefix*>/<*subnetMask*>| [0-9A-Fa-f:]+[0-9A-Fa-f]*
118. show bgp vpnv6 unicast summary | include BGP router identifier
119. show bgp vrf <*vrfName*> ipv6 unicast
120. show bgp vrf <*vrfName*> ipv6 unicast advertised neighbor <*neighborId*> summary | include <*ceDeviceIpAddr*>

121. show cef ipv6 summary
122. show cef vrf <*vrfName*> ipv6 <*networkPrefix*> detail
123. show cef vrf <*vrfName*> ipv6 <*networkPrefix*>/<*subnetMask*> detail
124. show cef vrf <*vrfName*> ipv6 <*networkPrefix*> location <*location*>
125. show cef vrf <*vrfName*> ipv6 <*networkPrefix*>/<*subnetMask*> location <*location*>
126. show ipv6 interface <*interface*>
127. show ipv6 interface brief <*interface*>
128. show ipv6 vrf all interface brief
129. show ipv6 vrf <*vrfName*> interface brief
130. show ipv6 vrf <*vrfName*> interface <*interface*>
131. show rib ipv6 tables
132. show route ipv6 <*targetIp*>
133. show route vrf <*vrfName*> ipv6 <*targetIp*>
134. show vrf <*vrfName*> ipv6 detail

CHAPTER 12

Using the Topology Tool

This chapter explains about how topology tool provides a graphical view of networks set up through the Cisco Prime Fulfillment web client. It gives a graphical representation of the various physical and logical parts of the network, both devices and links. It contains the following sections:

Introduction

The topology tool includes three types of views:

- VPN view—shows connectivity between customer devices. The VPN view also gives an aggregate view of all services and individual logical and physical views of each of the services.
- Logical view—shows logical connections set up in a selected provider region
- Physical view—displays connectivity of named physical circuits in a provider region.

In addition, this chapter describes the following features:

- Filtering and Searching—filter out unnecessary detail in large graphs or jump straight to a particular device using the search tool
- Using Maps—associate maps with the individual views.

Please note that some details, such as window decorations, are system specific and might appear differently in different environments. However, the functionality should remain consistent.

Launching Topology Tool

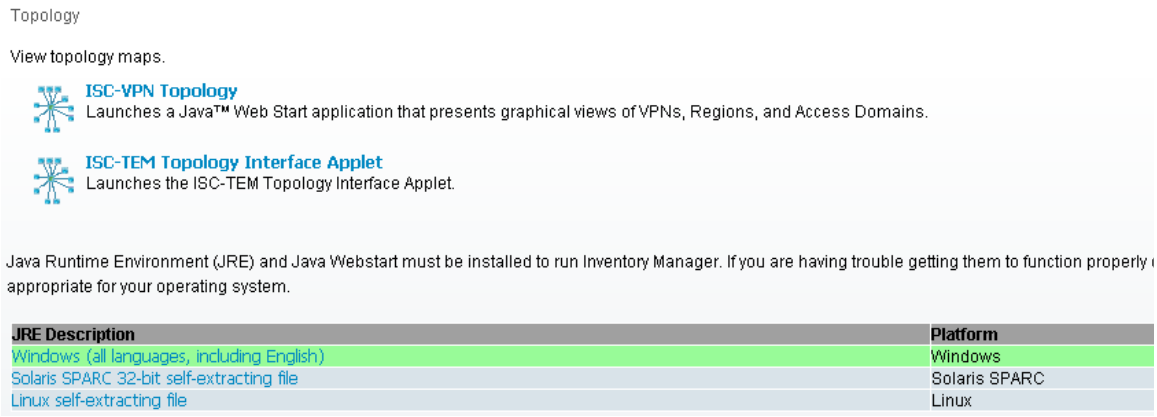
To launch the Topology Tool, follow these steps:

Step 1 Log into Prime Fulfillment.

Step 2 Choose **Inventory > Logical Inventory > Topology** and a window appears, as shown in Figure 12-1.

If you do not have the proper Java Runtime Environment (JRE) as specified at the bottom of the window, click the corresponding link for your system, follow that path, then quit the browser, log in again, and go back to the Topology Tool page.

Figure 12-1 Topology Launch Window



Step 3 Click **Prime Fulfillment-VPN Topology** in Figure 12-1, to launch the Topology Tool application on the web client.

This starts up the Java Web Start application.

Note Name resolution is required. The Prime Fulfillment HTTP server host must be in the Domain Name System (DNS) that the web client is using or the name and address of the Prime Fulfillment server must be in the client host file.

Step 4 The first time Inventory Manager is activated, a Security Warning window appears. Click **Start** to proceed or **Details** to verify the security certificate, and the Desktop Integration window appears.

Step 5 Click **Yes** to integrate into your desktop environment, click **No** to decline, click **Ask Later** to be prompted the next time VPN Topology is invoked, or click **Configure ...** to customize the desktop integration.

The Login window appears whether or not a selection has been made in the Desktop Integration window.

Step 6 Enter your **User Name** and **Password** and click **OK**.

The Topology Tool launches and connects to the Master Prime Fulfillment server.

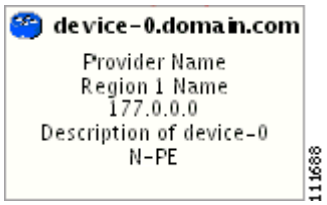
Conventions

Topology software uses several conventions to visually communicate information about displayed objects. The shape and color of a node representing a device depends on the role of the device, as shown in Table 12-1.

Table 12-1 Device Role Icons

Shape	Description
	Green icon for a CAT OS customer device followed by the following information: - Device name - Customer Name - Site Name - Management IP Address - Description - Role (SPOKE or HUB of a VPN)
	Green icon for a router customer device followed by the following information: - Device name - Customer Name - Site Name - Management IP Address - Description - Role (SPOKE or HUB of a VPN)
	Green icon for an interface followed by the following information: - Interface name - Management IP Address - Encapsulation Type - Interface Type
	Blue icon for a CAT OS provider device followed by the following information: - Device name - Provider Name - Region Name - Management IP Address - Description - Role

Table 12-1 Device Role Icons (continued)

Shape	Description
	Blue icon for a router provider device followed by the following information: - Device name - Provider Name - Region Name - Management IP Address - Description - Role
	Blue icon for a region followed by the following information: - Region name - Provider Name
	Green icon for a site followed by the following information: - Site name - Customer Name - Role in which Site's device joined VPN (HUB, SPOKE, or combination of HUB and SPOKE)
	Green icon for a site followed by the following information: - Site name - Customer Name - Role in which Site's device joined VPN (HUB, SPOKE, or combination of HUB and SPOKE)

A distinct color scheme is used to highlight the link type as shown in Table 12-2:

Table 12-2 Link Type Color Scheme

Color	Connection Type
(green)	End-to-end wire

Table 12-2 Link Type Color Scheme (continued)

Color	Connection Type
(purple)	Attachment circuit
(brown)	MPLS VPN link

Finally, the four patterns shown in Table 12-3 are used to indicate the service request state:

Table 12-3 Link State Pattern Scheme

Pattern	Service Request State
	Deployed, functional, pending
	Failed audit, invalid, broken, lost
	Wait deploy, requested, failed deploy
	Closed

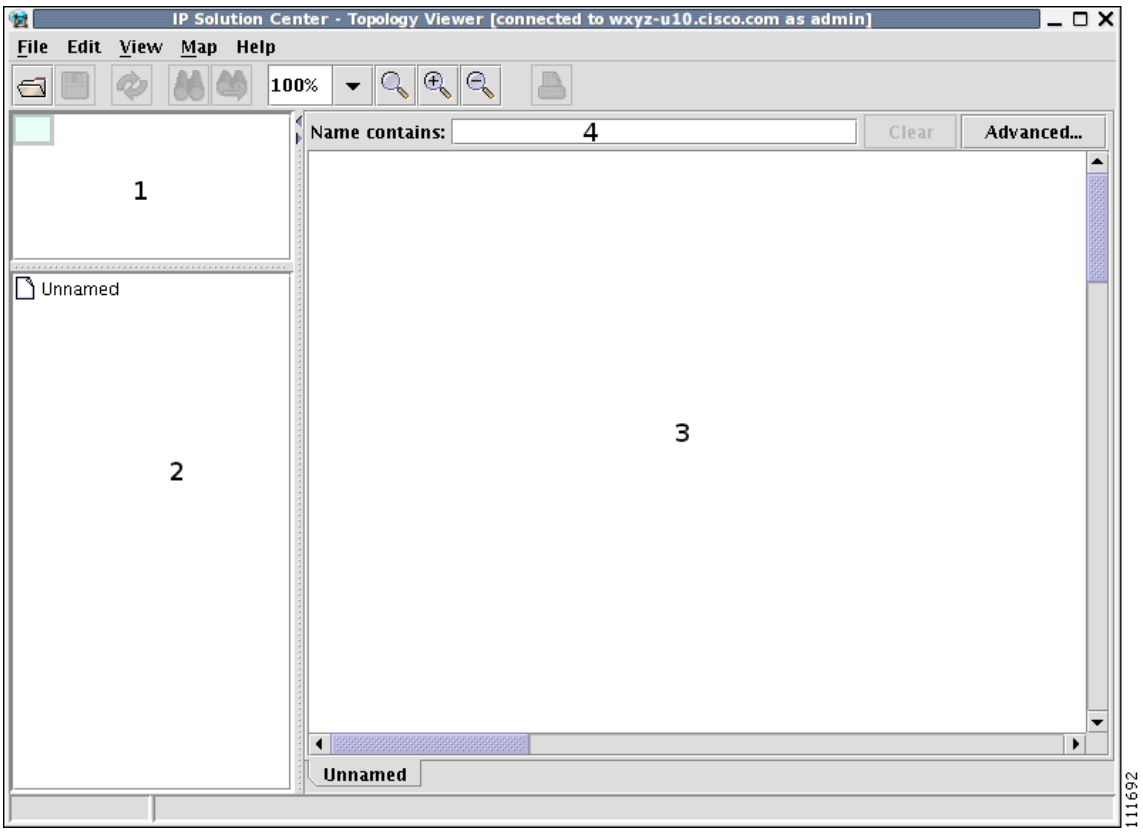
Accessing the Topology Tool for Prime Fulfillment-VPN Topology

Launch the Topology Tool as explained in Figure 12-1, "Topology Launch Window," in the "Launching Topology Tool" section on page 12-2 and then use the following steps to access the **Prime Fulfillment-VPN Topology** tool.

Step 1 Choose **Inventory > Logical Inventory > Topology > Prime Fulfillment-VPN Topology**.

The Topology window shown in Figure 12-2 appears.

Figure 12-2 Topology Application Window



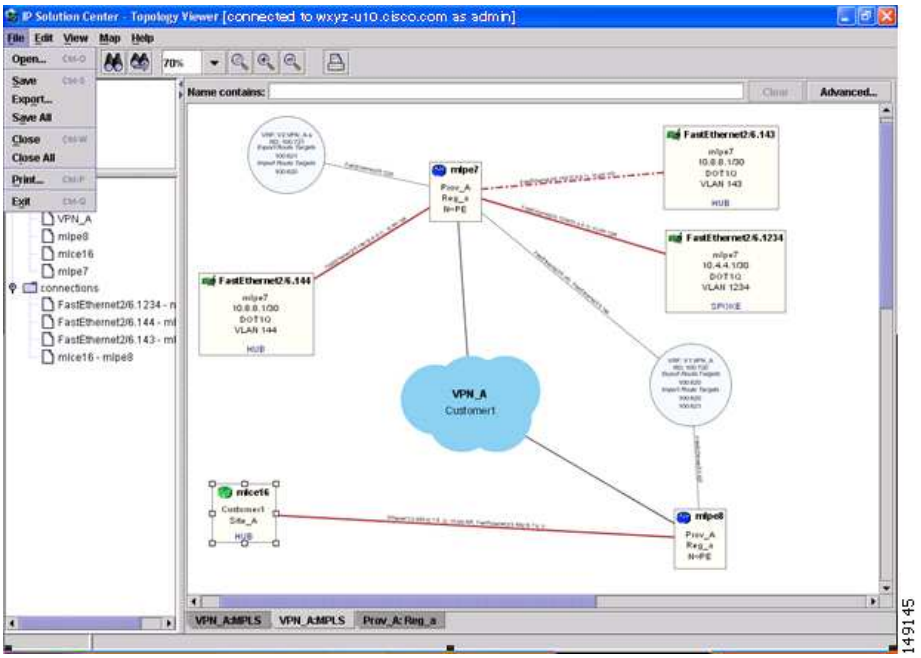
The application window is divided into four areas, as shown in Figure 12-2:

- area (1)—The top left corner shows the Overview area. The colored rectangular panel, called the panner, corresponds to the area currently visible in the main area. Moving the panner around changes the part of the graph showing in the main area. This is particularly useful for large graphs.
- area (2)—The bottom left area shows the Tree View of the graph. When no graph is shown, a single node called **Unnamed** is displayed. When a graph is shown, a tree depicting devices and their possible interfaces and connections is displayed. The tree can be used to quickly locate a device or a connection.
- area (3)—The main area (Main View) of the window shows a graph representing connections between devices. The name of the displayed network is shown at the bottom. When no view is present, the name defaults to **Unnamed**.
- area (4)—Above the main window is the Filter area. It allows you to filter nodes by entering a pattern. Nodes whose name contains the entered pattern maintain the normal level of brightness. All other nodes and edges become dimmed, as shown in Figure 12-14 and the "Filtering" section on page 12-20.

Note The bottom bar below all the areas, is a Status bar.

Views are loaded, saved, and closed using the **File** menu, as shown in Figure 12-3.

Figure 12-3 The File Menu



The **File** menu contains the following menu items:

- **Open**—Opens a view.
- **Save**—Saves the open and active view with the existing file name, if any.
- **Export**—Exports the active view in either Scalable Vector Graphics (SVG), Joint Photographics Expert Group (JPG), or Portable Network Graphics (PNG) format.
- **Save All**—Saves all open views.
- **Close**—Closes the open and active view.
- **Close All**—Closes all open views.
- **Print**—Prints the open and active view.
- **Exit**— Exits the Topology tool.

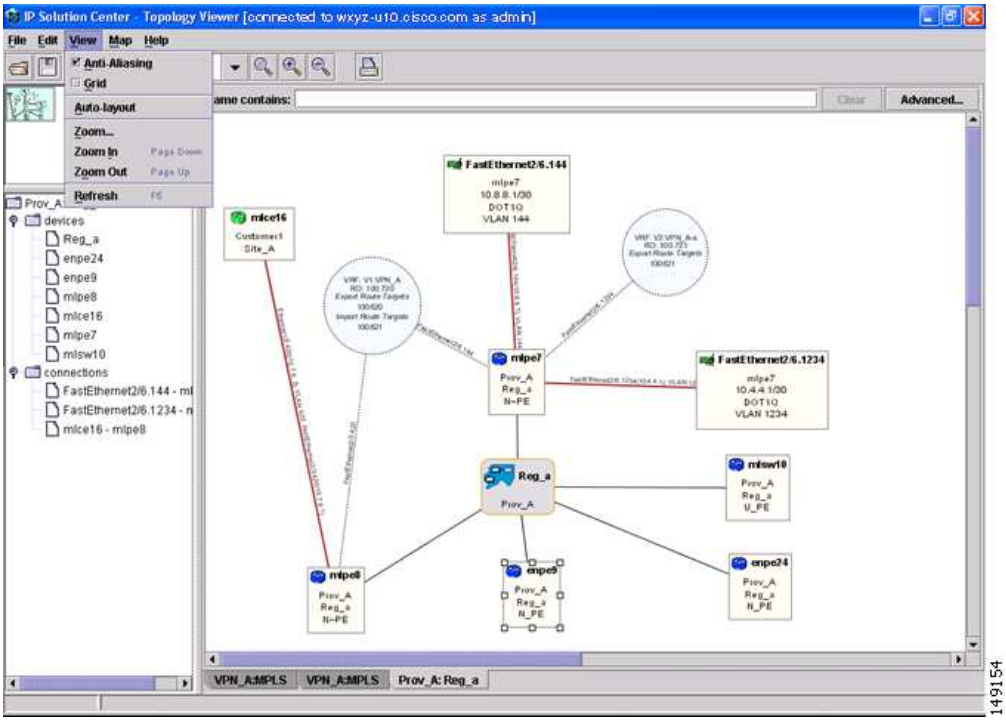
Types of Views

There are three view panes in the topology application and they are described in the following sections:

- VPN View, page 12-8, shows connectivity between devices in a VPN
- Logical View, page 12-13, shows connectivity between PEs and CPEs in a region
- Physical View, page 12-15, shows physical devices and links for PEs in a region.

The view attributes can be changed using the **View** menu, as shown in Figure 12-4.

Figure 12-4 The View Menu



The **View** menu contains the following menu items:

- **Anti-Aliasing**—When drawing a view, this creates smoother lines and a more pleasant appearance at the expense of performance.
- **Grid**—Activates a magnetic grid. The grid has a 10 by 10 spacing and can be used to help align nodes in a view.
- **Auto-Layout**—Generates an automatic layout of nodes in a view. If selected, the program tries to find the most presentable arrangement of nodes.
- **Zoom**—Opens a window where the desired magnification level can be specified.
- **Zoom In**— Increases the magnification level.
- **Zoom Out—**Decreases the magnification level.
- **Refresh**—Regenerates the view. This is especially useful if the data in the repository changes. To see an updated view, select **Refresh** or click the Refresh toolbar button.

VPN View

The VPN view shows connectivity between devices forming a given VPN. To activate the VPN view, follow these steps:

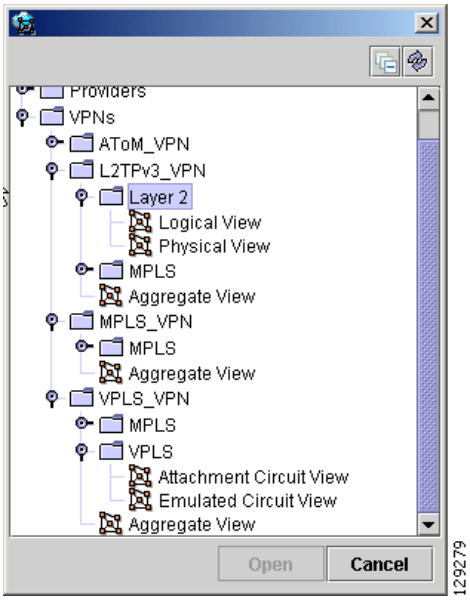
Step 1 In the menu bar, choose **File** > **Open**.

or

click the **Open** button in the tool bar.

The Folder View window in Figure 12-5 appears displaying a directory tree with available VPNs.

Figure 12-5 Folder View Window



Step 2 Choose the desired VPN's folder, select the folder, and click **Open**.

This opens the desired folder to display any logical and physical views associated with that VPN.

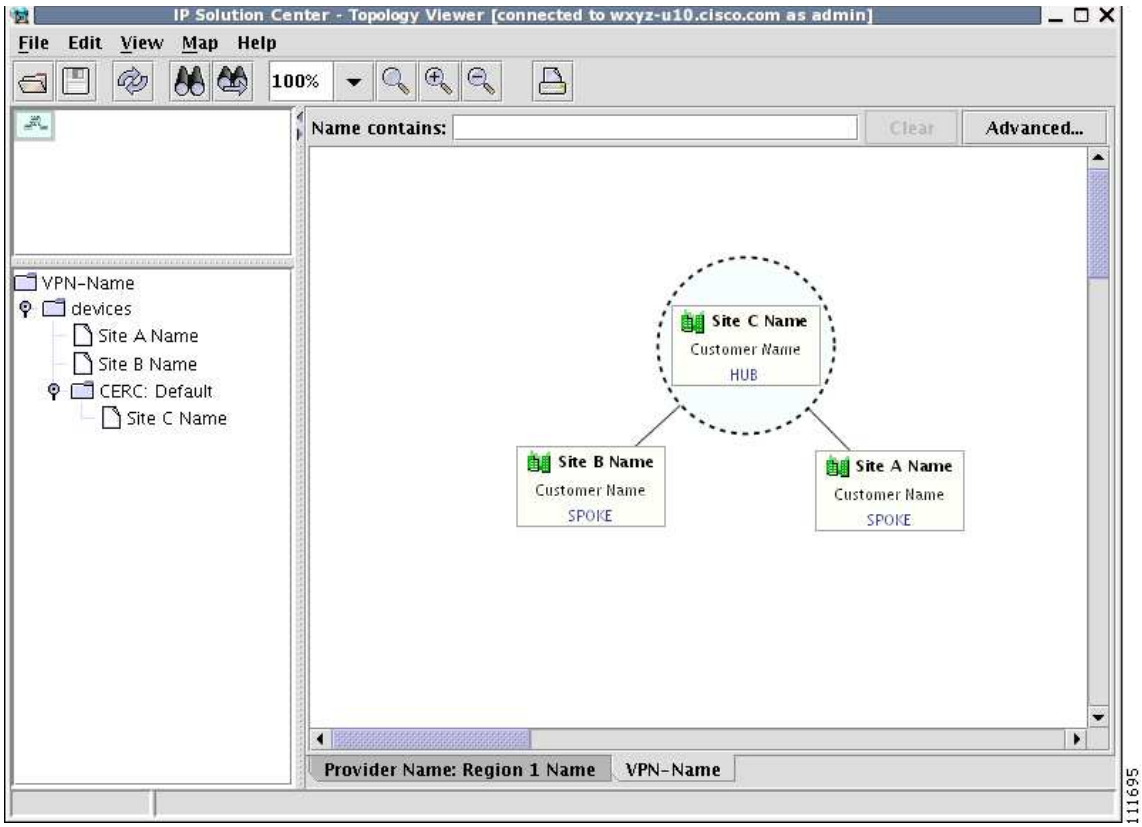
Click a logical or a physical view item in the folder tree. The logical view minimizes the amount of detail and shows connectivity between customer devices. The physical view reveals more about the physical structure of the VPN. For example, for MPLS it shows connectivity between customer and provider devices and the core of the provider.

Aggregate View

The Aggregate View, as shown in Figure 12-6, shows connectivity between all customer devices, regardless of the type of technology used to connect them.

A single view might show a combination of MPLS, Layer 2, and VPLS. For MPLS, only the Customer Premises Equipment devices (CPEs) are shown.

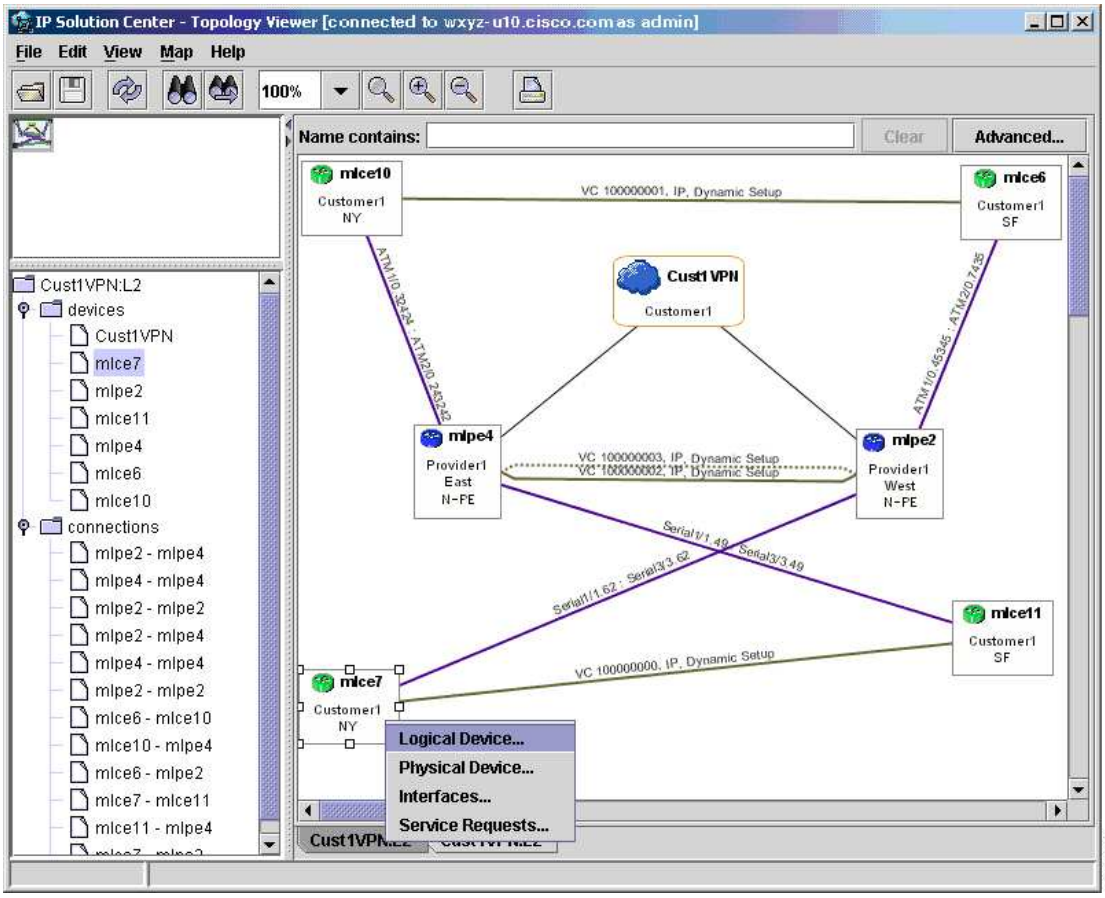
Figure 12-6 Aggregate View



The Layer 2 VPN might in addition to CPEs show connectivity between Customer Location Edge devices (CLEs) or Provider Edge devices (PE). For VPLS, you see connectivity between CPEs. For missing CPEs, you see connectivity to PEs.

In MPLS Layer 2 VPN, the topology displays Virtual Circuit (VC) with MPLS core (as MPLS string) but with L2TPv3, the topology will display Virtual Circuit (VC) with IP core (as IP string) as shown in Figure 12-7.

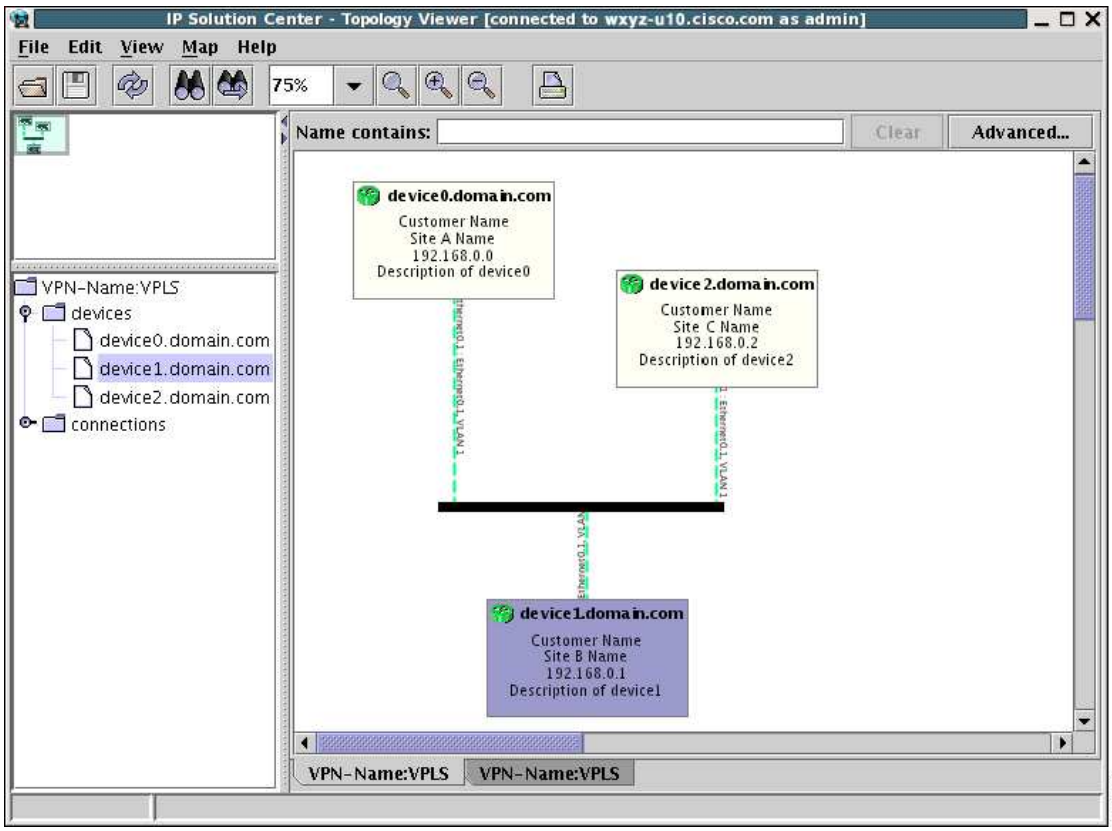
Figure 12-7 Virtual Circuit with IP Core



VPLS Topology

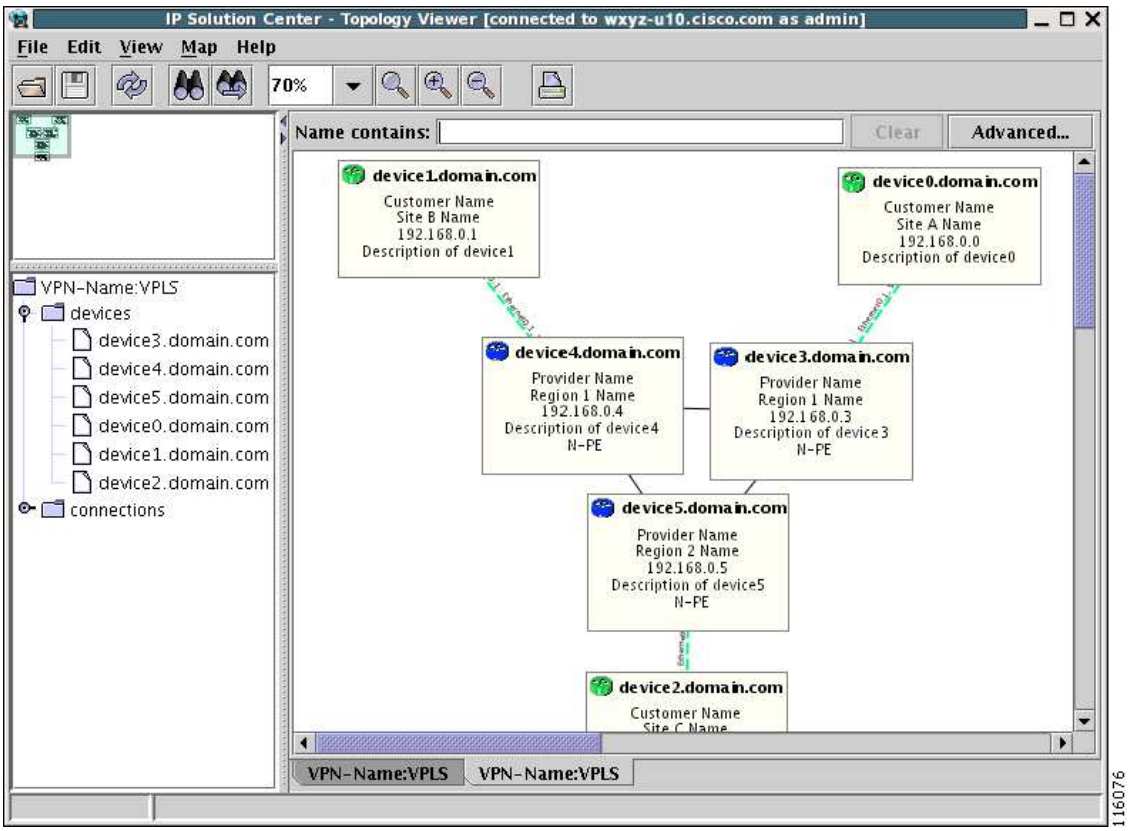
In the case of a VPLS topology, you can access an Attachment Circuit View or an Emulated Circuit View. The Attachment Circuit View corresponds to a logical view in other types of VPNs. It shows customer devices connected to a virtual private LAN, as shown in Figure 12-8.

Figure 12-8 Attachment Circuit View



The Emulated Circuit View shows the physical connectivity details omitted in the Attachment Circuit View. It shows connectivity between provider devices and customer devices connected to provider devices, as shown in Figure 12-9.

Figure 12-9 Emulated Circuit View



Logical View

The logical view shows connectivity, created through service requests, between PEs and CPEs of a given region.

To activate the logical view, follow these steps:

Step 1 In the menu bar, choose **File > Open**.

or

click the **Open** button in the tool bar.

The Folder View window, as shown in Figure 12-5, appears.

Step 2 Choose the desired VPN's folder and double-click on the desired folder.

Any logical and physical views associated with that VPN are displayed.

Step 3 To open the logical view for the selected VPN, do one of the following:

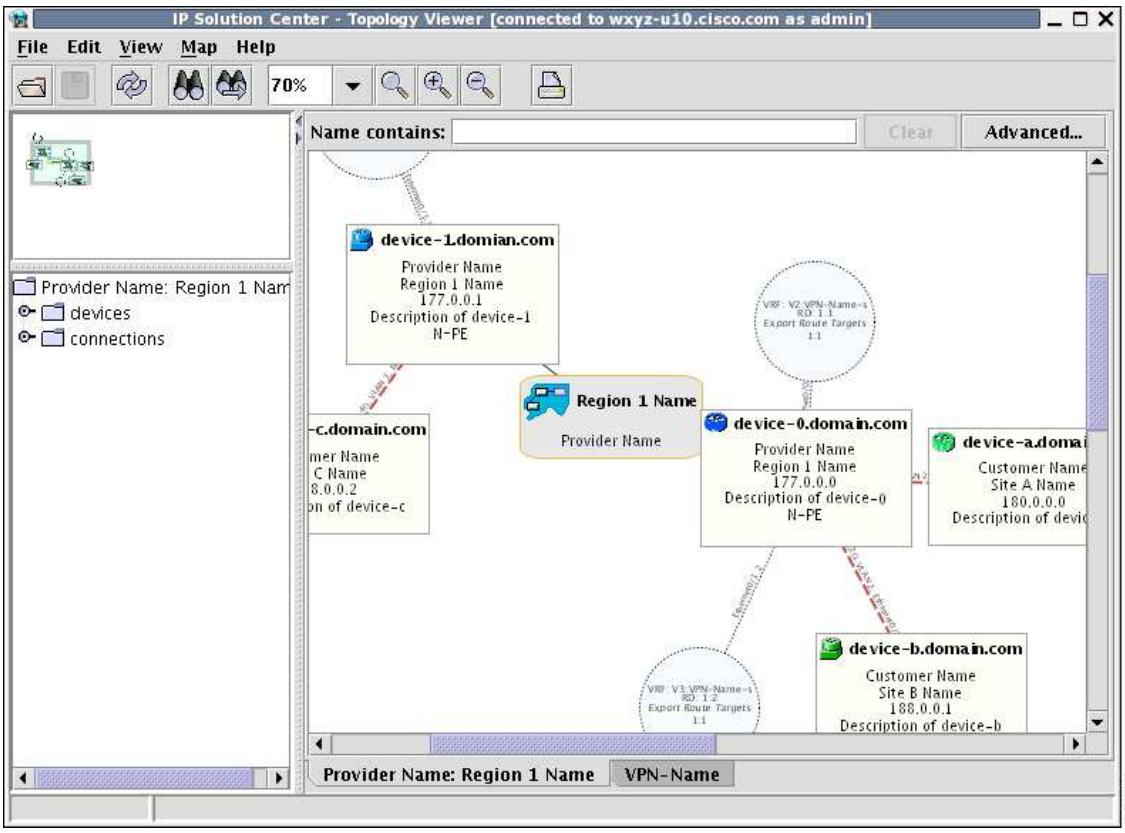
Single-click the **Logical View** icon and click **Open**

or

Double-click the **Logical View** icon.

This creates a logical view for the chosen VPN, as shown in Figure 12-10.

Figure 12-10 Logical View



In a created view, the node, usually located in the center of the graph, is the node representing a given region of a provider. The node is annotated with the name of the region and the name of the provider.

Each node directly connected to the regional node represents a PE. The icon of a node depends on the type and the role of the device it represents (see the "Conventions" section on page 12-3).

Each PE is annotated with the fully-qualified device name, provider name, region name, management IP address, description, and role. A right-click on a node displays the details of the logical and physical device, interfaces, and service requests (SR) associated with the node. For the regional node, details are shown in a tabulated form.

The various node and link properties are described in detail in Viewing Device and Link Properties, page 12-17.

Likewise, you can right-click on a link to learn about its link properties. For example, when selecting **Interfaces...** for a sample serial link, a Properties window appears.

Each PE can be logically connected to one or more CPEs. Such connections are created by either MPLS VPN links or Layer 2 Logical Links. Each such connection is represented by an edge linking the given PE to a CPE. If there are more connections between a particular PE and CPE, all of them are shown. Depending on the state of a connection, the edge is drawn using a solid line (for functioning connections), dotted line (for broken connections), or dashed line (for connections yet to be established).

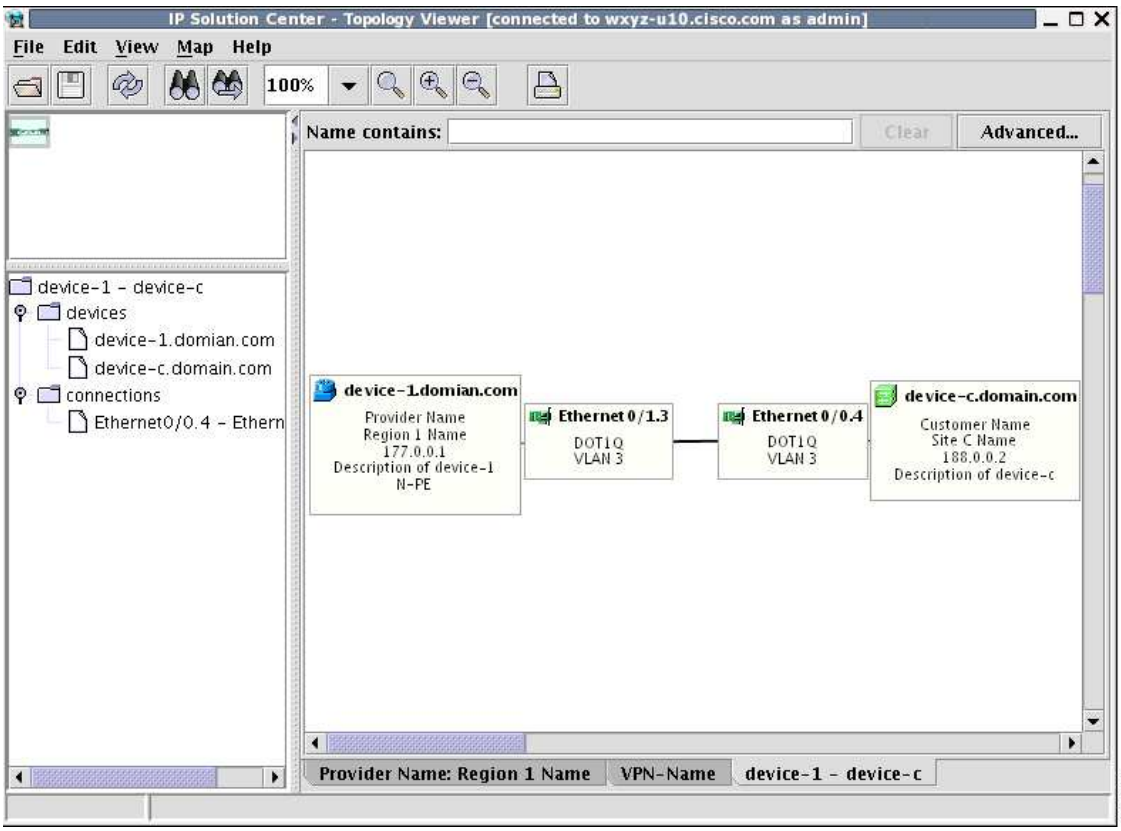
Depending on the connection type, the connection is drawn as described in Table 12-2 and Table 12-3. Each connection is annotated with the PE Interface Name (IP address), VLAN ID number, CPE Interface Name (IP address).

In the Overview area, a direct connection is drawn between a CPE and a PE, even if a number of devices are forming such a connection.

For more about viewing device properties, see Viewing Device and Link Properties, page 12-17.

To view the details of a connection, right-click on it and select the **Expand** option from a pop-up menu. The expanded view, displayed in a new tab, shows all devices and interfaces making a given PE to CPE connection, as shown in Figure 12-11.

Figure 12-11 Detailed Connection View



Physical View

A physical view shows all named physical circuits defined for PEs in a given region. Each named physical circuit is represented as a sequence of connections leading from a PE through its interfaces to interfaces of CLEs or CPEs. All physical links between PEs of a given region and their CLEs or CPEs are shown. Since physical links are assumed to be in a perfect operational order, edges are always drawn with solid lines.

To activate the physical view, follow these steps:

Step 1 In the menu bar, choose **File > Open**.

or

click the **Open** button in the tool bar.

The Folder View window, as shown in Figure 12-5, appears.

Step 2 Choose the desired VPN's folder and double-click on the desired folder.

Any logical and physical views associated with that VPN are displayed.

Step 3 To open the physical view for the selected VPN, do one of the following:

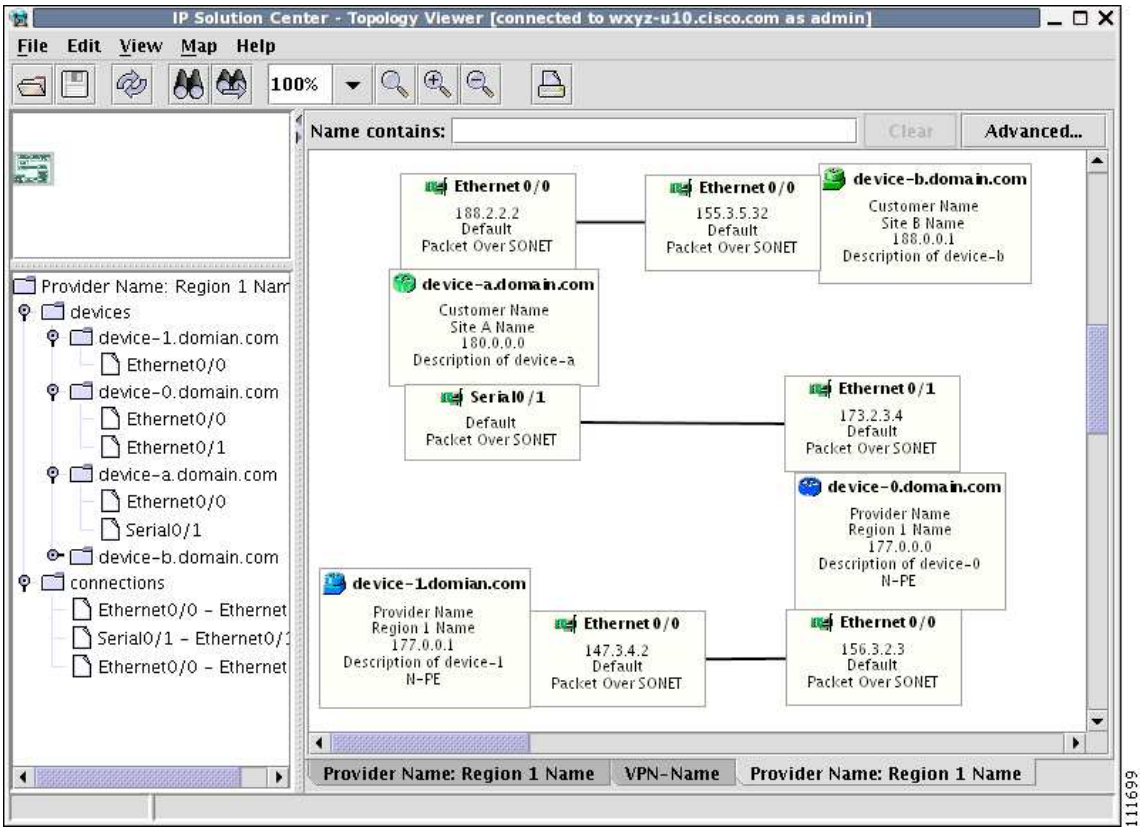
Single-click the **Physical View** icon and click **Open**

or

Double-click the **Physical View** icon.

This creates a physical view for the chosen VPN, as shown in Figure 12-12.

Figure 12-12 Physical View



In this view, each device is connected with a thin line to the interfaces it owns. Interfaces are connected to other interfaces with thick lines. If there is more than one connection between two interfaces, they are spaced to show all of them.

The tree shows devices and connections. Each device can be a folder, holding all interfaces connected to it.

Viewing Device and Link Properties

In the logical view, you can view the properties of both devices and links. In the physical view, only properties of physical devices are accessible.

Thus, device properties can be viewed in both the logical and physical views.

Device Properties

To view the properties of a device, right-click the device. The Device Properties menu appears.

The following properties are available:

Logical Device...—View the logical properties of the device.

Physical Device...—View the physical properties of the device.

Interfaces...—View interface properties of the device.

Service Requests...—View service request properties associated with the device.

Logical Device

When right-clicking a device and selecting **Logical Device...**, the logical device properties window appears.

The logical properties window displays the following information:

Device Name—Name of the device.

Provider Name—Name of the provider whom the device is serving.

Region Name—Name of the provider region.

Loopback Address—IP address of the loopback address.

Role Type—Role assigned to the device.

Physical Device

When right-clicking a device and selecting **Physical Device...**, the physical device properties window appears.

The physical properties window displays the following information:

Name—Name of the device.

Description—User-defined description of the device.

Collection Zone—Collection zone for device data.

IP Address—IP address of the interface used in the topology.

User ID—User ID for the interface.

Enable User—Password for the interface.

Device Access Protocol—Protocol used to communicate with the device.

Config Upload/Download—Upload/download method for the configuration file.

SNMP Version—Simple Network Management Protocol (SNMP) version on the device.

Community String RO—**public** or **private**

Community String RW—**public** or **private**

SNMP Security Level—Simple Network Management Protocol (SNMP) security level.

Authentication User Name—User name for performing authentication on the device.

Authentication Algorithm—Algorithm used to perform authentication.

Encryption Algorithm—Encryption algorithm used for secure communication.

Terminal Server—Name of the terminal server.

Terminal Server Port—Port number used by the terminal server.

Platform—Hardware platform.

Software—IOS version or other management software on the device.

Image Name—Boot image for device initialization.

Serial Number—Serial number of the device.

Interfaces

When right-clicking a device and selecting **Interfaces...**, the interface properties window appears.

The interface properties window displays the following information:

Name—Name of the device.

IP Address—IP address of the device.

IP Address Type—STATIC or DYNAMIC.

Encapsulation—Encapsulation used on the interface traffic.

Description—Description assigned to the interface, if any.

Select (link)—If a connection is attached to the interface, a drop-down list at the bottom of the window allows you to choose between the interfaces available on the device.

Service Requests

When right-clicking a device and selecting **Service Requests...**, the service request (SR) properties window appears.

The service request properties window displays the following information:

Job ID—SR identifier.

Type—Protocol type used in the SR.

State—SR state.

Operation Type—Encapsulation used on the interface traffic.

Creator—Description assigned to the interface, if any.

Creation Time—Date and time when the SR was created.

Customer Name—Name of customer associated with the SR.

Last Modified—Date and time when the SR was last modified.

Description—User-defined description of the SR.

Select (SR)—If more than one SR is associated with the interface, the drop-down list at the bottom of the window allows you to choose between these SRs.

Link Properties

To view the properties of a given link, right-click the link. The Link Properties menu appears.

The following options are available:

Expand—View link details, including devices local to the link not shown in the general topology.

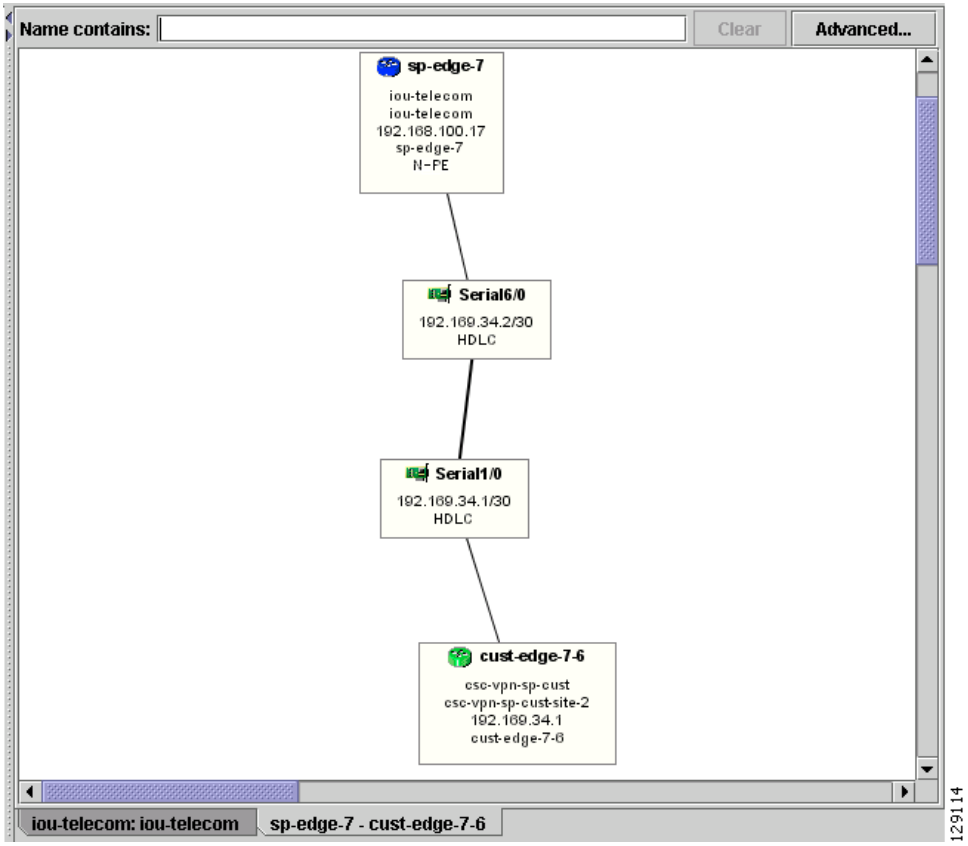
Service Request...—View service request properties associated with the link.

MPLS VPN—View the MPLS VPN properties of the link. Other link protocol properties than MPLS VPN are currently not available.

Expand

When right-clicking a link and selecting **Expand...**, the Topology Display will display any devices and connections local to that link. An Expand Link window similar to the one in Figure 12-13 will appear.

Figure 12-13 Expand Link Window



Properties information for devices and links can only be obtained in the master view as described earlier in this section.

Service Request

When right-clicking a link and selecting **Service Requests...**, the service request (SR) properties window appears.

The service request properties window displays the following information:

Job ID—SR identifier.

Type—Protocol type used in the SR.

State—SR state.

Operation Type—Encapsulation used on the interface traffic.

Creator—Description assigned to the interface, if any.

Creation Time—Date and time when the SR was created.

Customer Name—Name of customer associated with the SR.

Last Modified—Date and time when the SR was last modified.

Description—User-defined description of the SR.

Select (SR)—If more than one SR is associated with the interface, the drop-down list at the bottom of the window allows you to choose between these SRs.

MPLS VPN

When right-clicking a link that is configured for MPLS VPN and selecting **MPLS VPN...**, the MPLS VPN properties window appears.

The service request properties window displays the following information:

Status—Status of the MPLS VPN link.

Status Message—Displays any error or warning messages.

Operation Type—MPLS operation type.

Policy Type—The policy type applied to the link.

Data MTD Threshold—Memory Technology Driver (MTD) data threshold.

Default MTD Address—Default MTD IP address.

Data MTD Subnet—Data MTD subnet.

Data MTD Size—Data MTD size.

SOO Enabled—Site of Origin Enabled - **Yes** or **No**.

Manual Config—**Yes** or **No**.

Filtering and Searching

On large graphs, the amount of detail can be overwhelming. In such cases, filtering might help eliminate unnecessary details, while searching can lead to a prompt location of a device you want to examine further.

Both advanced filtering and searching use the same window to enter conditions on nodes to be either filtered or located. The filtering area also allows you to quickly filter viewed objects by name.

Filtering

The topology view can be filtered in two ways, simple and advanced.

Simple Filtering

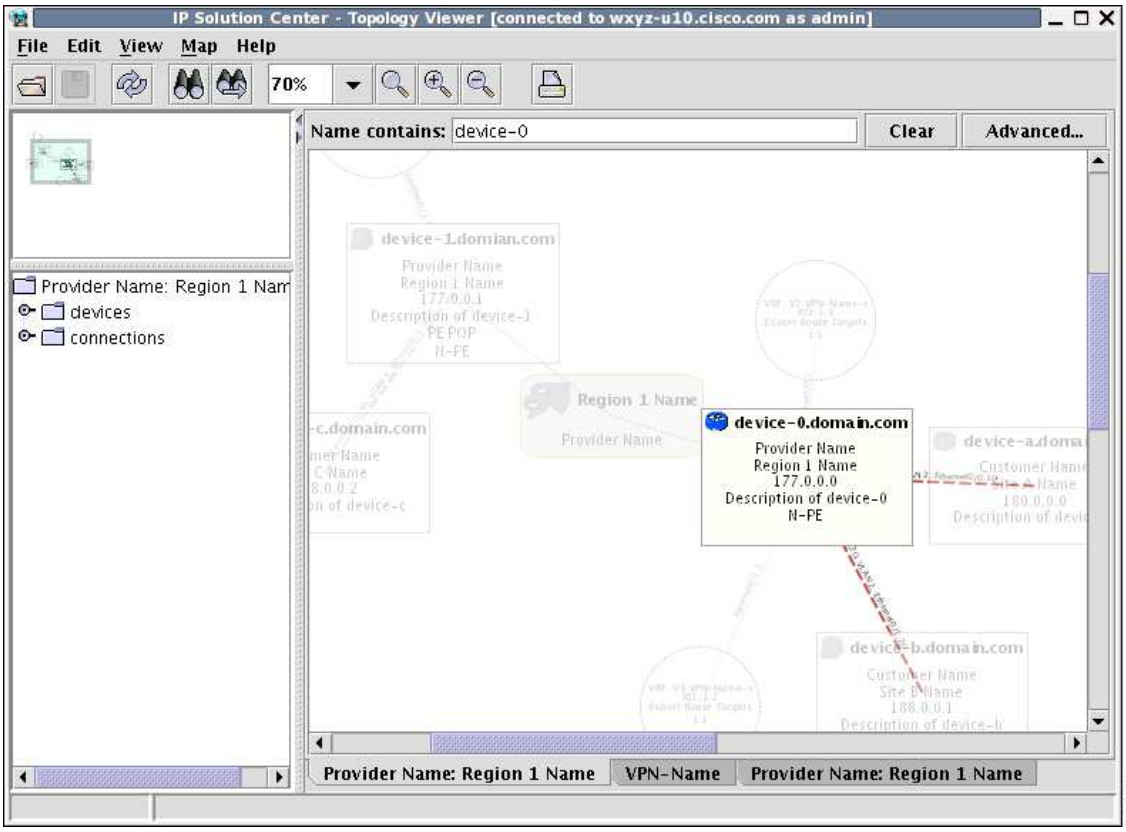
To perform simple filtering of the view, follow these steps:

Step 1 Enter a string in area (4) of the main window, as shown in Figure 12-2.

Step 2 Press **Enter** to dim all objects whose name does not contain the specified string.

For example, to locate nodes that contain string **router** in their name you would enter **router** in area (4) and click **Enter**. All objects whose name does not contain the entered string are dimmed, as shown in Figure 12-14.

Figure 12-14 Physical View with Dimmed Nodes



Note Regular expressions are supported but only in the advanced window (click **Advanced...** button). For example, by entering ^foo.*a, you only request nodes that have names starting with "foo" followed by arbitrary characters and containing the letter 'a' somewhere in the name. The regular expressions must follow the rules defined for Java regular expressions.

Advanced Filtering

To perform advanced filtering, follow these steps:

Step 1 Open the advanced filtering window by clicking the **Advanced...** button.

The Advanced Filter window appears.

Step 2 Make the desired filtering elections.

The window allows you to enter one or more conditions on filtered nodes. The first drop-down list allows you to specify the attribute by which the filtering is performed. The second allows you to decide how the matching between the value of the attribute and text entered in the third column is performed.

The following matching modes are supported from the drop-down list:

- **contains**—The attribute value is fetched from the device and it is selected if it contains the string given by you. The string can be located at the start, end, or middle of the attribute for the match to succeed. For example, if the pattern is **cle** the following values match it in the **contains** mode: **clean**, **nucleus**, **circle**.
- **starts with**—The value of the attribute must start with the string given by you. For example, if the pattern is **foot**, **footwork** matches, but **afoot** does not.
- **ends with**—This is the reverse of the **starts with** case, when a given attribute matches only if the specified pattern is at the end of the attribute value. In this mode, for example, the pattern **foot** matches **afoot** but not **footwork**.
- **doesn't contain**—In this mode, only those strings that do not contain the given pattern match. The results are opposite to that of the **contains** mode. For example, if you specify **cle** in this mode, **clean**, **nucleus**, and **circle** are rejected, but **foot** is deemed to match, because it does not contain **cle**.
- **matches**—This is the most generic mode, in which you can specify a full or partial expression that defines which nodes you are interested in.

By clicking one of the two radio buttons, **Match any conditions** or **Match all conditions**, you can request that any or all of the conditions are matched. In the first case, you can look for devices where, for example, the name contains **cisco** and the management IP address ends with **204**. When all conditions must be met, it is possible to look for devices that, for example, have a given name and platform.

Click **More** or **Fewer** to add more rows of conditions or remove existing rows of conditions.

By default, all matches are performed without regard for upper or lower case. However, in some cases it is beneficial to have a more exact matching that takes the case into account. To do so, check the **Match case** check box.

Step 3 Click **OK** to start the filtering process. Click **Cancel** to hide the window without any changes to the state of the filters.

The **Clear** button allows you to clear all conditions. Clicking **Clear** followed by **OK** effectively removes all filtering, restoring all nodes to their default brightness level. If filtering is active, the same can be achieved by clicking **Clear** in area (4) of the main window, as shown in Figure 12-2.

Searching

Searching can be conducted by using the menus or the tool bar. To perform a search, follow these steps:

Step 1 Select **Find** in the **Edit** menu

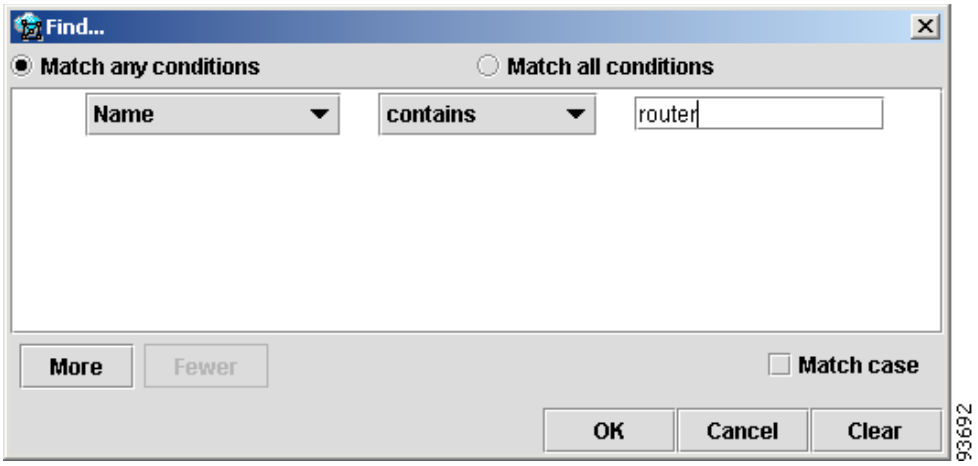
or

Click the **Find** icon in the main toolbar.

Both approaches bring up the same window, as shown in Figure 12-15.

Again, you can enter one or more conditions to locate the node.

Figure 12-15 Find Window



Step 2 Make the desired filtering selections.

Match modes, case check box, and the radio button are used as described under Advanced Filtering, page 12-21.

Step 3 Click **OK** to start searching for the first node that matches the given criteria.

If found, the node is highlighted and the view is shifted to make it appear in the currently viewed area of the main window.

Step 4 After the first search, press **F3** or click the **Find Again** button to repeat the search

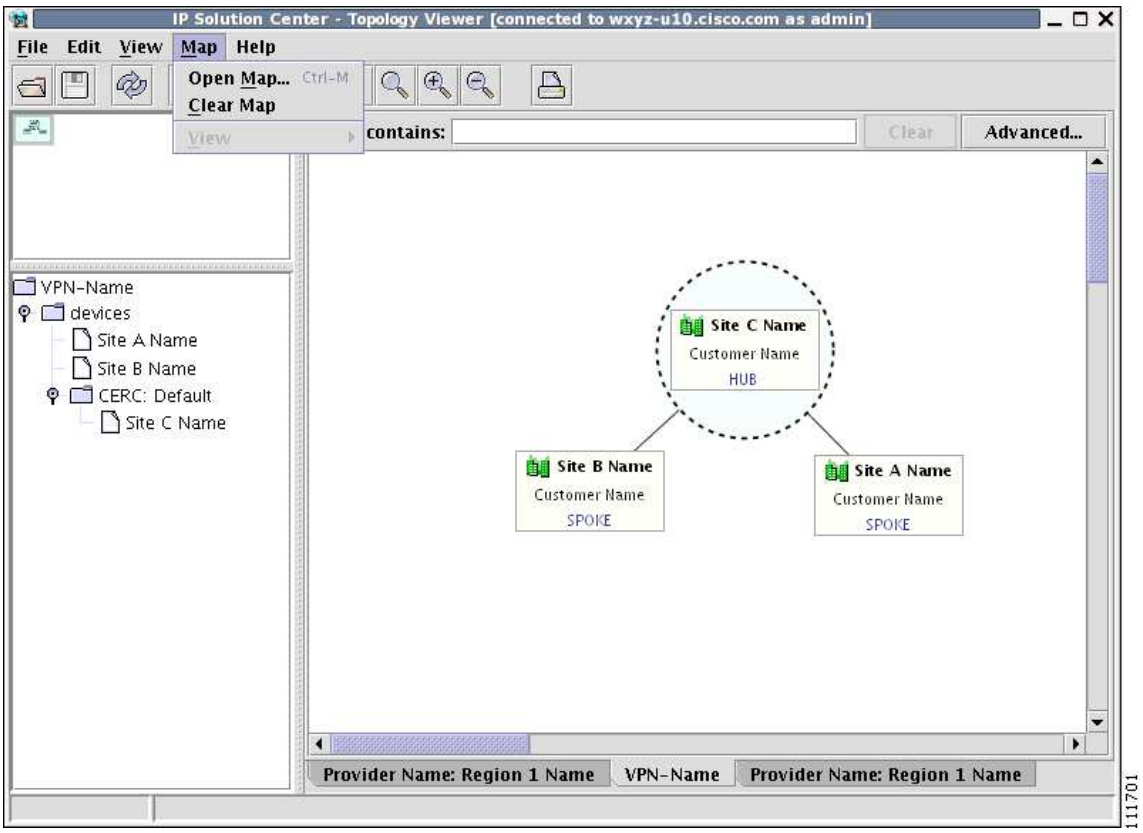
If more than one node matches the condition the **Find Again** function highlights each one of them. If no nodes match the entered criteria, the **Object Not Found** window appears.

Using Maps

You can associate a map with each view. Currently, the topology viewer only supports maps in the Environmental Systems Research Institute, Inc. (ESRI) shape format. The following sections describe how to load maps and selectively view map layers and data associated with each map.

The map features are accessed from the **Map** menu shown in Figure 12-16.

Figure 12-16 The Map Menu



The **Map** menu contains the following menu items:

- **Open Map**—Loads a map into the application
- **Clear Map**—Clears the active map from the current view
- **View**—Allows you to select which layers in the map should be displayed (for example, country, state, city).

Loading a Map

You might want to set a background map showing the physical locations of the displayed devices. To load a map, follow these steps:

Step 1 In the menu bar, select **Map > Open Map...**.

or

Press **Ctrl-M**

Step 2 Make your selections in the Load Map window.

The right-hand side of the window contains a small control panel, which allows you to select the projection in which a map is shown. A map projection is a projection that maps a sphere onto a plane. Typical projections are Mercator, Lambert, and Stereographic.

For more information on projections, consult the Map Projections section of Eric Weisstein's World of Mathematics at:

http://mathworld.wolfram.com/topics/MapProjections.html

For each projection, you can also select the region of the map to be shown. In most cases, the predefined values should be sufficient.

If desired, make changes to the settings in the **Longitude Range** and **Latitude Range** fields.

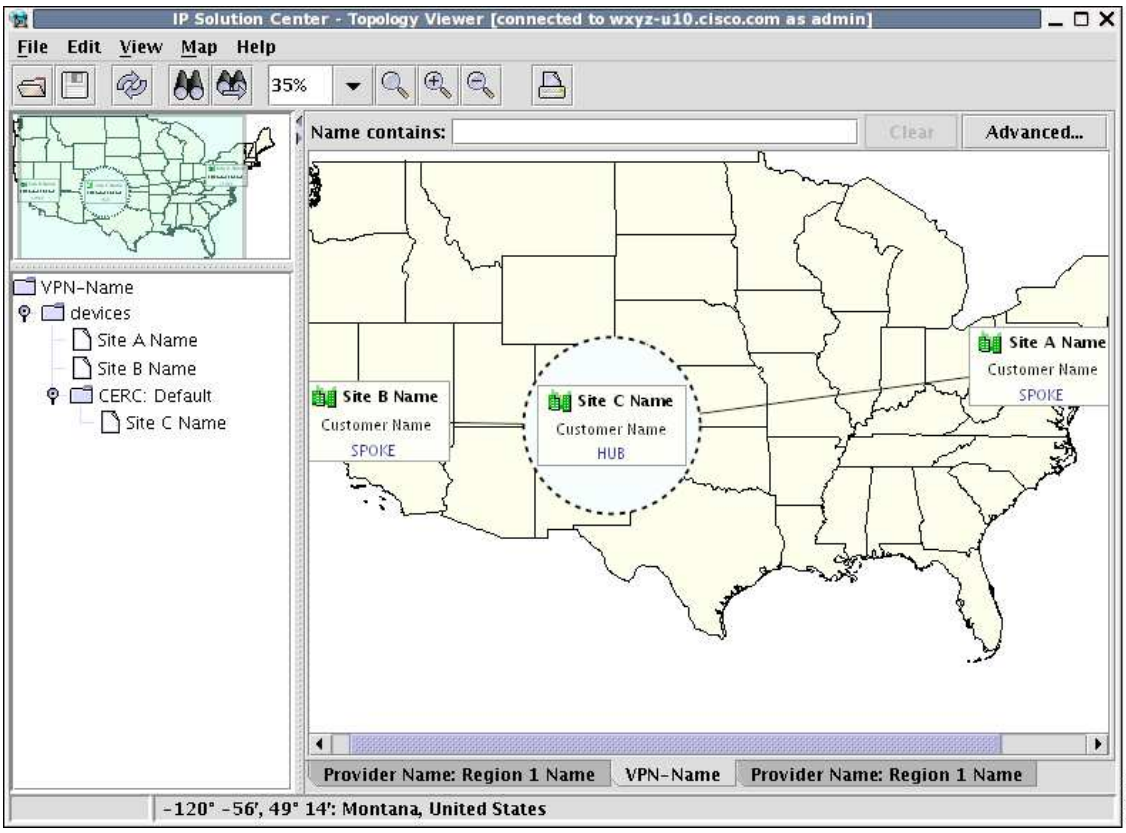
Step 3 Select a map file and click **Open** to load the map.

Selecting the map file and clicking the **Open** button starts loading it. Maps can consist of several components and thus a progress window is shown informing you which part of the map file is loaded.

Layers

Each map can contain several layers. For example most country maps have country, region, and city layers, as shown in Figure 12-17.

Figure 12-17 Map Layers



After a map is loaded, the **View** submenu of the **Map** menu is automatically populated for you. A name of each available layer is shown together with the check box indicating visibility of the layer. If a given map shows too many details, you can turn off some or all layers by unchecking the corresponding check box(es). The same submenu can be used to restore visibility of layers.

If an incorrect map is loaded or the performance of the topology tool is unsatisfactory with the map loaded, you can clear the map entirely. To do this, select **Clear Map** from the **Map** menu. Maps are automatically cleared if another map is loaded.

Consequently if you want just to load another map, there is no need to clear the existing map. The act of loading a new map does this.

Map Data

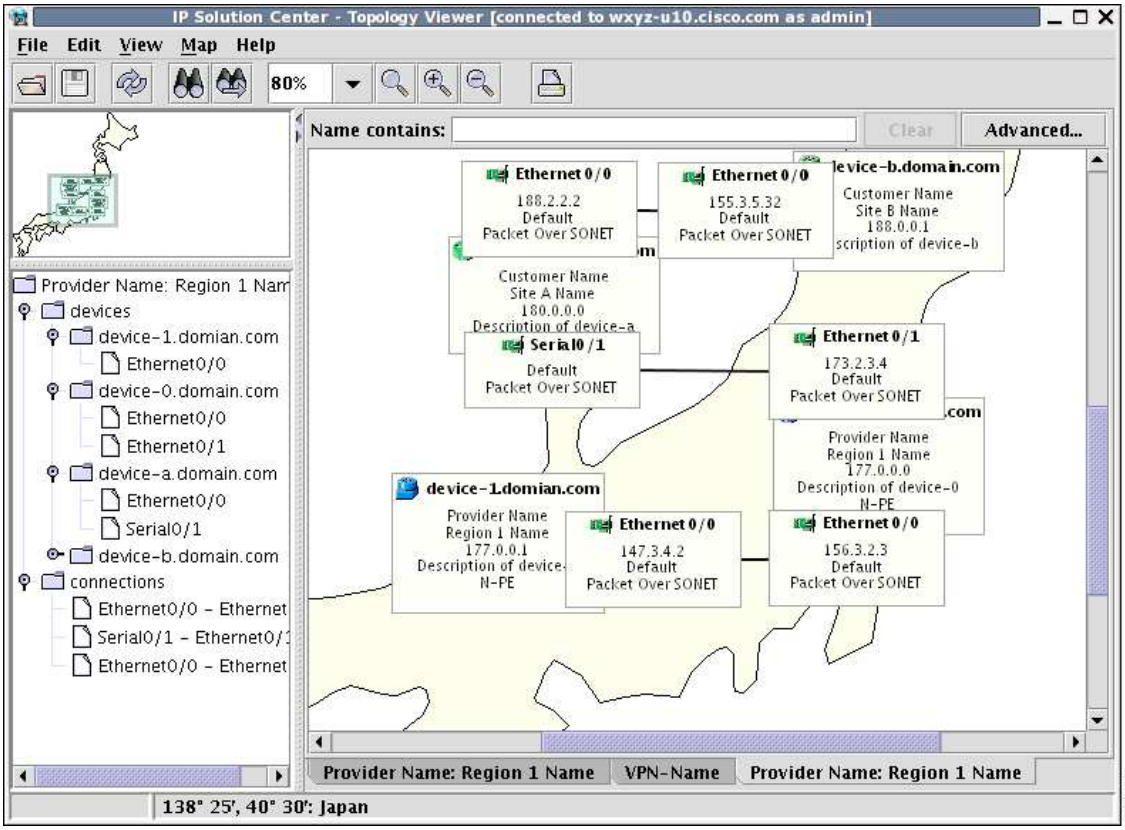
If map data files are successfully loaded with the map, the right field of the Status bar shows the longitude and latitude location of the cursor on the map. If map objects, such as cities, lakes, and so on, have data associated with them, their names are displayed after the longitude and latitude coordinates.

Node Locations

After a map is successfully loaded, the view area is adjusted to fully accommodate it, as shown in Figure 12-18. If nodes shown on the window had longitude and latitude information associated with them, they are moved to locations on the map corresponding to their geographical location. If not, their positions remain unchanged.

However, you can manually move them to the desired location and save the positions for future reference. The next time the image of a given network is loaded, node positions are restored and the map file is loaded.

Figure 12-18 Physical View with a Map of Japan



Adding New Maps

You might want to add your own maps to the selection of maps available to the topology application. This is done by saving maps in the root directory. To make this example more accessible, assume that you want to add a map of Toowong, a suburb of Brisbane, the capital of Queensland. The first step to do so is to obtain maps from a map vendor. All maps must be in the ESRI shape file format (as explained at the web site: http://www.esri.com). In addition, a data file might accompany each shape file. Data files contain information about objects whose shapes are contained within the shape file. Let us assume that the vendor provided four files:

- toowong_city.shp
- toowong_city.dbf
- toowong_street.shp
- toowong_street.dbf

Then assume you want to create a map file that informs the topology application about layers of the map. In this case, you have two layers: a city and a street layer. The map file, say, Toowong.map, would thus have the following contents:

```
toowong_city
toowong_street
```

It lists all layers that create a map of Toowong. The order is important, as the first file forms the background layer, with other layers placed on top of the preceding layers.

Having obtained shape and data files and having written the map file, decide on its location. As mentioned, Toowong is a suburb of Brisbane, located in Queensland, Australia. All map files must be located in or under the **$PRIMEF_HOME/resources/webserver/tomcat/webapps/ipsc-maps/data** directory. Since by default this directory contains a directory called **Oceania** intended for all maps from that region, simply create a path **Australia/Queensland/Brisbane** under the directory **Oceania**. Next, place all five files in this location. After this is done, the map is automatically accessible to the topology viewer.

CHAPTER 13

Using Inventory Manager

This chapter explains how Inventory Manager provides a method of managing mass changes to inventory and service model data in the Cisco Prime Fulfillment provisioning process. In this process, Inventory Manager enables an operator to import network-specific data into the Prime Fulfillment Repository (Repository) in bulk mode. Prime Fulfillment now supports the import of inventory from Prime Network. The inventory that can be imported are device credentials, software version, and SNMP details. All other physical and logical inventory is retrieved from the device using collect configuration. It contains the following sections:

Inventory - Device Console

Inventory - Device Console is the starting point for many operations. Inventory Manager performs three primary functions:

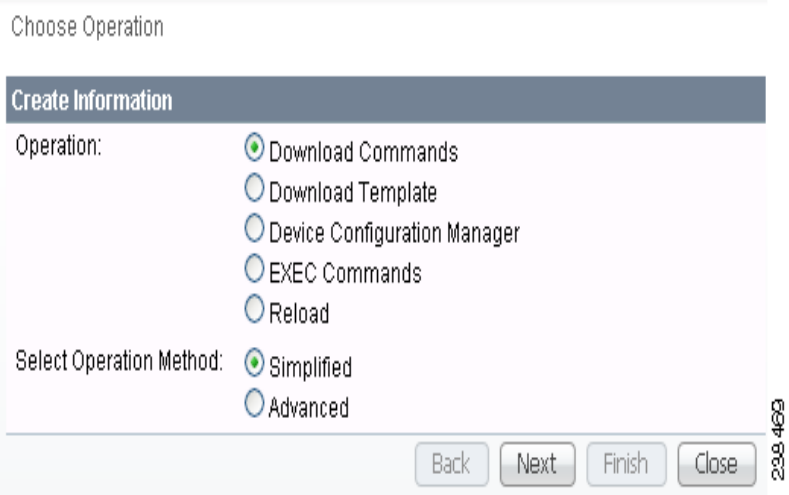
- Imports devices from configuration files and configures CPEs and PEs by associating devices with a Customer or Provider.
- Edits devices, CPEs or PEs stored in the Prime Fulfillment repository.
- Assigns a device to a provider or customer.

To navigate through **Device Console**, follow these steps:

Step 1 Choose **Inventory> Device Tools > Device Console** and you receive a window as shown in the example in Figure 13-1.

Note The radio button last selected will be the one shown in Figure 13-1.

Figure 13-1 Device Console window



Step 2 To select one of the operations, click the radio button for one of the following selections and then click **Next**.

Note All operations apply only to Live mode, *not* ECHO mode.

- Download Commands, page 13-2—Download operation commands and configlets. The **Select Operation Method** selections of **Simplified** and **Advanced (via wizard)** are only available for **Download Commands** and are explained in that section.
- Download Template, page 13-3—Downloads template configlets to the specified devices.
- Device Configuration Manager, page 13-6—Displays different versions of configuration files created on a repository per timestamp and writes to running-configuration or start-up configuration.
- EXEC Commands, page 13-8—Allows you to send to target devices any Cisco IOS commands that can be executed in enable mode.
- Reload, page 13-10—Remotely reloads devices.

Download Commands

To download commands, follow these steps:

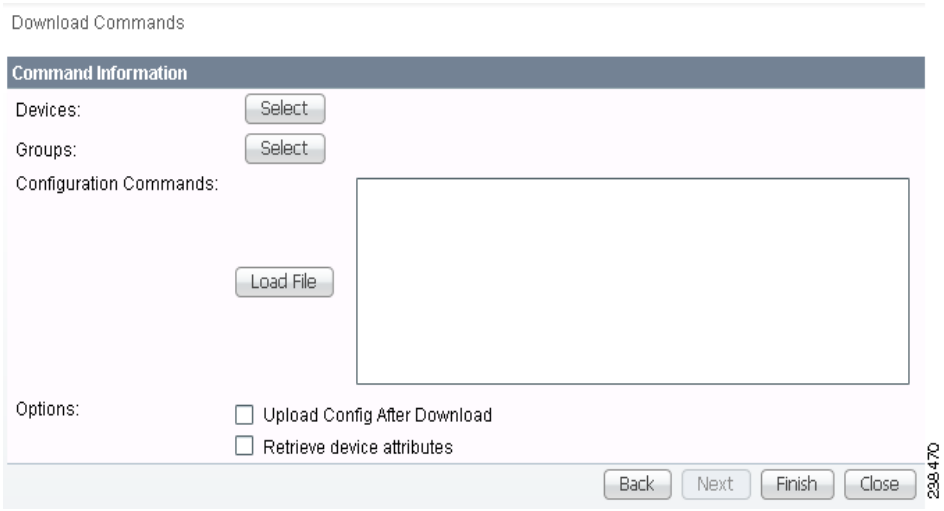
Step 1 Choose **Inventory> Device Tools > Device Console > Download Commands.**

Step 2 The **Select Operation Method** default is **Simplified**, which indicates that in a single window you have the options for selecting the Devices, Device Groups, and Operation Commands. You do not need to multi-click. In a single window you can submit the required parameters to complete the task. **Advanced (via wizard)** indicates you must go to multiple windows to achieve the task. In this method, you select Device, click **Next,** select Device Groups, click **Next**, select Operation Command, and then the summary.

Step 3 Click **Next**.

A window as shown in Figure 13-2.

Figure 13-2 Device Console—Download Commands: Select Devices



Step 4 In the **Devices** row, click **Select/Deselect**. In the new window, check the check box for each device you want. Uncheck a check box if you do not want this device. Then click **Select**. Figure 13-2 then reappears with the selected devices in the **Devices** row.

Step 5 In the **Groups** row, click **Select/Deselect**. In the next window, check the check box for each group you want. Uncheck a check box if you do not want this group. Then click **Select**. The selected groups appear in the **Groups** row.

Step 6 In the **Operation Commands** field, enter the commands you want to download or click **Load File** to select a set of commands to place in the **Operation Commands** field.

If you leave the **Upload Config After Download** check box unchecked, you do *not* upload the configuration file after the download.

If you leave the **Retrieve device attributes** check box unchecked, you do not retrieve any device attributes. If you check the **Retrieve device attributes** check box, after the template is downloaded, SNMP is used to retrieve interface information and issue additional **show** commands, such as **show version**.

Step 7 Click **OK** to submit the download and you receive a window with the **Device Console Operation Result** and in the bottom left corner a **Status**. You can click **Download** or **Done**.

Step 8 When you click **Download**, you return to Step 6 to download additional commands on the selected devices.

Step 9 When you click **Done**, you return to Figure 13-1.

Download Template

Note Multiple datafiles belonging to different templates cannot be downloaded through the device console.

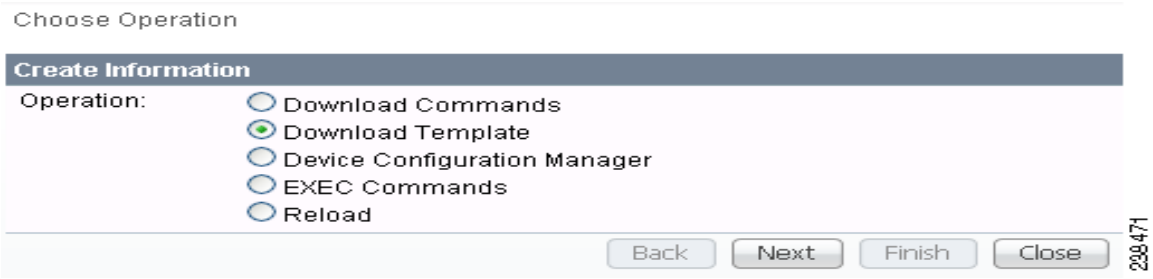
To download a template, follow these steps:

Step 1 Choose **Inventory> Device Tools > Device Console .**

Step 2 Select **Download Template** and click **Next**.

A window as shown in Figure 13-3.

Figure 13-3 Device Console—Download Template: Select Devices



Step 3 Continue with Step 4 if you want to add devices; proceed to Step 9 to delete devices; or click **Next** to proceed to Step 11 for **3. Select Device Groups**.

Step 4 Click **Add**, as shown in Figure 13-3, to **2. Select Devices**.

Step 5 From the resulting Device Selection window, check the check box(es) for each device you want to select. Then click **Select**.

Step 6 You return to Figure 13-3 with the added devices.

Step 7 For each device, you can click the added **Clear** button to clear the **Upload to Customer/Site** column to reflect **none selected**, or you can click the added **Select** button and a new window allows you to **Create Customer**, **Create Site**, **Select**, or **Cancel**. When you click **Select** in this new window, you return to Figure 13-3 with the added customer or site.

Step 8 You can repeat Step 4 to Step 7 to **add** more devices, you can delete devices, as explained in Step 9, or you can proceed by going to Step 10.

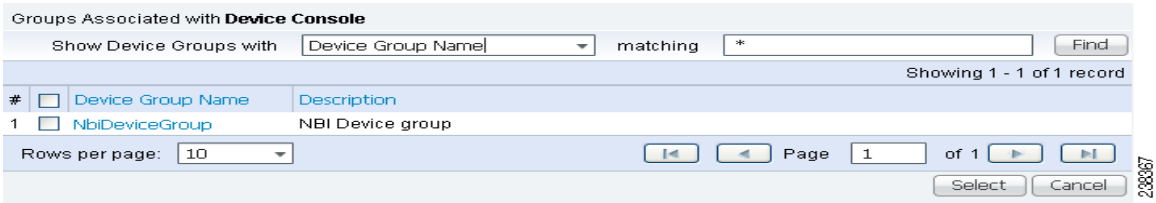
Step 9 To delete devices, check the check box(es) for the devices you want to delete and then click **Delete**. Select carefully, because there is no chance to confirm this deletion.

Step 10 When you have all the devices you want, click **Next**. You proceed to **3. Select Device Groups**, starting in Step 11.

Step 11 Continue with Step 12 if you want to add device groups; proceed to Step 14 to delete device groups; or click **Next** to proceed to Step 16 for **4. Enter Download Commands**.

Step 12 Click **Add**, as shown in Figure 13-4, to **3. Select Device Groups**. Adding Device Groups is optional.

Figure 13-4 Device Group Selection



Step 13 From the resulting window, check the check box(es) for each device group you want to select. Then click **Select**.

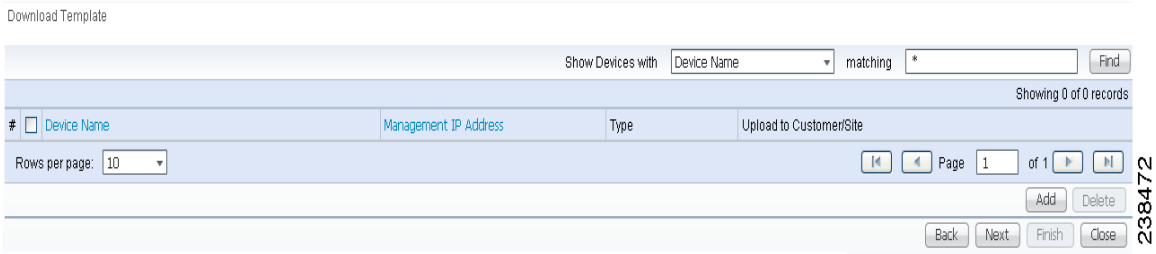
You return to Figure 13-4 with the added device groups. You can repeat Step 12 to Step 13 to **add** more device groups, you can delete device groups, as explained in Step 14, or you can proceed by going to Step 15.

Step 14 To delete device groups, check the check box(es) for the devices you want to delete and then click **Delete**. Select carefully, because there is no chance to confirm this deletion.

Step 15 When you have all the device groups you want, click **Next**. You proceed to **4. Select Download Template**, starting in Step 16.

Step 16 For **4. Select Download Template**, the resulting window is shown in Figure 13-5.

Figure 13-5 Select Download Template



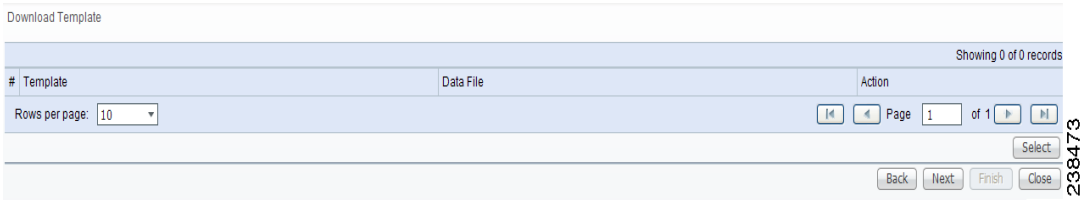
Step 17 In Figure 13-5, you can click the **Select** button.

A window as shown in Figure 13-6.

Step 18 Click **Add** to add templates or **Remove** to remove templates. When you have the templates you want, click **OK**.

When you click **Add** you get a Template Datafile Chooser window with the template choices in the tree. Click **+** to open the folders and subfolders in the tree, until you get the property you want to choose. Click on that property and it is added to your list. Repeat this until all the templates you want are in your list. In each added property, you can click **View** and you receive the configlet for that data file. To return, click **OK**. In Figure 13-6, check the check box(es) for the template(s) you want. In each template row, click the **Action** drop-down list and choose **APPEND** or **PREPEND** to add information after or before, respectively; check or uncheck the **Active** check box; and then click **OK**.

Figure 13-6 Add/Remove Templates

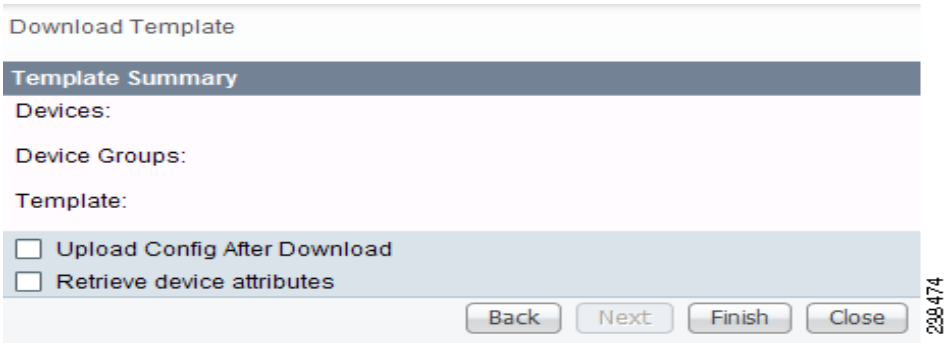


Step 19 You return to Figure 13-5 with the updated information.

Step 20 Click **Next** and you proceed to **5. Download Template Summary**, as explained in Step 21.

Step 21 For **5. Download Commands Summary**, a window as shown in Figure 13-7.

Figure 13-7 Download Template Summary



Step 22 In Figure 13-7, if you leave the **Upload Config After Download** check box unchecked, you do *not* upload the configuration file after the download. If you check the **Upload Config After Download** check box, you upload the new configuration file after you download the templates in . If you leave the **Retrieve device attributes** check box unchecked, you do not retrieve any device attributes. If you check the **Retrieve device attributes** check box, after the template is downloaded, SNMP is used to retrieve interface information and issue additional **show** commands, such as **show version**.

Step 23 Click **Back** until you correct any information you want to change or click **Finish** to submit the download and you receive a window with the **Download Template Results** and in the bottom left corner a **Status** with a green check mark for **Succeeded**.

Step 24 Click **Done and** you return to Figure 13-1.

Device Configuration Manager

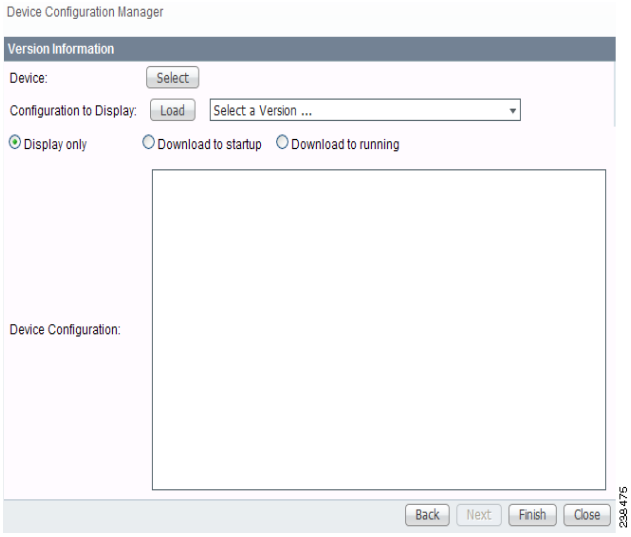
To display the configuration, download the configuration to the startup configuration on the device, or download the configuration to the running configuration on the device, follow these steps:

Step 1 Choose **Inventory> Device Tools > Device Console.**

Step 2 Select **Device Configuration Manager** and click **Next.**

A window as shown in Figure 13-8.

Figure 13-8 Device Configuration Manager



Step 3 In the **Device** row, click **Select**.

Step 4 From the devices listed, click the radio button for the device you want to select. Then click **Select**.

Step 5 You return to Figure 13-8 with the added device. You can repeat Step 3 to Step 4 to change the device.

Step 6 When you have selected the device you want, go to the **Configuration to Display** row and click the **Select a Version...** drop-down list. Click the version you want and then click **Load** to load that configuration file.

Step 7 Click one of the following radio buttons or keep the default:

- **Display only**—The configuration file can only be viewed.
- **Download to startup**—The configuration file is downloaded to the start up configuration of the selected router.

Note For **Download to startup**, the Device Access Protocol (defined in device creation) must be either **ftp** or **tftp**. If this is not the case, the Device Configuration Manager Results window appears and indicates that you must set up either **ftp** or **tftp**. Dynamic Component Properties Library (DCPL) properties for DCS for both FTP and TFTP are specified in Appendix B, "Property Settings".

- **Download to running** The configuration file is downloaded to the router's running configuration file.

Note When the DCPL property **copy-running-to-startup** in the **GTL/ios** folder is set to the default of **true**, the router's running configuration file is also copied to the start up configuration.

Step 8 Click **Finish**. If in Step 7 you chose **Display only, you automatically** return to Figure 13-1. If in Step 7 you clicked **Download to startup** or **Download to running**, you get a Device Configuration Manager Results window. In the **Status** box, you get a green check mark for **Succeeded** or a red **Failed** status and you must click **Done** to return to Figure 13-1.

EXEC Commands

EXEC Commands allows you to send to target devices any Cisco IOS commands that can be executed in enable mode. You can only view the router information. You cannot edit or delete the information.

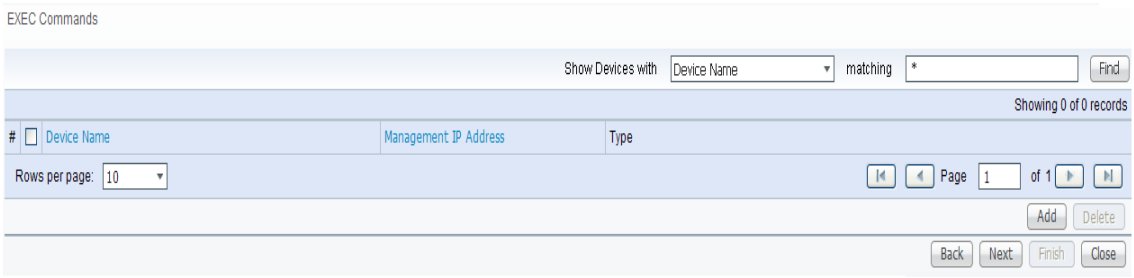
To execute **EXEC Commands**, follow these steps:

Step 1 Choose **Inventory> Device Tools > Device Console.**

Step 2 Select **EXEC Commands** and click **Next**.

A window as shown in Figure 13-9.

Figure 13-9 Device Console—EXEC Commands: Select Devices



Step 3 Continue with Step 4 if you want to add devices; proceed to Step 7 to delete devices; or click **Next** to proceed to Step 9 for **3. Select Device Groups**.

Step 4 Click **Add**, as shown in Figure 13-9, to **2. Select Devices**.

Step 5 From the resulting window, check the check box(es) for each device you want to select. Then click **Select**.

Step 6 You return to Figure 13-9 with the added devices. You can repeat Step 4 to Step 5 to **add** more devices, you can delete devices, as explained in Step 7, or you can proceed by going to Step 8.

Step 7 To delete devices, check the check box(es) for the devices you want to delete and then click **Delete** in Figure 13-9. Select carefully, because there is no chance to confirm this deletion.

Step 8 When you have all the devices you want, click **Next**.

You proceed to **3. Select Device Groups**, starting in Step 9.

Step 9 Continue with Step 10 if you want to add device groups; proceed to Step 13 to delete device groups; or click **Next** to proceed to Step 15 for **4. Enter EXEC Commands**.

Step 10 Click **Add**, as shown in Figure 13-10, to **3. Select Device Groups**.

Figure 13-10 Device Group Selection

Step 11 From the resulting window, check the check box(es) for each device group you want to select. Then click **Select**.

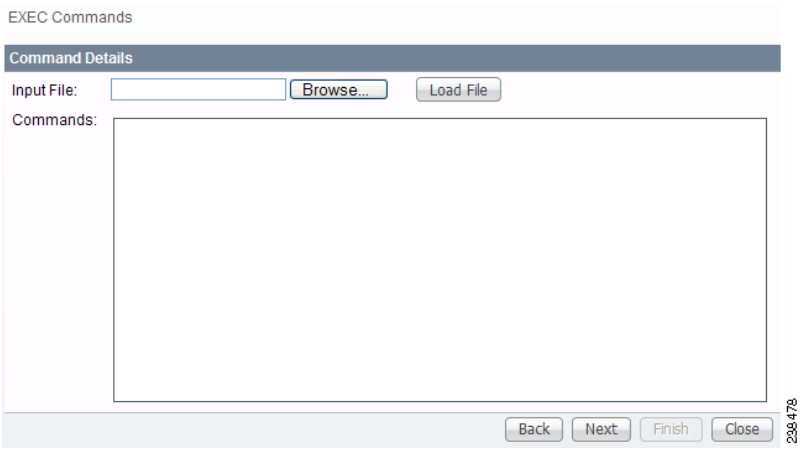
Step 12 You return to Figure 13-10 with the added device groups. You can repeat Step 10 to Step 11 to **add** more device groups, you can delete device groups, as explained in Step 13, or you can proceed by going to Step 14.

Step 13 To delete device groups, check the check box(es) for the devices you want to delete and then click **Delete**. Select carefully, because there is no chance to confirm this deletion.

Step 14 When you have all the device groups you want, click **Next**. You proceed to **4. Enter EXEC Commands**, starting in Step 15.

Step 15 For **4. Enter EXEC Commands**, the resulting window is shown in Figure 13-11.

Figure 13-11 Operation Commands



Step 16 In Figure 13-11, you can click the **Browse** button to input an existing file with Cisco IOS configuration commands. Then click the **Load File** button to put the file's information in the **Commands** field. Otherwise, you can enter the Cisco IOS configuration commands directly in the **Commands** field.

Step 17 Click **Next** and you proceed to **5. EXEC Commands Summary**, as explained in Step 18.

Step 18 For **5. EXEC Commands Summary**, a window as shown in Figure 13-12.

Figure 13-12 EXEC Commands Summary

Step 19 Click **Back** until you correct any information you want to change or click **Finish** to retrieve the information from the router. You then receive a window with the **EXEC Commands Results** and a **Status** with a green check mark for **Succeeded**. You can click **EXEC** or **Done**.

Step 20 When you click **EXEC**, you return to Step 15 to enter additional commands on the selected devices.

Step 21 When you click **Done**, you return to Figure 13-1.

Reload

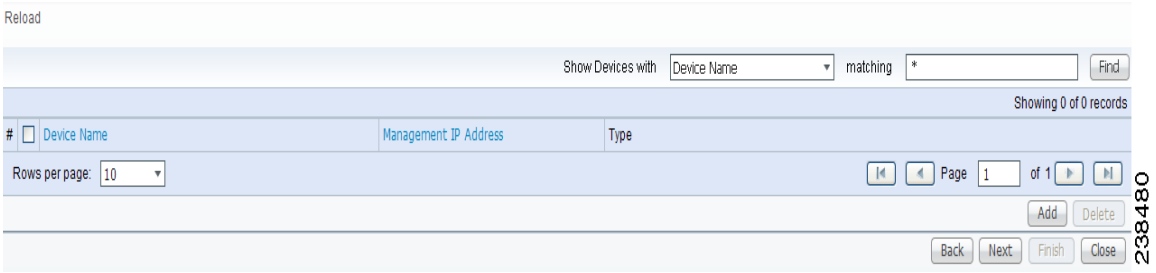
To reload (reboot) the router, follow these steps:

Step 1 Choose **Inventory> Device Tools > Device Console.**

Step 2 Select **Reload** and click **Next**.

A window as shown in Figure 13-13.

Figure 13-13 Device Console—Reload: Select Devices



Step 3 Continue with Step 4 if you want to add devices; proceed to Step 7 to delete devices; or click **Next** to proceed to Step 9 for **3. Select Device Groups**.

Step 4 Click **Add**, as shown in Figure 13-13, to **2. Select Devices**.

Step 5 From the resulting window, check the check box(es) for each device you want to select. Then click **Select**.

Step 6 You return to Figure 13-13 with the added devices. Repeat Step 4 to Step 5 to **add** more devices; delete devices, as explained in Step 7; or proceed by going to Step 8.

Step 7 To delete devices, check the check box(es) for the devices you want to delete and then click **Delete**. Select carefully, because there is no chance to confirm this deletion.

Step 8 When you have all the devices you want, click **Next**. You proceed to **3. Select Device Groups**, starting in Step 9.

Step 9 Continue with Step 10 if you want to add device groups; proceed to Step 13 to delete device groups; or click **Next** to proceed to Step 15 for **4. Reload Devices Summary**.

Step 10 Click **Add**, as shown in Figure 13-14, to **3. Select Device Groups**.

Figure 13-14 Device Group Selection

Step 11 From the resulting window, check the check box(es) for each device group you want to select. Then click **Select**.

Step 12 You return to Figure 13-14 with the added device groups. Repeat Step 10 to Step 11 to **add** more device groups; delete device groups, as explained in Step 13; or proceed by going to Step 15.

Step 13 To delete device groups, check the check box(es) for the devices you want to delete in Figure 13-14 and then click **Delete**. Select carefully, because there is no chance to confirm this deletion.

Step 14 When you have all the device groups you want, click **Next**. You proceed to **4. Reload Devices Summary**, starting in Step 15.

Step 15 For **4. Reload Devices Summary**, a window as shown in Figure 13-15.

Figure 13-15 Reload Summary

Step 16 Click **Back** until you correct any information you want to change or click **Finish** to submit the reload and you receive a window with the **Reload Results** and a **Status** with a green check mark for **Succeeded**.

Step 17 Click **Finish** and you return to Figure 13-1.

Prime Network Device Import

Prime Fulfillment now supports the import of inventory from Prime Network. The inventory that can be imported are device credentials, software version, and SNMP details. All other physical and logical inventory is retrieved from the device using collect configuration. Set the DCPL property from **InventoryImport** before importing Prime Network Device. For more information on setting DCPL properties, see Config, page 14-3 in Chapter 14, “Administration Tasks”.

Note This configuration is required for every new device added to the network.

This feature allows you to perform:

- Device import from Prime Network
- Automated Ring Discovery Process
- Customer Device Insertion via an integrated Single Screen
- Enhanced Inventory Manager for Bulk import from Prime Network

Cisco IOS routers that function as N-PE, U-PE, or PE-AGG are defined as devices from which Prime Fulfillment collects information. Every network element that Prime Fulfillment manages is defined as a device in the system.

The two ways to import devices from Prime Network are:

- Single Device Import during Device Creation, page 13-12
- Bulk Import using Inventory Manager, page 13-13
- Import Prime Network certificate into Prime Fulfillment Trust Store, page 13-14

Single Device Import during Device Creation

To navigate through **Devices** and import a device manually, follow these steps:

Step 1 Choose **Inventory > Physical Inventory > Devices.**

The Device List window appears. Click the **Create** button.

Step 2 Select **Cisco Device** from the drop-down menu.

The **Create Cisco Device** window appears.

See the following sections for descriptions of the fields:

- General Attributes, page 2-7
- Login and Password Attributes, page 2-9
- Device and Configuration Access Information Attributes, page 2-9
- SNMP v1/v2c Attributes, page 2-10

Step 3 Select the device type as Customer Device or Provider Device from the drop-down menu under **Roles** section.

Enter the region name for the Provider that you are creating. To enter the provider region name follow these steps:

a. Click the **Select** button next in Provider Region Name.

A list of provider region names appears.

b. Click the radio button next to provider region name and then **Select**.

Select the device role from the Role Type drop-down menu.

Note The Provider Region Name and PE Role Type options are enabled only if you choose Provider Device as the device type.

Step 4 Check the check box next to **Config Collect** to perform a configuration collection on saving the device.

Configuration Collection is performed at the device creation and device import stages. You can also navigate to **Operate > Task Manager > Task** to create a config task and select the devices created.

Step 5 Check the check box next to **Ring Discovery** to perform ring collection on saving the device. The devices associated with the REP rings are discovered from Active Network Abstration (ANA) and imported into Prime Fulfillment. You can perform ring discovery task from:

- Device Creation window
- Inventory Manager window

Step 6 Check the check box next to MPLS-TP Discovery and MPLS Label Sync to access these details.

Step 7 To access the Additional Properties section of the **Create Cisco Device**, click **Show**.

The Additional Properties window appears.

See the following sections for descriptions of the Additional Properties fields:

- SNMP v3 Attributes, page 2-10
- Terminal Server Options Attributes, page 2-10
- Device Platform Information Attributes, page 2-11

Step 8 Enter any desired Additional Properties information for the Terminal Server device you are creating.

Step 9 Click **Save**.

Step 10 The Devices window reappears with the new imported device listed.

Bulk Import using Inventory Manager

Devices which already exist in ANA can be imported directly into Prime Fulfillment using the option available on the Inventory Manager window.

To perform bulk import of Cisco devices, follow these steps:

Step 1 Choose **Inventory > Physical Inventory > Inventory Manager**.

The **Device List** window appears.

Step 2 Click the **Import Devices** button. Select **ANA**.

Step 3 The **Inventory Import Filter** window appears.

a. You can filter the import of devices from ANA before getting it into Prime Fulfillment.

- The devices available in ANA can be filtered based on Device Host Name, Management IP Address, Element Management Key and Software Version.
- Once filteration is done, a success message displays the number of devices found matching the filter criteria.
- The devices found matching the criteria are displayed on the **Inventory Manager** window. You can perform additional configuration such as role assignment by clicking on **Assign CE/PE** button.
- Select the device and click on **Edit** button to change any of the device parameters before saving the device.
- Click **Save** button to import and save the device into Prime Fulfillment.

b. If you want to import all the devices available in ANA, click **OK** button without providing any filtering criteria on the filter screen.

Step 4 The **Device List** window appears.

Step 5 The Config Collect and Ring Discovery can be scheduled during device import. Click on **Action** button to schedule:

- Config Collect
- Config Collect + Ring Discovery
- Ring Discovery

Step 6 Click **Save**.

The Devices window reappears with the new devices added.

Import Prime Network certificate into Prime Fulfillment Trust Store

To perform import of Prime Network certificate, follow these steps:

Step 1 Add the Prime Network server details in Prime Fulfillment and log into the Prime Network server.

Step 2 Navigate to `<Installation-Path>/ Main/resourcebundle/com/sheer` directory and provide `(ls -alrt)` list command.

Note Make sure the files .keystore & security.properties are available.

Step 3 Export the server certificate from a server keystore (.keystore) using the following command:

```
keytool -export -alias ana -file <certificate-name>.cer -keystore <keystore-name>
<certificate-name> can be - sheer.cer (must end with .cer)
<keystore-name> must be - .keystore
Example: keytool -export -alias ana -file sheer.cer -keystore .keystore
```

Step 4 Transfer (FTP) the certificate `(sheer.cer)`to Prime Fulfillment server installation, etc directory, i.e., `<PRIMEF_INSTALLATTION-DIR>/etc/.`

Step 5 Run the following command from the `<PRIMEF_INSTALLATTION-DIR>` directory to source the environment:

```
./prime.sh shell
```

Step 6 Run the following command from <PRIMEF_INSTALLATTION-DIR>/etc/ directory to import the certificate to Prime Fulfillment keystore:

```
keytool -import -file <certificate-name>.cer -keystore <keystore-name> -alias
<alias-name>
<certificate-name> - must be the name of the Prime  Network certificate.
<keystore-name>  - must be prime.keystore
<alias-name> - unique name to identify the certificate.
Example: keytool -import -file sheer.cer -keystore prime.keystore -alias anacer
```

Step 7 The keytool will prompt for the password. Use the password as **changeit**.

Note In order to confirm the password, check the security. properties file present in the <PRIMEF_INSTALLATTION-DIR>, etc directory.

Step 8 A keytool confirmation to import the certificate while executing the command occurs.

Enter **Yes** to import. The message Certificate was imported successfully appears.

Step 9 To ensure if the certificate is imported, run the following command that lists the trusted certificates added to the keystore:

```
keytool -list -v -keystore prime.keystore
```

Restart the server to reflect the changes.

CHAPTER 14

Administration Tasks

This chapter explains administration tasks to be performed. It contains the following sections:

Manage Active Users and User Account

This section explains how to communicate with active users and manage the user account.

Active Users

Choose **Administration> Active Users > Active Users** and follow these steps:

Step 1 After you choose **Administration> Active Users > Active Users**, a Active Users window appears that shows the currently logged users.

Step 2 If you have the privileges of **SysAdmin** or **UserAdmin**, you can disconnect one or more users. Check the check box next to each user you want to disconnect. Then click the **Disconnect** button at the bottom of the window.

Caution The current login sessions for the disconnected users are terminated and their work is lost.

Step 3 To exit this list of all active users, choose another feature from the main product tabs.

User Account

Choose **Administration> Account > User Account** and follow these steps:

Step 1 After you choose **Administration> Account > User Account**, a User Account window appears that shows the active users.

Step 2 Click **Edit** to change the password, permissions, personal information, and user preferences.

Step 3 Click **Save** to save the changes or click **Cancel**.

Manage Control Center

This section explains how to view and change the properties in the Dynamic Component Properties Library (DCPL); how to view status information about a host, servers, the WatchDog, and logs; how to define collection zones; and how to install license keys.

Choose **Administration > Control Center > Hosts** and you go to the default page of **Hosts**.

The Control Center Hosts window appears.

From **Administration > Control Center > Hosts**, you have the following choices:

- Hosts, page 14-2—**Hosts** allows you to manage the various servers.
- Licensing, page 14-6—**Licensing** is where you install license keys, which is the only way to access services and APIs.

Hosts

Choose **Administration > Control Center > Hosts**.

The Control Center Hosts window appears.

Note Only the **Logs** buttons are enabled by default when there is no host selected. When the host is selected by checking the check box, the Logs buttons is disabled and the other buttons are enabled.

Click any of the buttons and proceed as follows:

- Details, page 14-2—Available only when the host system is chosen.
- Config, page 14-3—Available only when the host system is chosen.
- Servers, page 14-4—Available only when the host system is chosen.
- Watchdog, page 14-5—Available only when the host system is chosen.
- Logs, page 14-5—Available only when no host system selection is made.

Details

For details about a chosen host, follow these steps:

Step 1 Choose a host by checking the check box to the left of the hostname and then click the **Details** button.

You receive the Hosts Details window. This shows the details about the chosen host.

Step 2 Click **OK** and you return to Control Center Hosts window.

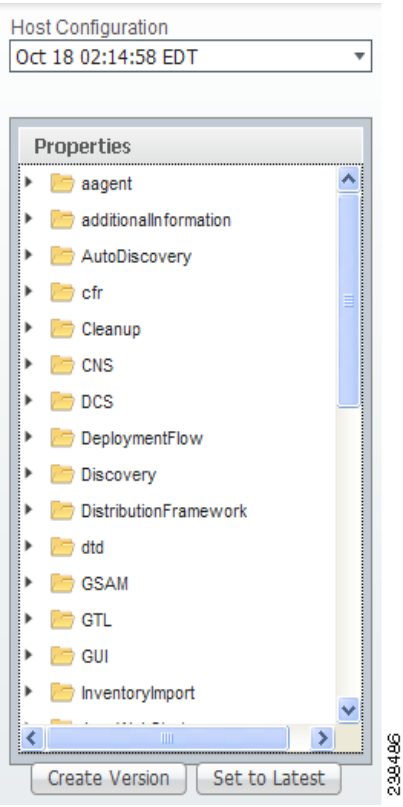
Config

To view or change the Dynamic Component Properties Library (DCPL) properties, follow these steps:

Step 1 From the Control Center Hosts window, check a check box next to a hostname for which you want to know the existing properties and then click the **Config** button.

A window as shown in Figure 14-1, appears. It is a list of all the folders with all the properties. See Appendix B, "Property Settings" for a list of all the properties with explanations, defaults, and ranges/rules. If you do not know the property name, you can use a key word and do a Find on the pdf version of this appendix.

Figure 14-1 Properties



Step 2 Click the **+** sign to expand each folder.

The result could be more subfolders and the final level is the property name.

Step 3 Position the mouse over the folder or property name and you see a description.

Step 4 Click on an entry to get details and instructions on how to change the value, as shown in the example in Figure 14-2.

Figure 14-2 Properties Detail Example

Step 5 For each property that can be modified, you can modify the value and click **Set Property**. If when making your modifications, you want to return to the previous settings, click **Reset Property**.

Step 6 After making all the changes you choose in each of the specific properties, you can click **Create Version** to create a new version of these properties. This feature gives you the option of saving multiple property sets for future use.

Step 7 To view the values of previous versions of property sets, click the drop-down list in **Version** and select any version you choose.

Step 8 When you click **Set to Latest** after selecting a version in Step 7, this version is dated as the most current.

Step 9 To return, click to the navigation path you want to use next.

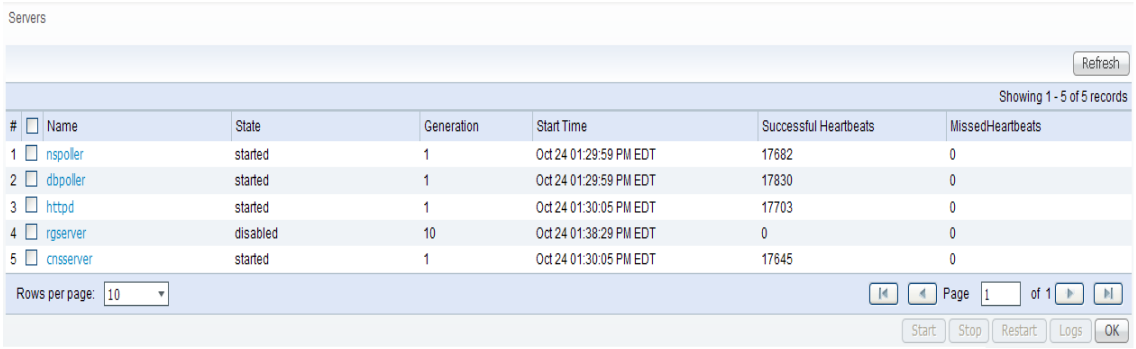
Servers

To view the status information about the servers, follow these steps:

Step 1 From the Control Center Hosts window, check a check box next to a hostname for which you want to know the server statistics and then click the **Servers** button.

A window as shown in Figure 14-3, appears.

Figure 14-3 Servers



Step 2 Check any one check box next to the server you want to address and you have access to **Start**, **Stop**, **Restart**, and **Logs**. When you click on a specific server name or the Logs button, you get a list of server logs. If you then click on the log name for which you want details, the log viewer appears. You can filter this information in the log viewer. After you complete the task of your choice, you return to Figure 14-3.

Step 3 You can click a different server and click the button for the process of your choice. Or you can unclick the server choice and click **OK**.

Step 4 After you click **OK** in Figure 14-3, you return to the Control Center Hosts window.

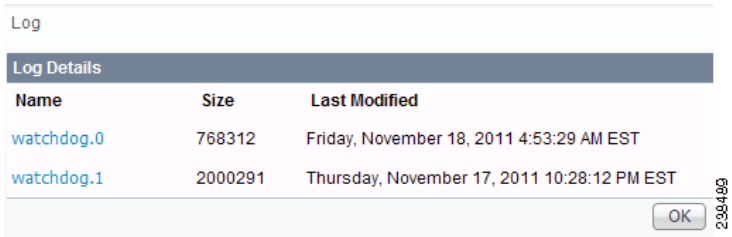
Watchdog

To view the log information about WatchDog, follow these steps:

Step 1 From the Control Center Hosts window, check a check box next to a hostname for which you want to know the WatchDog logs and then click the **Watchdog** button.

A window as shown in Figure 14-4, "WatchDog Logs," appears.

Figure 14-4 WatchDog Logs



Step 2 Click on a specific WatchDog log name in the **Name** column to get the contents of that log. You can filter the information in this log. Click **OK** to return to Figure 14-4.

Step 3 You can repeat the process in Step 2 or click **OK** to return to the Control Center Hosts window.

Logs

To view install and uninstall logs for the Master server, follow these steps:

Step 1 From the Control Center Hosts window, be sure that no check boxes are checked.

Step 2 Click the **Logs** drop-down list and select **Install** or **Uninstall**.

The window that appears is the log of installations or uninstallations, dependent on your selection in Step 2.

Step 3 Click the link in the **Name** column to view the detailed log information.

Step 4 Click **OK** to return to the window.

Step 5 Click **OK** again to return to the Control Center Hosts window.

Licensing

Choose **Administration > Control Center > Licensing**.

To install license keys, follow these steps:

Step 1 Choose **Administration > Control Center > Licensing**, and a window as shown in Figure 14-5, appears.

Figure 14-5 Choose Administration > Control Center > Licensing

Step 2 From the **Installed Licenses** table, click the **Install** button, as shown in Figure 14-5. The Installed Licenses table explains the current statistics. The columns of information tell the **Type** of license keys you have installed (which can include ACTIVATION, API-L2VPN, API-L3MPLS, L2VPN, L3MPLS/VPN,MPLSDIAG, TE, TE/BRG, TE/RG, VPLS, VPN); the **Size**, which is valid for the **ACTIVATION** (licensed maximum global count of services), **TE** (number of TE-enabled nodes), or the **VPN** (maximum number of VPNs licensed); the **Usage**, which gives the number currently used for the rows; and the **Date Updated**, which reflects the refresh of the license usage (on an hourly basis, by default).

Note When you purchase Traffic Engineering Management (TEM), you automatically receive **TE**, **TE/BRG**, and **TE/RG** licenses. All of these licenses *must* be installed to have access to all the Cisco Prime Fulfillment TEM features, including Planning Tools for protection planning (backup tunnels). The **TE** license serves as an activation license for the maximum number of TE-enabled nodes to be managed by TEM (you purchase licenses and upgrade licenses based on a range of nodes); the **TE/RG** license enables primary tunnel placement; and the **TE/BRG** license enables the Fast ReRoute (FRR) protection function.

Note Click **Refresh** to give the most current status.

Step 3 In the resulting Enter License Key window, enter a **License Key** that you received on your *Right to Use* paperwork with your product.

Step 4 Click **Save**.

Your newly installed license appears in an updated version o f the Installed License table, as shown in Figure 14-5.

Step 5 Repeat Step 2, Step 3, and Step 4 for each of the *Right to Use* documents shipped with your product.

Note When you receive multiple *Right to Use* documents to upgrade either the ACTIVATION License, which activates and sets the maximum global count of the services, or VPN licenses, which activates and set the maximum number of VPNs, be sure to enter the licenses in the correct order. For example, if you are upgrading from 500 to 3000 global count of the services and there are two steps to get there, enter the license to upgrade from 500 to 1500 and then the license key to upgrade from 1500 to 3000.

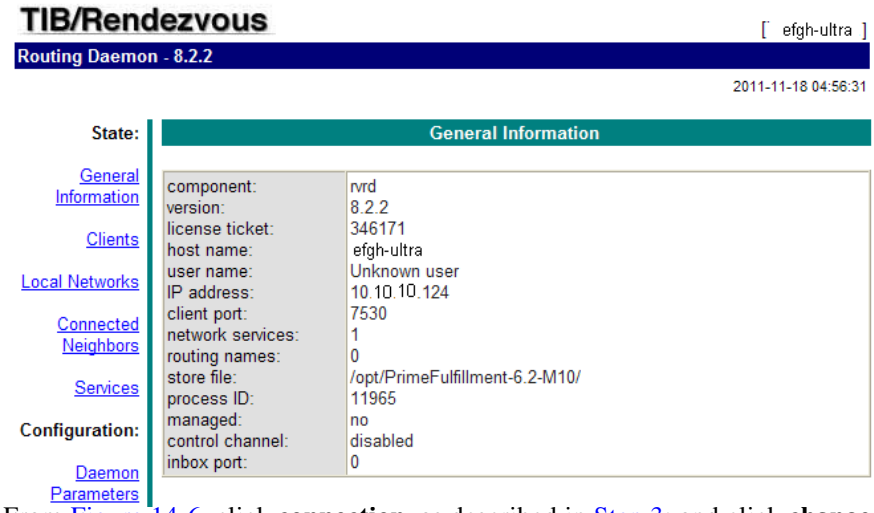
Manage TIBCO Rendezvous

The only reason you would ever use this functionality is if you change the TIBCO ports for TIBCO Rendezvous Agent (rva) or TIBCO Rendezvous Routing Daemon (rvrd) after installation. The changes being made here only affect the topology tool, a Java WebStart application.

Choose **Administration > Tibco > Tibco Manager** and follow these steps:

Step 1 After you choose **Administration > Tibco > Tibco Manager**, a window appears as shown in Figure 14-6.

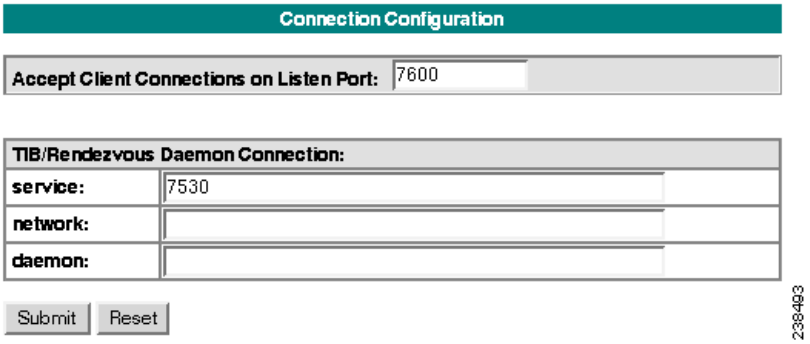
Figure 14-6 TIBCO Rendezvous



Step 2 From Figure 14-6, click **connection**, as described in Step 3; and click **change state**, as described in Step 4. These are choices in the left column of Figure 14-6.

Step 3 In Figure 14-6, when you click **connection**, a window appears as shown in Figure 14-7.

Figure 14-7 Connection Configuration



If you must change the **rva** port number from the existing value, change the **Accept Client Connections on Listen Port:** field to your new rva port number for Prime Fulfillment. If you must change the **rvrd** port number from the existing value, change the **service** field to your new rvrd port number for Prime Fulfillment. Then click **Submit**. Then Figure 14-7 returns with the new value and a note that says "Configuration change will take effect after RVA is re-activated. To re-activate RVA set it into idle state and then back to active state."

Step 4 In Figure 14-6, click **change state**, follow the instructions, and you complete this functionality.

Step 5 From a terminal window, change to the **bin** directory of your Prime Fulfillment installation, such as **/opt/PrimeFulfillment/bin**.

Step 6 Source the Prime Fulfillment environment file in the **$PRIMEF_HOME/bin** directory:

- For K Shell or Bash - use the command **$PRIMEF_HOME/bin/vpnenv.sh**

Step 7 To start the script, at the command line type **updateWebStartJars**.

The next time you start a Java WebStart, such as the topology tool, these changes are in effect.

Manage Security

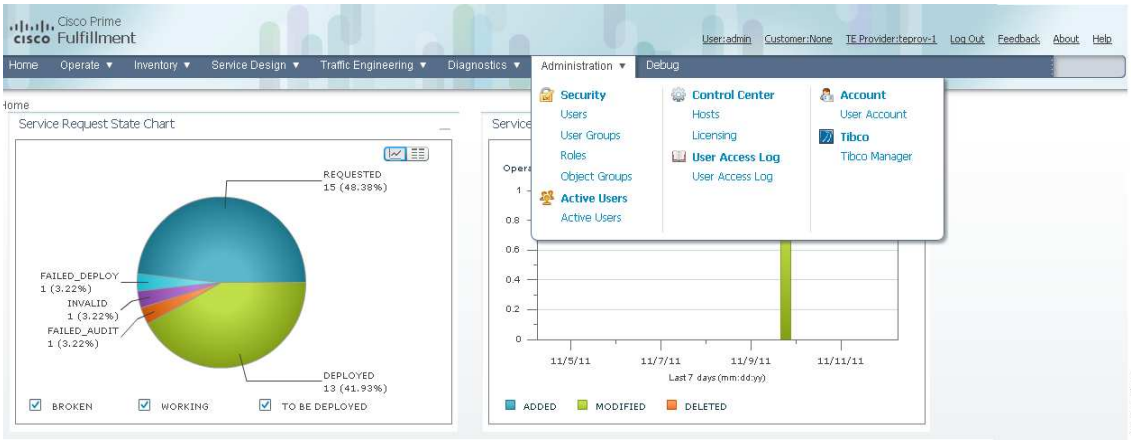
This section describes how system administrators create, edit, and delete users, user groups, user roles, and object groups and how privileges are assigned to these entities.

The security features are only accessible to the user **admin** or users with the following roles:

- **SysAdminRole**—Gives access to all the Prime Fulfillment tools. This is similar to "root" in a UNIX system.
- **UserAdminRole**—Gives access to only the user management tools.

Choose **Administration > Security** to access the user management tools. The window shown in Figure 14-8, appears.

Figure 14-8 Administration, Security Window



You can choose one of the following options:

- Users, page 14-9—To manage users.
- User Groups, page 14-14—To manage user groups.
- User Roles, page 14-16—To manage user roles.
- Object Groups, page 14-20—To manage object groups.

For an example of how to use the Users, User Groups, User Roles, and Object Groups, see the "User Roles Design Example" section on page 14-23.

Users

Choose **Administration > Security > Users** and the Users window appears.

The explanations of the buttons are given as follows:

- Details, page 14-10—View a User Detail Report
- Create, page 14-10—Create a new user
- Copy, page 14-13—Make a copy of an existing user and make changes to create a new user
- Edit, page 14-13—Edit selected user

- Delete, page 14-13—Delete selected user(s).

Details

When you click the **Details** button, located at the bottom of the Users window, you receive the following columns of information: **User ID**; **User Group** that a user belongs to; **Role** that a user occupies; **Resource Privilege** permissions that a user has for each role occupied; **Object Group** that a user role is associated with; **Customer View** that a user's role is limited to; **Provider View** that a user's role is limited to.

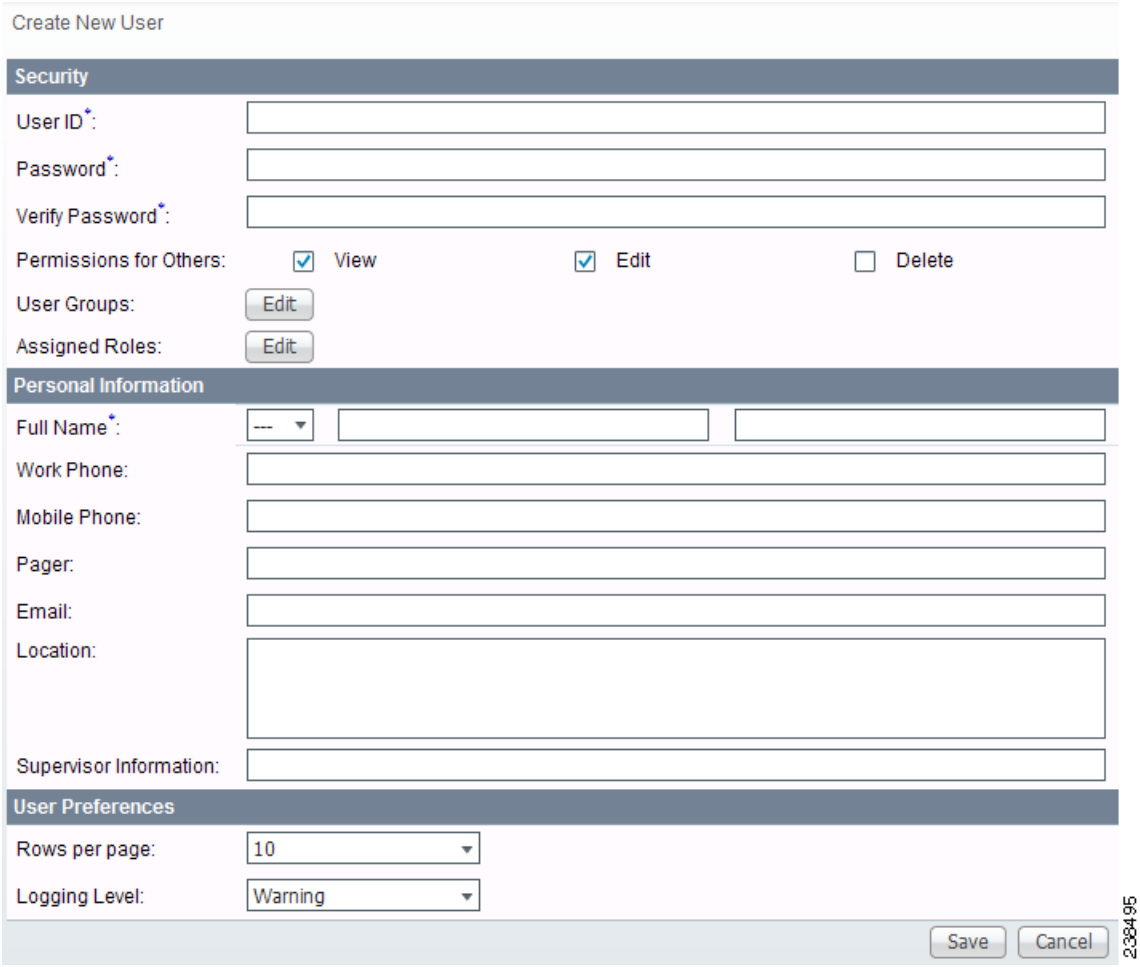
Create

When you click the **Create** button, located at the bottom of the Users window, a user with the required privileges can create a new user. Follow these steps:

Step 1 Choose **Administration > Security > Users**.

Step 2 Click the **Create** button and the window shown in Figure 14-9, appears.

Figure 14-9 Create/Copy/Edit Users Window



Step 3 Enter information in the **Security** section, as follows:

- **User ID** (required)—Enter a User ID for this new user.
- **Password** (required)—New password to replace any existing password:
  - Prime Fulfillment requires a non-blank password.
  - Prime Fulfillment passwords must be a minimum of five characters and no practical maximum length.
  - Prime Fulfillment does not employ any password restrictions or complexity rules; use good judgment in determining passwords.
  - Prime Fulfillment passwords are encrypted when stored in the repository.
  - Prime Fulfillment passwords do not expire.
  - Prime Fulfillment monitors inactivity and auto-logoff per the settings defined in the Dynamic Component Properties Library (DCPL) properties for **repository/rbac**, see Appendix B, "Property Settings."
- **Verify Password** (required)—Confirm by re-entering the selected password.

- **Permission for Others**—Check each of the associated check boxes for the permission that the user (to be created) wants to give to other users. The user who creates the object is the owner of the objects. The creator can allow or disallow other users to **View**, **Edit**, and/or **Delete** the objects owned by the creator by defining permissions. This is the last line of defense. For UserA to delete an object X that UserB created, UserA must first have Delete permission for object X, then UserB's settings for permissions for others is checked, to finally decide whether UserA can delete object X. Permission for others can be enabled or disabled by setting the property: **repository.rbac.checkCreatorPermissionEnabled**. After you make a change, you must restart the WatchDog by entering **stopwd** followed by **startwd**. For more WatchDog details, see Appendix C, "WatchDog Commands".
- **User Groups**—Click **Edit** and you receive a list of the groups. Add this user to a user group(s). The user inherits all the roles assigned to the group(s). You can filter this list. From the selected groups, check the check box next to each group to which you want to add this user. Then click **OK**. You can repeat this procedure if you want to change your selection.

  A user's group membership can also be changed in the group editor (see the "Edit" section on page 14-15).
- **Assigned Roles**—Click **Edit** and you receive a list of the roles. You can filter this list. From the selected roles, check the check box next to each role to which you want to assign this user. Then click **OK**. You can repeat this procedure if you want to change your selection.

  The user inherits all the privileges from the groups in which it participates and from the roles assigned to it. That is, the permissions received by the user is an OR result of the permissions in each role.

Step 4 Enter information in the **Personal Information** section, as follows:

- **Full Name** (required)—Click the drop-down list and select a title; enter the first name; and then enter the last name.
- **Work Phone** (optional)—Enter the work phone number.
- **Mobile Phone** (optional)—Enter the **user's cell phone or mobile phone number.**
- **Pager** (optional)—Enter the user's pager number.
- **Email** (optional)—Enter the user's e-mail address.
- **Location** (optional)—Enter the user's location.
- **Supervisor Information** (optional)—Enter information about the supervisor.

Step 5 Enter information in the User Preferences section, as follows:

- **Language** (optional)—Click the drop-down list to select a language (at this time only English is supported).
- **Rows per page** (optional)—This defines the number of rows per page for object listing. The default is **10**. The choices are: **5**, **10**, **20**, **30**, **40**, **50**, **100**, **500**, **1000**, and **2500**.
- **Logging Level** (optional)—The default is **Warning**. The choices are: **Off**, **Severe**, **Warning**, **Config**, **Info**, **Fine**, **Finer**, **Finest**, and **All** (see all levels of logs). This defines the logging level for viewing logging events. The list progresses from the least number of messages to the most number of messages.
- **Initial Screen** (optional)—The default is **Home**. The choices are: **Home**, **Service Inventory**, **Service Design**, **Monitoring**, **Administration**, **Site Index**, and **Diagnostics**. This is a way to specify the first window you will see after logging in.

Step 6 Click **Save**.

The Users window reappears with the new user listed.

Copy

The **Copy** button, located at the bottom of the Users window, provides a convenient way to create a new User by copying the information for an existing User including User Groups, Assigned Roles, and User Preferences. Follow these steps:

Step 1 Choose **Administration > Security > Users**.

Step 2 Check one check box for the existing User you want to copy and edit to create a new User.

Step 3 Click the **Copy** button and the window shown in Figure 14-9, appears.

Required entries are a **User ID**, **Password**, **Verify Password, and Full Name**.

Step 4 Make all the other changes you want by following the instructions in the "Create" section on page 14-10.

Step 5 Click **Save** and you will return to the Users window.

The newly created **User** is added to the list and a Status Succeeded message appears in green.

Edit

The **Edit** button, located at the bottom of the Users window, allows a user with the required privileges to edit user-specific information. Follow these steps:

Step 1 Choose **Administration > Security > Users**.

Step 2 Check the check box for the row of the user you want to edit.

Step 3 Click the **Edit** button and a window as shown in Figure 14-9, appears.

Note To change your password without the SysAdmin or UserAdmin privileges, click the **Account** tab on the top of the Home page. This allows the user to edit the user profile, including changing the password.

Step 4 Enter the desired information for the user profile, as specified in the **"Create" section on page 10**.

Step 5 Click **Save**.

The Users window reappears with the edited user listed.

Delete

The **Delete** button, located at the bottom of the Users window, allows a user with the required privileges to delete user-specific information. Follow these steps:

Step 1 Choose **Administration > Security > Users**.

Step 2 Check the check box(es) for the row(s) of the user(s) you want to delete.

Step 3 Click the **Delete** button and a confirmation window appears.

Step 4 Click **Delete** to continue with the process of deleting information for the specified user(s). Otherwise click **Cancel**.

The Users window reappears. If this was successful, the newly updated information appears and a **Status** box appears in the lower left corner of the window with a green check mark for **Succeeded**.

User Groups

A user group is a logical grouping of users with common privileges. The **User Groups** feature is used to create, edit, or delete user groups.

To access the User Groups window, choose **Administration > Security > User Groups**. The User Groups window appears.

The explanations of the remainder of the buttons is given as follows:

- Create, page 14-14—Create a new user group
- Edit, page 14-15—Edit selected user group
- Delete, page 14-15—Delete selected user group(s)

Create

The **Create** button, located at the bottom of the User Groups window, allows a user with the required privileges to create a user group. Follow these steps:

Step 1 Choose **Administration > Security > User Groups**.

Step 2 Click the **Create** button and the window shown in Figure 14-10, appears.

Figure 14-10 Create/Edit User Groups Window

Step 3 Enter information for the user group profile, as follows:

- **Name** (required)—Enter a name for the new user group.

- **Description** (optional)—Enter a description of this new user group.
- **Roles**— This allows you to assign roles to this user group. Click **Edit** and you receive a list of the roles. You can filter this list. From the selected roles, check the check box next to each role you want to attach to this user group. Then click **OK**. You can repeat this procedure if you want to change your selection.
- **Users**—This allows you to add users to this user group. Click **Edit** and you receive a list of the users. You can filter this list. From the selected users, check the check box next to each user you want to attach to this user group. Then click **OK**. You can repeat this procedure if you want to change your selection.

Step 4 Click **Save**. The User Groups window reappears with the new user group listed.

Edit

The **Edit** button, located at the bottom of the User Groups window, allows a user with the required privileges to edit user group-specific information. Follow these steps:

Step 1 Choose **Administration > Security > User Groups**.

Step 2 Check the check box for the row of the user group you want to edit.

Step 3 Click the **Edit** button and a window as shown in Figure 14-10, appears.

Step 4 Enter the desired information for the user group profile, as specified in Step 3 of the **"Create" section on page 14-14**.

Step 5 Click **Save.**

The User Groups window reappears with the edited user group list.

Delete

The **Delete** button, located at the bottom of the User Groups window, allows a user with the required privileges to delete user group-specific information. Follow these steps:

Step 1 Choose **Administration > Security > User Groups**.

Step 2 Check the check box(es) for the row(s) of the user group(s) you want to delete.

Step 3 Click the **Delete** button and a confirmation window appears.

Step 4 Click **Delete** to continue the process of deleting information for the specified user group(s). Otherwise click **Cancel**.

The User Groups window reappears. If this was successful, the newly updated information appears and a **Status** box appears in the lower left corner of the window with a green check mark for **Succeeded**.

User Roles

A user role is a predefined or a user-specified role defining a set of permissions. The **User Roles** feature is used to create, edit, or delete user roles.

To better understand the way roles are managed, certain specific characteristics of roles are defined as follows:

- **Parent Role**—All permission of the parent roles are inherited by the role that is being created or edited (child role). A child role always has the same or more privileges than its parent role.
- **Customer**—If a role is associated with a customer, a user of this role does not have access to the objects associated with other customers. Object types that are constrained by customer view are: Persistent Task, Customer Site, VPN, CPE, SR, Policy, Service Order, and resource pools that are associated with a Customer, Customer Site, or VPN.
- **Provider**—If a role is associated with a provider, a user of this role does not have access to the objects associated with other providers. Object types that are constrained by provider view are: Persistent Task, Access Domain, Region, PE, Policy, and some resource pools that are associated with a provider, Access Domain, Region, or PE.

Customer view and provider view within a role have no affect on those objects that do not belong to either a customer or a provider. Those object types are: task, probe, workflow, device, Prime Fulfillment host, and template.

Permission operation types in a Role editor, namely View, Create, Edit, and Delete mean View, Create, Modify, and Delete a database object. For example, SR modification (or subsumption) is viewed as Role Based Access Control (RBAC) Creation. SR purge is viewed as RBAC Delete.

A Role can be enabled to be associated with Object Group(s). When Object Group association is enabled, a Role can no longer be associated with a Customer or a Provider, and it cannot have a Parent Role. Resources are limited to PE, CPE, and Named Physical Circuit only. PE and CPE permission implies Device Permission.

Note A global policy, the one that is not associated with any customer or provider, is accessible by both customer-view roles and provider-view roles.

Separate provider-view from customer-view roles when defining a role. When a role is associated with a provider, choose only the resources for which an access scope can be constrained by a provider view. Do the same for a customer-view role.

To access the User Roles window, choose **Administration > Security > Roles**. The User Roles Administration window appears.

The predefined roles are provided with associated permissions that cannot be edited or deleted. They are intended to cover most of the needed use cases to facilitate a rapid assignment of roles to users and groups with minimum manual configuration. They can also be used as examples to create new roles.

The explanations of the buttons is as follows:

- Create, page 14-17—Create a new user role
- Copy, page 14-19—Copy selected user role
- Edit, page 14-20—Edit selected user role
- Delete, page 14-20—Delete selected user role(s)

Create

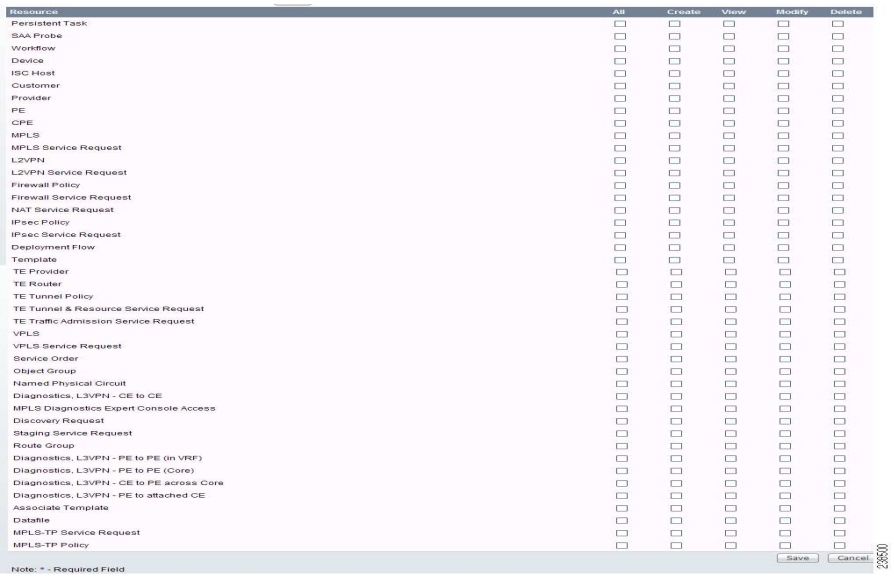
The **Create** button, located at the bottom of the User Roles Administration window, allows a user with the required privileges to create a new user role. Follow these steps:

Step 1 Choose **Administration > Security > Roles**.

Step 2 Click the **Create** button and a window comprised of Figure 14-11 and Figure 14-12, appears.

Figure 14-11 Create/Copy/Edit User Roles Window (Top)

Figure 14-12 Create/Copy/Edit User Roles Window (Bottom)



Step 3 Enter the following information in Figure 14-11:

- **Name** (required)—Enter the name of this new user role.
- **Enable Object Group Association**—The default is that this check box is unchecked. In this case, **Parent Role**, **Customer**, and **Provider** are enabled and **Object Groups** is not enabled. A complete list of resources appears, as shown in the example in the User Roles Administration window. If you

check this check box, **Parent Role**, **Customer**, and **Provider** are not enabled and **Object Groups** is enabled. A window, as shown in Figure 14-12, is reduced to just **PE**, **CPE**, and **Named Physical Circuit**.

- **Parent Role** (optional)—Click **Edit** and a list of the existing roles appears, similar to the User Roles Administration window, from which you can click the radio button for the parent role you choose. Then click **Select**. You can repeat this procedure if you want to change your selection. Click the **Clear** button if you want no parent selection.
- **Customer** (optional)—Click **Edit** and a list of the existing customers appears. You can filter this list. From the selected customers, click the radio button for the customer you want to select to own this role. Then click **Select**. You can repeat this procedure if you want to change your selection. Click the **Clear** button if you want no customer selection.

Note A customer can only be associated with a logical device, such as **CPE** and **PE**. This is not possible with a physical device, such as **device**.

- **Provider** (optional)—Click **Edit** and a list of the existing providers appears. You can filter this list. From the selected providers, click the radio button for the provider you want to select to own this role. Then click **Select**. You can repeat this procedure if you want to change your selection. Click the **Clear** button if you want no provider selection.
- **Object Groups** (optional)—Click **Edit** and a list of the existing object groups appears. You can filter this list. From the selected object groups, check the check box(es) for the object group(s) you want to associate with this User Role. Then click **OK**. You can repeat this procedure if you want to change your selection. Deselect the **Enable Object Group Association** button is you want no object group selection.
- **Description** (optional)—Enter the descriptive information about permissions in this field, as shown in the Description column of the User Roles Administration window.
- **Users** (optional)—Click **Edit** and a list of the existing users appears. You can filter this list. From the selected users, check the check box(es) for the user(s) you want assigned to this role. Then click **OK**. You can repeat this procedure if you want to change your selection.

Note A user who is associated with a specific role cannot see objects associated with other customers or with other providers.

- **User Groups** (optional)—Click **Edit** and a list of the existing user groups appears. You can filter this list. From the selected user groups, check the check box(es) for the user group(s) you want assigned to this role. Then click **OK**. You can repeat this procedure if you want to change your selection.

Step 4 In Figure 14-12, click any combination of the following permissions: **Create**; **View**; **Modify**; **Delete**. If you want all the permissions, click **All**.

Note **Prime Fulfillment Host** refers to **Administration > Control Center > Hosts**. Here, you can view host details, perform configuration tasks, start and stop servers, activate a watchdog, and so on.

Note **SAA Probe** is intended for management of SLA under **Inventory > Device Tools > SLA**. **Any user who wants** to generate SLA reports *must* have **View** permission on **Prime Fulfillment Host** in addition to **View** permission on **SAA Probe**.

Note The **Workflow** object is currently not used.

Note **Template** controls the template manager functions and **Associate Template** controls the ability to associate templates with service requests. If you choose **Create** permission in Template, you also automatically receive **Modify** permission. If you choose any or all permissions in **Associate Template**, you automatically turn on the **View** permission in **Template**.

Note **Datafile** permission allows you to manage datafiles and list all Service Requests associating the datafile. If you choose any or all permissions in **Datafile**, you automatically turn on the **View** permission in **Template**.

Step 5 Click **Save**.

The User Roles Administration window reappears with the new user role listed.

Copy

The **Copy** button, located at the bottom of the User Roles Administration window, provides a convenient way to copy the information from an existing User Role and edit it to create a new User Role. Follow these steps:

Note All fields in the existing role are copied to the new role, even including Users and User Groups. You should edit the new role *carefully* to reflect your intention.

Step 1 Choose **Administration > Security > Roles**.

Step 2 Check one check box for the existing User Role you want to copy and edit to create a new User Role.

Step 3 Click the **Copy** button and the window comprised of Figure 14-11 and Figure 14-12 appears.

Step 4 The required entry is a **Name**. A default name is given, **Copy of** and the name of the original User Role. You cannot duplicate a **Name**.

Step 5 Make all the other changes you want by following the instructions in the "Create" section on page 14-17.

Step 6 Click **Save** and you will return to the User Roles Administration window.

The newly created **User** is added to the list and a Status Succeeded message appears in green.

Edit

The **Edit** button, located at the bottom of the User Roles Administration window, allows a user with the required privileges to edit user role-specific information. Follow these steps:

Step 1 Choose **Administration > Security > Roles**.

Step 2 Check the check box for the row of the user role you want to edit.

Step 3 Click the **Edit** button and a window appears combining Figure 14-11 and Figure 14-12 for this user role.

Step 4 Enter the desired information for the user role profile, as specified in Step 3 and Step 4 of the "Create" section on page 14-17.

Step 5 Click **Save**.

The User Roles Administration window reappears with the edited user roles listed.

Delete

The **Delete** button, located at the bottom of the User Roles Administration window, allows a user with the required privileges to delete user role-specific information. Follow these steps:

Step 1 Choose **Administration > Security > Roles**.

Step 2 Check the check box(es) for the row(s) of the user role(s) you want to delete.

Step 3 Click the **Delete** button and a confirmation window appears.

Step 4 Click **Delete** to continue with the process of deleting information for the specified user role(s).

The User Roles Administration window reappears. If this was successful, the newly updated information appears and a Status box appears in the lower left corner of the window with a green check mark for **Succeeded**.

Otherwise click **Cancel**.

Object Groups

An Object Group is a named aggregate entity comprised of a set of objects. The object types can be PE, CE, Named Physical Circuit (NPC), and interfaces of PEs or CEs. An Object Group provides instance level of access granularity for users.

An Object Group can be associated with different roles. A role can be associated with an Object Group or it can be associated with a grouping of Customer and Provider, but it cannot be associated with both of these. The association with a grouping of Customer and Provider is either with Customer(s), with Provider(s), or with Customer(s) and Provider(s). When a role is associated with Object Group(s), you can only define permissions for PE, CE, and NPC. Permissions on interfaces is implied PEs or CEs, that is, PE Create or CE Create implies Interface Create. PE or CE Edit implies Interface Create, Edit, or Delete. CE or PE Delete implies Interface Delete.

When instance level of access is desired for PE, CE, NPC, or interface of PEs and CEs, you can usually define a role associated with Object Group(s) that contains a collection of PEs and CEs you are limited to operate. Then define other roles to include permissions on other types of objects. See the "User Roles Design Example" section on page 14-23.

If an Object Group contains PEs (or CEs) only, with no explicit interface as a group member, you can access all interfaces of grouped PEs or CEs. If an Object Group contains any explicit interface as group members, every single interface that you want to access you must manually choose to include as group members.

Note Permissions are the union of all roles that you occupy. If your intention is to limit access to a scope of devices or Named Physical Circuits (NPCs), define a role to be associated with Object Group(s), Device, CE, PE, and NPC.

To access the Object Groups window, choose **Administration > Security > Object Groups**. The Object Groups window appears.

The explanations of the buttons is as follows:

- Create, page 14-17—Create a new object group
- Edit, page 14-20—Edit a selected object group
- Delete, page 14-20—Delete selected object group(s)

Create

The **Create** button, located at the bottom of the Object Groups window, allows a user with the required privileges to create a new object group. Follow these steps:

Step 1 Choose **Administration > Security > Object Groups**.

Step 2 Click the **Create** button and the window appears as shown in Figure 14-13.

Figure 14-13 Create/Edit Object Group Window

Step 3 Enter the following information in Figure 14-13:

- **Name** (required)—Enter the name of this new object group.

- **Description** (optional)—Enter a description of this new object group.
- **PE Group Members** (optional)—Click **Edit** and a list of the existing PEs appears. You can filter this list. From the selected PEs, check the check box(es) for the PE(s) you want to include in this group. Then click **OK**. You can repeat this procedure if you want to change your selection(s). The **Interface Members** column will be empty. All existing interfaces for each of the PE Groups in the **Name** column will default to be members of the group unless you select only a subset. To limit the interfaces and select a subset of interfaces, click a PE Group in the **Name** column.You receive a list of all the interfaces for that PE from which you can individually select only the interfaces you want to associate with that PE Group. Then click **OK**. You return to Figure 14-13 and the **Name** and selected **Interface Members** for each PE Group Member appear. If no entries exist in the **Interface Members** column for both **PE Group Members** and **CE Group Members**, the default is all existing interfaces for both (if any exist).
- **CE Group Members** (optional)—Click **Edit** and a list of the existing CEs appears. You can filter this list. From the selected CEs, check the check box(es) for the CE(s) you want to include in this group. Then click **OK**. You can repeat this procedure if you want to change your selection(s). The **Interface Members** column is empty. All existing interfaces for each of the CE Groups in the **Name** column default to be members of the group unless you select only a subset. To limit the interfaces and select a subset of interfaces, click a CE Group in the **Name** column.You receive a list of all the interfaces for that CE from which you can individually select only the interfaces you want to associate with that CE Group. Then click **OK**. You return to Figure 14-13 and the **Name**, and selected **Interface Members** for each CE Group Member appear. If no entries exist in the **Interface Members** column for both **CE Group Members** and **PE Group Members**, the default is all existing interfaces for both (if any exist).
- **NPC Group Members** (optional)—Click **Edit** and a list of the existing NPCs appears. You can filter this list. From the selected NPCs, check the check box(es) for the NPC(s) you want to select to own this role. Then click **OK**. You can repeat this procedure if you want to change your selection(s). You return to Figure 14-13 and the **Name** for each NPC Group Member appears.

Step 4 Click **Save**.

Figure 14-13 reappears with the new object group listed.

Edit

The **Edit** button, located at the bottom of Figure 14-13, allows a user with the required privileges to edit object group-specific information. Follow these steps:

Step 1 Choose **Administration > Security > Object Groups**.

Step 2 Check the check box for the row of the object group you want to edit.

Step 3 Click the **Edit** button and a window appears as shown in the Object Groups window, with the object group chosen specified in the **Name** field.

Step 4 Enter the desired information for the object group, as specified in Step 3 of the "Create" section on page 14-21.

Step 5 Click **Save**.

The Object Groups window reappears with the edited object groups listed.

Delete

The **Delete** button, located at the bottom of the Object Groups window, allows a user with the required privileges to delete object group-specific information. Follow these steps:

Step 1 Choose **Administration > Security > Object Groups**.

Step 2 Check the check box(es) for the row(s) of the object group(s) you want to delete.

Step 3 Click the **Delete** button and a confirmation appears.

Step 4 Click **Delete** to continue with the process of deleting information for the specified object group(s).

The Object Groups window reappears. If this was successful, the newly updated information appears and a Status box appears in the lower left corner of the window with a green check mark for **Succeeded**.

Otherwise click **Cancel**.

User Roles Design Example

This section gives an example situation, an illustration that shows this setup, and steps on how to setup this design:

Example

This section explains an example data center for which the following sections, "Illustration of Setup" section on page 14-24 and "Steps to Set Up Example" section on page 14-25 give an illustration setup and steps, respectively.

Finance Customer XYZ built an MPLS network to connect its branch offices to its data center. Subsidiaries of XYZ are running different parts of the MPLS network. Each subsidiary uses a different BGP AS domain, which results in different Provider Administrative Domains (PADs) inside Prime Fulfillment.

Each subsidiary acts as a Provider and owns therefore its own Devices, like PE and CE devices, and should also own logical attributes inside Prime Fulfillment, like Regions, Sites, Customers, and VPNs. Therefore, the view of the devices for each subsidiary must be separated into PAD views. Thus, Provider A cannot manipulate or view the configuration files for devices of Provider B. Devices are not shared between PADs.

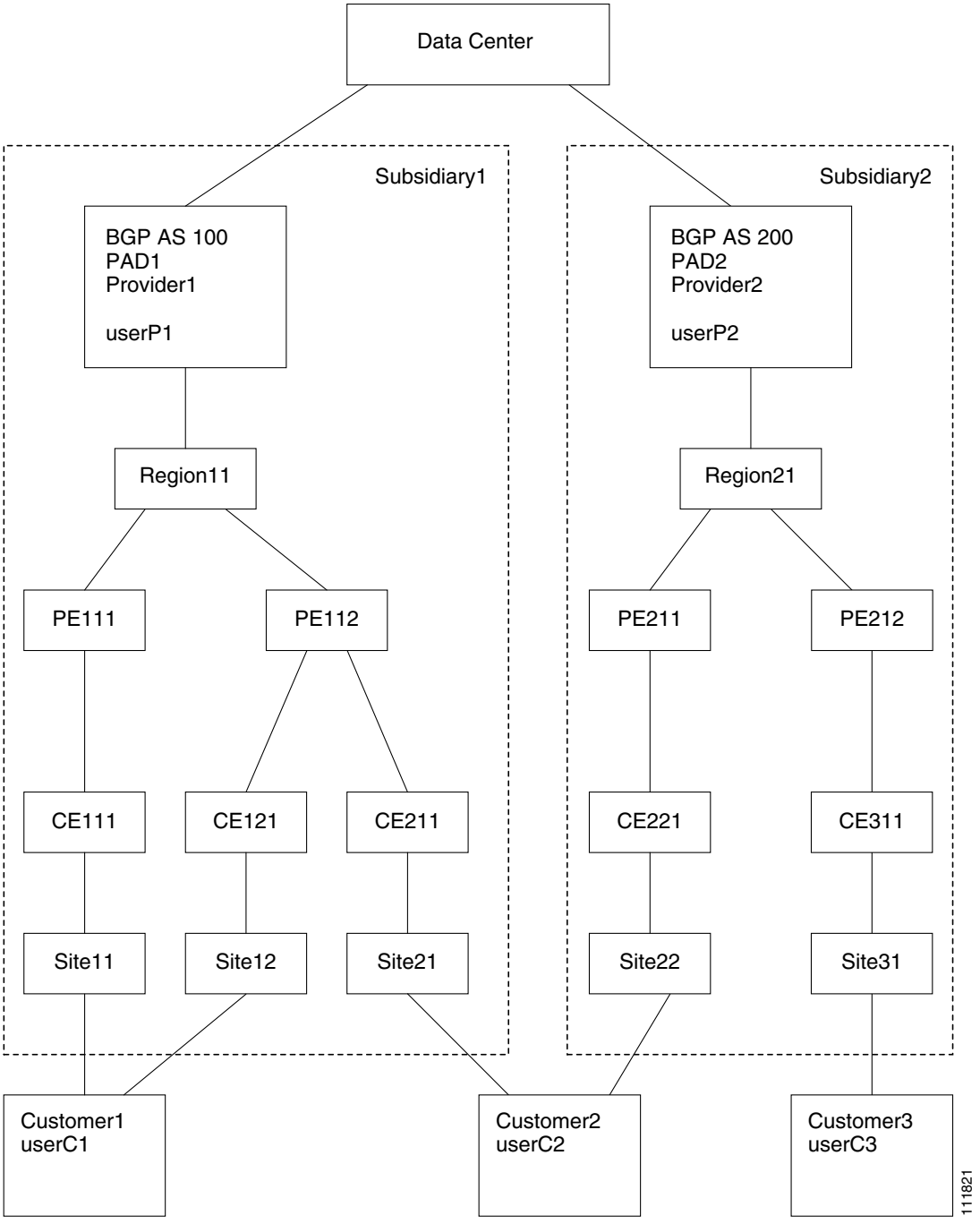
Inside a PAD, there are Customers with sites and VPNs with only local significance. Also, the IP addressing should be defined per PAD.

But there are also Customers that have sites in different PADs. This means that there is a need for Inter-AS VPNs. The Provider who owns the Customer should also have the right to share this Customer with other Providers. In this case, the VPNs and Route Targets should be shared between the providers.

Illustration of Setup

Figure 14-14 shows the setup described in the "Example" section on page 14-23.

Figure 14-14 Contents in Example



Steps to Set Up Example

This section explains the steps to create the example explained in the "Example" section on page 14-23 and shown in the "Illustration of Setup" section on page 14-24.

Step 1 Create the following Object Groups (see the "Create" section on page 14-21, which is for the section Object Groups):

- P1PEGroup that has members PE111 and PE112
- P2PEGroup that has members PE211 and PE212
- C1CEGroup that has members CE111 and CE121
- C2CEGroup that has members CE211 and CE221
- C3CEGroup that has the member CE311
- C2DeviceGroup that has members PE112, CE211, PE211, and CE221
- C3DeviceGroup that has members PE212 and CE311.

Step 2 Create the following User Roles that are associated with one or more groups created in Step 1 (see the "Create" section on page 14-17, which is for the section User Roles.

- P1DeviceGroupRole, associated with groups P1PEGroup, C1CEGroup, and C2CEGroup, and have the Modify and Delete permissions on for PE and Cpe.
- P2DeviceGroupRole, associated with groups P2PEGroup, C2CEGroup, and C3CEGroup, and have the Modify and Delete permissions on for PE and Cpe.
- C1DeviceGroupRole, associated with groups P1PEGroup, C1CEGroup, and have the Modify permission on for PE and the Modify and Delete permissions on for Cpe.
- C2DeviceGroupRole, associated with group C2DeviceGroup, and have the Modify permission on for PE and the Modify and Delete permissions on for Cpe.
- C3DeviceGroupRole, associated with group C3DeviceGroup, and have the Modify permission on for PE and the Modify and Delete permissions on for Cpe.

Step 3 Create the following User Roles that have Customer View or Provider View, as explained in the "User Roles" section on page 14-16.

- P1MplsRole, associated with Provider P1, and have permissions on Provider, Task, Prime Fulfillment Host, Mpls SR, Mpls Policy, NPC, and Probe. (Add Service, Template, and ServiceOrder if needed.)
- P2MplsRole, associated with Provider P2, and have permissions on Provider, Task, Prime Fulfillment Host, Mpls SR, Mpls Policy, NPC, and Probe. (Add Service, Template, and ServiceOrder if needed.)
- C1MplsRole, associated with Customer C1, and have permissions on Customer, Task, Prime Fulfillment Host, Mpls SR, Mpls Policy, NPC, and Probe. (Add Service, Template, and ServiceOrder if needed.)
- C2MplsRole, associated with Customer C2, and have permissions on Customer, Task, Prime Fulfillment Host, Mpls SR, Mpls Policy, NPC, and Probe. (Add Service, Template, and ServiceOrder if needed.)
- C3MplsRole, associated with Customer C3, and have permissions on Customer, Task, Prime Fulfillment Host, Mpls SR, Mpls Policy, NPC, and Probe. (Add Service, Template, and ServiceOrder if needed.)

Step 4 Assign the User Roles defined in Step 2 and Step 3 to Users, as explained in the "Users" section on page 14-9.

- User P1 has User Roles: P1DeviceGroupRole, P1MplsRole, C1MplsRole, and C2MplsRole.
- User P2 has User Roles: P2DeviceGroupRole, P2MplsRole, C2MplsRole, and C3MplsRole.
- User C1 has User Roles: C1DeviceGroupRole and C1MplsRole.
- User C2 has User Roles: C2DeviceGroupRole and C2MplsRole.
- User C3 has User Roles: C3DeviceGroupRole and C3MplsRole.

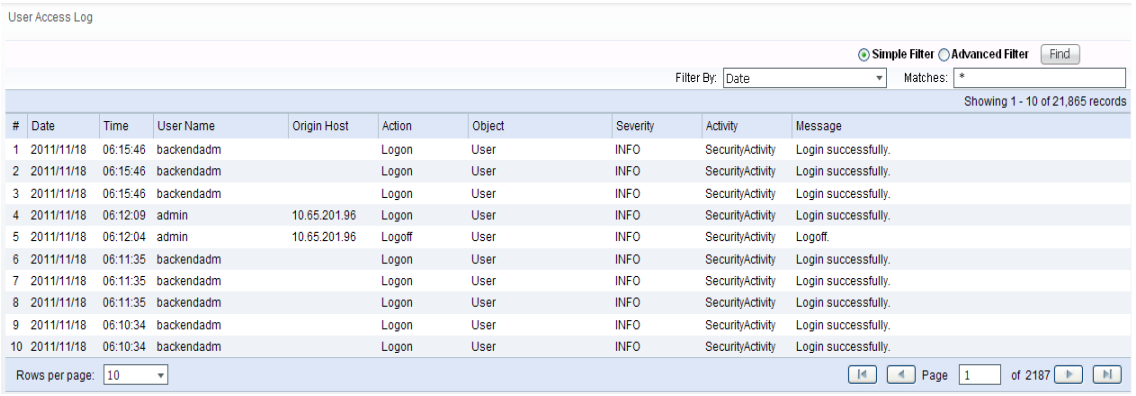
User Access Log

This section shows a detailed report of every activity by every user.

Choose **Administration > User Access Log > User Access Log** and follow these steps:

Step 1 After you choose **Administration > User Access Log > User Access Log**, a window appears as shown in Figure 14-15.

Figure 14-15 User Access Log Viewer with Simple Filter



All the log information about user actions appears.

Note The types of activities or objects to be logged can be configured. This can be done directly through SQL. By default, security-related activities and activities on objects listed in the Role editor are logged.

Step 2 The default **Simple Filter** radio button is selected. To filter using the **Simple Filter**, continue with Step 3. To filter using **Advanced Filter**, proceed to Step 5.

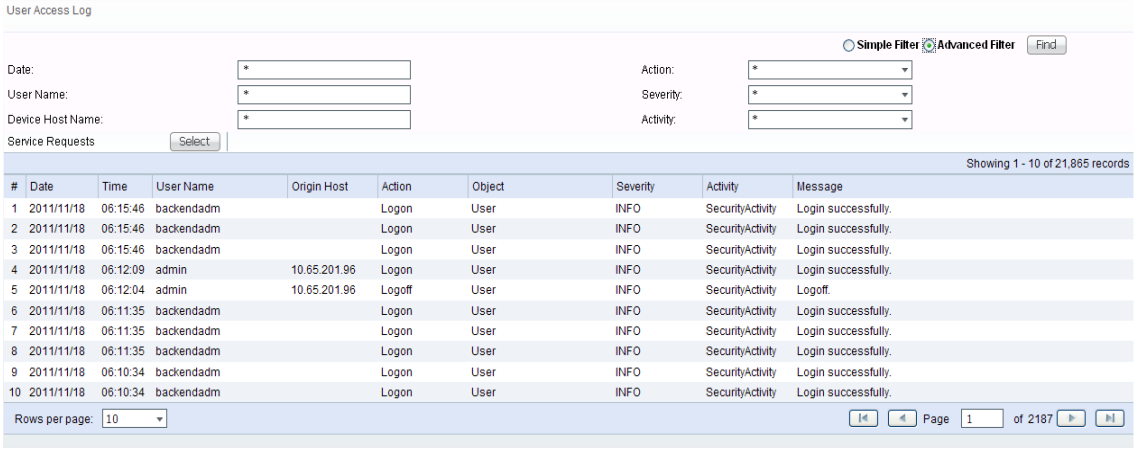
Step 3 To filter the information with **Simple Filter**, keep the **Simple Filter** radio button selected and from **Filter By**, choose: **Date**, **User Name**, **Origin Host**, **Action**, **Severity**, or **Activity** (also column names). For **Matches**, enter the beginning characters of what you want to match followed by *. Then click **Find**. The result is that only the log information matching the entered filter appears.

Step 4 To exit this log report, choose another feature from the main product tabs.

Step 5 To filter the information with **Advanced Filter**, click the **Advanced Filter** radio button.

A window as shown in Figure 14-16 appears.

Figure 14-16 User Access Log Viewer with Advanced Filter



All the log information about user actions appears.

Step 6 Enter filter information you want to match in one or more of the following categories and then click **Find**.

Note When you choose multiple filters, the log results that appear are only the ones that match all the specified filter information.

- **Date** Enter the beginning characters of the date you want to view followed by a ***,** in the format given in the **Date** column.
- **User Name** Enter the beginning characters of the specific **User Name** you want to view followed by a *****.
- **Device Host Name** Enter the beginning characters of the specific **Host Name** you want to view followed by a *****.
- **Action** Click the drop-down list and choose from: **UNKNOWN**; **View**; **Create**; **Modify**; **Delete**; **Logon**; **Logoff**; **Session Timeout**. If you decide not to use this filter, just keep *****.
- **Severity** Click the drop-down list and choose from: **UNKNOWN**; **INFO**; **WARNING**; **ERROR**. If you decide not to use this filter, just keep *****.
- **Activity** Click the drop-down list and choose from: **UNKNOWN**; **SecurityActivity**; or **UserActivity**. The result is that only the log information matching the entered filter appears.

Step 7 **Service Requests** has a selection of **Select/Deselect**. Click this and you receive a list of Service Requests in the system from which you can check check box(es) for the User Access Log to handle. Then click the **Select** button. These Service Requests then appear on Figure 14-16.

Step 8 To exit this log report, choose another feature from the main product tabs.

APPENDIX A

Cisco Configuration Engine Server

Note The Cisco Configuration Engine Server is referred to as IE2100 throughout the Cisco Prime Fulfillment user interface. The IE2100 appliance referenced within Prime Fulfillment represents any server configured to run the Cisco Configuration Engine software. This server can be either the IE2100 appliance itself for all supported software versions prior to 2.0 or a Solaris workstation for all supported software versions from 2.0 and beyond.

Cisco Prime Fulfillment supports the Cisco CNS IE2100 Device Access Protocol for communication with any Cisco IOS device, such as uploading a configuration file from a device, downloading a configlet to a device, or executing a command on a device and obtaining a result. Prime Fulfillment also supports CNS Plug-and-Play.

To use the Cisco CNS IE2100 functionality on Prime Fulfillment, you must first set up the Cisco CNS IE2100 appliance and the Prime Fulfillment workstation as explained in an appendix in the *Cisco Prime Fulfillment Installation Guide 6.2*.

This appendix includes the following sections. Implement these sections in sequence:

Note The "Using Plug-and-Play" section on page A-3 is optional.

Creating a Cisco CNS IE2100 Appliance

Prime Fulfillment supports multiple Cisco CNS IE2100 appliances. To create a Cisco CNS IE2100 appliance, follow these steps:

Note For more information, see the Devices, page 2-1 section of Chapter 2, "Before Setting Up Prime Fulfillment."

Step 1 Choose **Inventory > Physical Inventory > Devices**.

The Device window appears.

Step 2 Click the **Create** button.

Step 3 From the **Create** menu, click **IE2100**.

The Create IE2100 Device window appears.

Step 4 Enter the **Device Host Name** and if applicable, the IE2100 **Device Domain Name**. The **Description** field is optional. If the Cisco CNS IE2100 appliance is not registered with DNS, then you *must* enter the **IP Address** of the Cisco CNS IE2100 appliance. Click **Save**.

The Device window reappears with the IE2100 listed as a device.

Creating a Cisco IOS Device Using the Cisco CNS Device Access Protocol

Each Cisco CNS IE2100 appliance can serve multiple Cisco IOS devices. A Cisco IOS device can only be served by one Cisco CNS IE2100 appliance. To create a Cisco IOS device using the Cisco CNS Device Access Protocol, follow these steps:

Note For more information, see the Devices, page 2-1 section of Chapter 2, "Before Setting Up Prime Fulfillment."

Step 1 Choose **Inventory > Physical Inventory > Devices**, and the Device window appears.

Step 2 Click the **Create** button.

Step 3 From the **Create** menu, click **Cisco Device**.

The Create Cisco Device window appears.

Step 4 In the **General** section, enter the **Device Host Name** and **Device Domain Name**.

For **CNS Device Access Protocol**, you do not need to define the parameters in the **Login User** and **Login Password** sections.

For the **Device and Configuration Access Information** section, you must choose **CNS** for the **Terminal Session Protocol**.

For the **Device and Configuration Access Information** section, the only valid **OS** selection is **IOS**. **IOS XR** is not supported for Cisco CNS IE2100 appliances with Prime Fulfillment.

Step 5 Click the **Show** button for **Additional Properties** at the bottom of the window and this window expands to add the additional information.

The following steps pertain to the **Terminal Server** and **CNS Options** section**.**

Step 6 Check the **Fully Managed** check box if you want the device to become a fully managed device. For fully managed devices, Prime Fulfillment sends e-mail notifications upon receipt of device configuration changes originated outside Prime Fulfillment and schedules enforcement audit tasks upon detection of possible intrusion.

Note Be sure to set the DCPL parameters for e-mail and Fully Managed, as explained in the "Config" section on page 14-3. Choose **Administration > Control Center > Hosts**. Choose a Host and then click **Config**. Then in the TOC in the left column, be sure to enter appropriate information in the following fields:

SYSTEM > email > from; **SYSTEM > email > smtpHost**; **SYSTEM > fullyManaged > auditableCommandsFileLocation (if information is not given here, all commands are audited)**; **SYSTEM > fullyManaged > enforcementAuditScript**; and **SYSTEM > fullyManaged > externalEventsEmailRecipients**.

Note Verify that the **cns config notify** command is configured for the IOS device. This command ensures that configuration change events, which are the basis of the fully-managed feature, are sent out on the event bus. If this command is not configured on the device, the fully-managed feature will not work, because there will be no config-changed events reaching Prime Fulfillment.

Step 7 Specify the **Device State**, as follows:

- Choose **ACTIVE** (the default)—f the router is physically present on the network.
- Choose **INACTIVE**—If the router is not yet physically present on the network.

Step 8 Specify the **Device Event Identification**, as follows:

- Choose **HOST_NAME**—If the **Device Host Name** as defined in Step 4 is to be used as the **CNS Identification** for this device.
- Choose **CNS_ID**—If the device CNS Identification string is other than the **Device Host Name**.
- If you have selected **CNS_ID** as the **Device Event Identification**, you must enter the **CNS Identification** parameter in the field labeled **CNS Identification**. This must be a unique argument. It is used to create the device in the corresponding Cisco CNS IE2100 repository and to listen to events pertaining to this device.

Note Verify that the **cns id string {CNS_ID} event** command is configured for the IOS device. If this command is not present on the device, the IE2100 will not send out any events on the bus using this CNS ID, and hence communication with the device will fail.

Step 9 Select the Cisco CNS **IE2100** appliance that serves this Cisco IOS device. Select one entry from the drop-down list of IE2100 devices already defined in the repository.

Step 10 Use the drop-down list for **CNS Software Version** to choose the version of Cisco CNS Configuration Engine that manages the IOS device (1.3, 1.3.1, 1.3.2, 1.4, 1.5, 2.0, 3.0, or 3.5).

Step 11 Use the drop-down list for **CNS Device Transport** to choose HTTP or HTTPS as the transport mechanism used by Prime Fulfillment to create, delete, or edit devices in the IE2100 repository. If HTTPS is used, the Cisco CNS Configuration Engine must be running in secure mode.

Step 12 Click **Save**. The Device window reappears with the Cisco IOS device listed.

Using Plug-and-Play

Prime Fulfillment supports the Plug-and-Play device configuration through a Cisco CNS IE2100 appliance. Prime Fulfillment supports devices not physically present on the network.

The procedures for using Plug-and-Play when the Cisco IOS device is not physically present on the network vary depending on whether there is an initial configuration file for the device.

Follow these steps if the Cisco IOS device *does not* have an initial configuration file:

Step 1 Create a Cisco IOS Device as described in the "Creating a Cisco IOS Device Using the Cisco CNS Device Access Protocol, page A-2" section.

Step 2 Define the Cisco IOS device properties.

Be sure to specify the **Device State** as **INACTIVE** because the device is not physically present on the network

Step 3 Click **Save**.

A Cisco IOS Device entry is created in the Prime Fulfillment repository and in the corresponding Cisco CNS IE2100 appliance repository.

If the Cisco IOS device *does* have an initial configuration file, import the initial configuration file into Prime Fulfillment using the Inventory Manager functionality, explained in Chapter 13, "Using Inventory Manager" in this manual.

Be sure to specify the **Device State** as **INACTIVE** because the device is not physically present on the network.

The Inventory Manager create a Cisco IOS Device entry in the Prime Fulfillment repository. Also, it creates an entry in the corresponding Cisco CNS IE2100 repository, and associates the specified initial configuration file with this new device in the Cisco CNS IE2100 repository.

You can provision the newly created inactive Cisco IOS Device for different services. Because the device is not physically present on the network, Prime Fulfillment saves the configlets associated with these services in its repository and tries to download them to the device only after the device has come up. Until the device is physically present on the network, the service request goes into the **WAIT_DEPLOY** state. The service requests are explained in the user guides for each of the services.

After the device comes up and connects to its corresponding Cisco CNS IE2100 appliance, the device retrieves and applies its initial configuration if there is one waiting for it in the Cisco CNS IE2100 repository.

Prime Fulfillment detects that the device has come onto the network and performs the following actions:

- Changes the Cisco IOS Device state from **INACTIVE** to **ACTIVE**.

  Prime Fulfillment performs a collect config of the IOS device and stores it in the Prime Fulfillment repository.

- Verifies whether any Prime Fulfillment service has been waiting for this device to come up and tries to download the corresponding configlets to the device to complete the service request.

APPENDIX B

Property Settings

To navigate to the properties, known as Dynamic Component Properties Library (DCPL), in the Graphical User Interface (GUI) navigate to the tab **Administration > Control Center > Hosts**. Then select a check box for a specific host and click the **Config** button. These updates are effective only for this session.

None of these properties can be set on a per user basis, including logging.

Note More details about this are explained in the "Config" section on page 14-3.

When you click on the folder or subfolder, it expands to more subfolders or eventually to the property itself. Then you receive an explanation, default values, and in some cases range and rules. This table can help you understand all the properties available at a glance. The properties are listed alphabetically. When a / ends an entry, this means it can be expanded further. Also, if you are searching for a property and do not know the name, you can use some key words and do a Find on the pdf version.

Table B-1 DCPL Properties

Property	Default Value	Range/Rules	Explanation
AutoDiscovery Properties:			Controls the operation of Autodiscovery.
/DiscoveryTemplateFolder	/Discovery	string	Template folder under which the templates to be discovered for MPLS VPN Discovery will reside.
/TopologyHandler	Default	string	This property points to the topology handler for the discovery run.
/createVpnAndCustomerFromVRFName	true	The valid values are **true** and **false**.	This property controls whether the VPN and Customer objects can be created from the VRF names. This is valid only in certain scenarios when Service Providers have maintained such a mapping.
/performTemplateDiscovery	false	The valid values are **true** and **false**.	With this flag, the user can control the template discovery. For performance reasons, if the template discovery is not desired this should be set to false.
Cleanup Properties:			Cleans up various system resources such as log files and temporary files.
/Cleanup/TaskLogs/			This component cleans up old TaskLogs.

Table B-1 DCPL Properties (continued)

maxAgeInHours	168	integer	Maximum age of the TaskLogs in hours. TaskLogs older than this age will be deleted during the next cleanup cycle. Set to 0 to disable this feature.
sleepIntervalInHours	24	integer, 1-1000 hours	Time in hours for taskLog cleanup service to sleep between clean up cycles.
/Cleanup/Tasks/			This component cleans up old TaskLogs.
maxAgeInHours	0	integer	Maximum age of the Tasks in hours. Tasks that have not been modified in over maxAge hours and that have no Active schedules will be deleted during the next cleanup cycle. Set to 0 to disable this feature.
sleepIntervalInHours	24	integer, 1-1000 hours	Time in hours for task cleanup service to wait between clean up cycles. Changing this value initiates an immediate cleanup cycle.
/Cleanup/TempFiles/			This component cleans up old temporary files.
maxAgeInHours	168	integer	Maximum age of the temporary files in hours. Temporary files older than this age will be deleted during the next cleanup cycle. Set to 0 to disable this feature.
sleepIntervalInHours	24	integer, 1-1000 hours	Time in hours for tempFile cleanup service to sleep between clean up cycles.
/Cleanup/logLevel	CONFIG	selection	This log Level is used only if there is no log Level defined for a component. The log Level is the level at which logging is done for this component. These levels are identical to the logging levels defined for JDK1.4 logging package. The levels in descending order are: SEVERE (highest value) WARNING INFO CONFIG FINE FINER FINEST (lowest value).
CNS Properties:			
defaultVersion	1.4	1.3, 1.3.1, 1.3.2, 1.4, 1.5, 2.0, and 3.0	Default version of CNS to be selected while creating a device. The supported versions are: 1.3, 1.3.1, 1.3.2, 1.4, 1.5, 2.0, and 3.0.

Table B-1 DCPL Properties (continued)

deprecatedReboot	0	The valid values are **0** and **1**.	This is the flag to be used for reloading IOS 12.3 devices using cisco.mgmt.cns.config.reboot CNS event. Value 0 means IOS 12.3 devices may not be rebooted using cisco.mgmt.cns.config.reboot CNS event. So, IOS versions other than 12.3 can be rebooted. Value 1 means only IOS 12.3 devices are rebooted using cisco.mgmt.cns.config.reboot CNS event.
DCS Properties:			Device Configuration Service. This component corresponds to a library that is used by Prime Fulfillment to communicate with network devices using protocols such as telnet, ssh, tftp, and so forth.
/DCS/FTP/			FTP Settings.
ftpPassword		string	Password for FTP server login, used by DCS and GTL.
ftpRootDirectory		string	FTP root directory, used by DCS and GTL.
ftpServer		string	FTP Server host name or IP address, used by DCS and GTL.
ftpSubDirectory		string	FTP sub directory, used by DCS and GTL.
ftpUsername		string	Username for FTP server login, used by DCS and GTL.
/DCS/IOSUsePrimaryWarningExprOnly	false	The valid values are **true** and **false**.	If **true**, DCS uses only the primary warning expression list, specified in DCS/IOSWarningExpressions. If **false**, DCS uses the primary list specified in DCS/IOSWarningExpressions for add and modify operations and uses the list specified in DCS/ IOSWarningExpressionsRemoveCfg during delete (decommissioning) operations.

Table B-1 DCPL Properties (continued)

/DCS/IOSWarningExpressions		string	IOS warning expressions that can be safely ignored; case insensitive; . matches any char except newline, * means zero or more, + means one or more, ? means zero or one. All regular expressions except the last one should have a **$** at the end of the regular expression. %Aborting Save. Compress the config$.*Access Rules Download Complete$ % Access VLAN does not exist.$ Address aliases with.*$ % All RSA Keys will be removed.$ % All router certs issued using these keys will also be removed.*$ % Already found same .* statement in this profile$ % A profile is deemed incomplete until it has match identity statements$.*certificate accepted$ Certificate request sent$.?Changes to the System MTU will not take effect until the next reload.*$ CNS config partial agent is running already$ % Configuration buffer full, can't add command.*$.*Crypto EzVPN does not exist.*$ Enter configuration commands, one per line$ Explicit Path name .*$ % Generating .* bit RSA keys$ Global .* will be Port Address Translated.*$ Global Ethernet MTU is set to.*$ If the interface doesn't support baby giant frames.*$ Increasing .* burst size to$ % Interface .* IP address .* removed due to enabling VRF$ % Interface .* IP address .* removed due to disabling VRF$ % IP addresses from all interfaces in VRF .*have been removed$

Table B-1 DCPL Properties (continued)

/DCS/IOSWarningExpressions (Continued)		string	% IP routing table V.* does not exist. Create first$ % IP routing table g.*does not exist. Create first$ % No CEF interface information$ %No matching route to delete$ %Translation not found$.*Not all config may be removed and may reappear after reactivating$ ^%.?NOTE:$ OSPF: Unrecognized virtual interface .* Treat it as loopback stub route$ outside interface address added$ % Profile already contains this keyring$ %PVC is already defined$ Restarting RADIUS authentication service on port .* $ Restarting RADIUS accounting service on port .*$ Redundant .* statement$ security level for .* changed to$.*Service policy .* is already attached$ % Signature RSA Keys not found in configuration.$.*success$ The .*command will also show the fingerprint$ %The static routes in .* with outgoing interface .* will be removed$ Unable to disable parser cache$ % Unknown VPN$.* Unknown VRF specified$ % VRF .* does not exist or does not have a RD$.?warning.*
/DCS/IOSWarningExpressionsExitCfgMode		string	IOS warning expressions that can be safely ignored when exiting config term mode; regular expression must match whole warning message; for messages that wrap more than one line replace line terminations (CR and/or LF chars) with a single space character; replace each variable field with the meta-character sequence \\S+ that will match a single group of non-whitespace chars; literals are case insensitive; use $ to separate entries.

Table B-1 DCPL Properties (continued)

/DCS/IOSWarningExpressionsRemoveCfg		string	IOS warning expressions that can be safely ignored during decommissioning; case insensitive; . matches any char except newline, * means zero or more, + means one or more, ? means zero or one.
/DCS/RCP/			RCP Settings.
rcpDirectory	/tmp	string	Directory to use for uploaded/downloaded config files.
/DCS/SSH/			SSH Client Settings.
overWriteSSHKeys	true	The valid values are **true** and **false**.	Overwrite SSH Keys: If **true**, will allow new keys to overwrite existing keys in the key file for a given host. If **false**, an error will be displayed if host sent key does not match the server sent key.
sshEncryptionCipher	3DES->DES	selection	Cipher to use for SSH Encryption/Decryption; requires restart on change. Values: 3DES->DES first tries 3DES then if not available falls back to DES; 3DES, only tries 3DES; DES, only tries DES.
/DCS/SSHv2/			SSHv2 Client Settings.
overWriteSSHv2Keys	true	The valid values are **true** and **false**.	Overwrite SSHv2 Keys: If **true**, will allow new keys to overwrite existing keys in the key file for a given host. If **false**, an error will be displayed if host sent key does not match the server sent key.
/DCS/TFTP/			TFTP Settings.
tftpCreateFileOnServerBeforeUpload	true	The valid values are **true** and **false**.	Some TFTP servers require a file to exist on the server with write access before a TFTP client can upload it. This is sometimes called write-replace or overwrite mode. Other TFTP servers require a that a file NOT exist, this is sometimes called write-create or no overwrite mode. When **true**, DCS will create the file on the TFTP server before uploading device configuration.
tftpRootDirectory	/tftpboot	string	TFTP Root Directory used by DCS and GTL.
tftpServerIPAddress		string	TFTP Server host name or IP Address used by DCS and GTL must be the same as that of the Prime Fulfillment server.
tftpSubDirectory		string	TFTP Sub Directory used by DCS and GTL.
/DCS/XR			IOS XR properties.

Table B-1 DCPL Properties (continued)

WarningExpressions	^.?.?warning$	string	IOS XR warning expressions that can be: safely ignored; case insensitive; . matches any character except newline, where: * means zero or more, + means one or more, ? means zero or one.
commitConfigTimeout	120	integer, 30-600	Maximum time in seconds to commit config target buffer to running config.
maxRetriesEnterCfgExcIMode	3	integer, 0-10	Maximum number of times to retry entering configure exclusive mode. 0 = no retries. Retry delay interval is fixed at 30 seconds.
/DCS/allowCommandDownloadOnError	false	The valid values are **true** and **false**.	Continue command download on error.
/DCS/cnsEventTimeout	120	integer, 0-120 seconds	CNS event wait time in seconds
/DCS/configUploadTimeout	300	integer, 60-900	Maximum time in seconds to wait for a device configuration to be uploaded.
/DCS/customPasswordPrompt	Password:		Device custom password prompt.
/DCS/customUsernamePrompt	Username:		Device custom username prompt.
/DCS/getCommitCLIConfigAfterDownload	true	The valid values are **true** and **false**.	Retrieve the committed CLI configuration after an XML configuration download. If the default of **true** is set, whenever a Service Request is deployed on an IOS XR device, a transaction is created. This transaction gets the configlet deployed in the CLI mode and stores it in the repository. This creation of a new transaction adds to the time of Service Request deployment. If this property is set to **false**, no transaction to retrieve the CLI configlet is created.
/DCS/logLevel	CONFIG	selection	The log Level is the level at which logging is done for this component. These levels are identical to the logging levels defined for JDK1.4 logging package. The levels in descending order are: SEVERE (highest value) WARNING INFO CONFIG FINE FINER FINEST (lowest value).
/DCS/maxDeviceConnectCompleteTime	60	integer, 15-600 seconds	Maximum time in seconds to wait for a terminal session connection to a device.

Table B-1 DCPL Properties (continued)

/DCS/maxDeviceConnectRetryCount	3	integer, 0-5	Maximum number of times to retry connecting to a device when the maxDeviceConnectCompleteTime expires. 0= no retries.
/DCS/maxOperationTimeout	30	integer, 5-300 minutes	Maximum time in minutes to wait for a device operation to complete.
/DCS/maxPromptTimeout	60	integer, 15-300 seconds	Maximum time in seconds to wait for a prompt during a terminal session with a device.
/DCS/maxSocketReadTimeout	30	integer, 10-300 seconds	Maximum time in seconds to wait for data on a socket connection read operation.
/DCS/misc			Miscellaneous settings.
ConfigForMergeXML		string, file name	Configuration file to be used for the merging of two XMLs.
allowPromptCharsInBanner	false	The valid values are **true** and **false**.	Controls if prompt characters, such as # and >, are allowed in banners. If **true**, a minimum of 2 seconds (default of loginSocketReadTimeout) is added to each login. Note that selecting this option requires “aaa authentication attempts login n” to be set to a minimum of 2.
loginSocketReadTimeout	2	integer, 1-45	Number of seconds to WAIT for a login authentication username or password prompt. Applicable if DCS\misc\allowPromptCharsInBanner is **true**. Increasing this value slows down device logins and counts against DCS\maxDeviceConnectCompleteTime who’s default is 60 seconds.
readBufferSize	32	integer, 4-96	Size in KBytes of the buffers used while reading device input streams with telnet and SSH. Increasing size might improve performance. Decrease size if there are memory issues.
DeploymentFlow Property:			Deployment flow Component: Used to create a flow of different types of steps such as mpls.
/DeploymentFlow/logLevel	CONFIG	selection	The log Level is the level at which logging is done for this component. These levels are identical to the logging levels defined for JDK1.4 logging package. The levels in descending order are: SEVERE (highest value) WARNING INFO CONFIG FINE FINER FINEST (lowest value).

Table B-1 DCPL Properties (continued)

Discovery Properties:			Prime Fulfillment auto discovery framework.
/Discovery/DeviceDiscovery			
continueOnError	false	The valid values are **true** and **false**.	A Boolean flag indicating whether device discovery should try to continue on an error. When the value is **true**, device discovery ignores the device and attempts to create other devices discovered. In this case, the device discovery is marked as SUCCESS, but indicates there were errors. The default behavior is device discovery is marked FAILED at the first error encountered. This property applies only to errors encountered during the device creation phase of device discovery like duplicate or missing hostnames in case of CDP and file based discovery options and invalid device configurations or insufficient read permissions for configurations files and so on, for the configuration file based discovery option. Any errors encountered during CDP discovery itself or while parsing XML files still result in the device discovery step being marked as FAILED. WARNING: If this property is set to **true**, discovery continues if there are any device creation errors, ignoring the device that caused the error, but only partial NPCs and services are discovered.
logLevel	CONFIG	selection	The log Level is the level at which logging is done for this component. These levels are identical to the logging levels defined for JDK1.4 logging package. The levels in descending order are: SEVERE (highest value) WARNING INFO CONFIG FINE FINER FINEST (lowest value).

Table B-1 DCPL Properties (continued)

mgmtIpAddressLoopkupPattern		string	A comma separated list of interface name patterns to look for to determine the management IP address of the device discovered using the import configuration option. The configuration is parsed for the interface information, and the first available IP address of the interface from the given list is used as the management IP address of the device. For example, if the IP address of the loopback 0 interface should be used as the management IP address, the value of the property should be set to “loopback0”. If the first available loopback should be used, set the value of the property to “loopback”. A comma separated list can be specified as “Loopback0,Ethernet0”. In this case, the first available IP address among the list of interfaces specified in that order is used as the management IP address.
/Discovery/DataCollection			
continueOnError	false	The valid values are **true** and **false**.	A Boolean flag indicating whether data collection should try to continue on an error. When the value is **true**, the data collection step does not collect discovery data for the failed device, but attempts to collect configuration for other devices discovered. In this case, the configuration collection step is marked as SUCCESS, but indicates there were errors. The default behavior is discovery data collection step is marked FAILED at the first error encountered. WARNING: If this property is set to **true**, discovery continues if there are any collection or parsing errors, ignoring the device that caused the error, but only partial NPCs and services are discovered.
logLevel	CONFIG	selection	The log Level is the level at which logging is done for this component. These levels are identical to the logging levels defined for JDK1.4 logging package. The levels in descending order are: SEVERE (highest value) WARNING INFO CONFIG FINE FINER FINEST (lowest value).

Table B-1 DCPL Properties (continued)

reuseConfigsIfAvailable	false	The valid values are **true** and **false**.	If the Boolean flag is **true**, the discovery data collection step uses the config from the repository if available. If the configs are not in the repository, an attempt is made to contact the device to collect the current running configuration. The default behavior is discovery tries to collect the current running configs from the device.
/Discovery/MPLSService			MPLS services discovery.
logLevel	CONFIG	selection	The log Level is the level at which logging is done for this component. These levels are identical to the logging levels defined for JDK1.4 logging package. The levels in descending order are: SEVERE (highest value) WARNING INFO CONFIG FINE FINER FINEST (lowest value).
/Discovery/MetroEService			Metro Ethernet services discovery.
logLevel	CONFIG	selection	The log Level is the level at which logging is done for this component. These levels are identical to the logging levels defined for JDK1.4 logging package. The levels in descending order are: SEVERE (highest value) WARNING INFO CONFIG FINE FINER FINEST (lowest value).
meConfigParsingRegistry		string	List of handlers to be invoked at collect config time for Metro Ethernet services.
meDiscoverIntraPopVPWS	false	The valid values are **true** and **false**.	Set this to **true** if local switched VPWS services are to be discovered. Do this only if you wish to discover VPWS services switched at NPE. If not, set this to **false** for performance reasons.
/Discovery/NPCDiscovery			NPC discovery.
logLevel	CONFIG	selection	The log Level is the level at which logging is done for this component. These levels are identical to the logging levels defined for JDK1.4 logging package. The levels in descending order are: SEVERE (highest value) WARNING INFO CONFIG FINE FINER FINEST (lowest value).
/Discovery/RoleAssignment			

Table B-1 DCPL Properties (continued)

logLevel	CONFIG	selection	The log Level is the level at which logging is done for this component. These levels are identical to the logging levels defined for JDK1.4 logging package. The levels in descending order are: SEVERE (highest value) WARNING INFO CONFIG FINE FINER FINEST (lowest value).
/Discovery/Workflow			Prime Fulfillment auto discovery workflow.
logLevel	CONFIG	selection	The log Level is the level at which logging is done for this component. These levels are identical to the logging levels defined for JDK1.4 logging package. The levels in descending order are: SEVERE (highest value) WARNING INFO CONFIG FINE FINER FINEST (lowest value).
/Discovery/configs.location	*<vpnsc_tmp>*/ Discovery/ configs		The directory name where the temporary device configurations are stored during the collect config process.
/Discovery/logLevel	CONFIG	selection	The log Level is the level at which logging is done for this component. These levels are identical to the logging levels defined for JDK1.4 logging package. The levels in descending order are: SEVERE (highest value) WARNING INFO CONFIG FINE FINER FINEST (lowest value).
/Discovery/logLocation	*vpnsc_tmp*>/ Discovery/ logs	string	The directory name where discovery logs files are kept.
/Discovery/restart	false	The valid values are **true** and **false**.	With this property, you can clear out all network objects from the repository that was created by the Discovery process and you can restart the Discovery process. Be very cautious in setting this value to **true**.
/Discovery/tmpdir	*<vpnsc_tmp> /Discovery*	string	A directory to store the temporary results of the discovery process.
DistributionFramework Properties:			Distribution Framework. This component handles the distribution of work (jobs) between different servers in a Prime Fulfillment distributed installation.
/DistributionFramework/Dispatcher/			Service that dispatches jobs to workers.
DefaultUnitDuration	1000	integer	The unit duration (in milliseconds) used to estimate jobs without a profile.

Table B-1 DCPL Properties (continued)

PingInterval	1000	integer	The interval (in ms) dispatcher pings the workers to get the load.
ProcessorEpsilon	10	integer	If two proccessors differ in usage by an amount less than this, they are considered identical from the point of view of the load balancer.
ProfileUpdateThreshold	10	integer	The percent change of a profile that triggers an update of the dispatcher.
logLevel	CONFIG	selection	The log Level is the level at which logging is done for this component. These levels are identical to the logging levels defined for JDK1.4 logging package. The levels in descending order are: SEVERE (highest value) WARNING INFO CONFIG FINE FINER FINEST (lowest value).
/DistributionFramework/NamingHost	*<master_server>*	string	The hostname or ip address of the name server.
/DistributionFramework/NamingPort	*<naming_port>*	string	The port of the name server.
/DistributionFramework/RemoteUtil/			Layer abstracting the remote call functionality.
logLevel	CONFIG	selection	The log Level is the level at which logging is done for this component. These levels are identical to the logging levels defined for JDK1.4 logging package. The levels in descending order are: SEVERE (highest value) WARNING INFO CONFIG FINE FINER FINEST (lowest value).
/DistributionFramework/ ServiceLauncher/			Manages the execution of multiple services in the same VM.
logLevel	CONFIG	selection	The log Level is the level at which logging is done for this component. These levels are identical to the logging levels defined for JDK1.4 logging package. The levels in descending order are: SEVERE (highest value) WARNING INFO CONFIG FINE FINER FINEST (lowest value).
/DistributionFramework/ThreadPool/			Thread pool component used by the worker to execute jobs.

Table B-1 DCPL Properties (continued)

logLevel	CONFIG	selection	The log Level is the level at which logging is done for this component. These levels are identical to the logging levels defined for JDK1.4 logging package. The levels in descending order are: SEVERE (highest value) WARNING INFO CONFIG FINE FINER FINEST (lowest value).
/DistributionFramework/Worker/			Worker.
Groups		string	The groups this worker belongs to. This property is deprecated because groups are stored in the database rather than being provided by the worker.
ThreadPoolSize	100	integer, 25-250	The maximum number of threads. Set it to 0 to allow the pool to use as many thread as necessary.
logLevel	CONFIG	selection	The log Level is the level at which logging is done for this component. These levels are identical to the logging levels defined for JDK1.4 logging package. The levels in descending order are: SEVERE (highest value) WARNING INFO CONFIG FINE FINER FINEST (lowest value).
GSAM Property:			Generic Service Access Model to get an XML dump from the repository for the provisioning driver.
/GSAM/logLevel	CONFIG	selection	The log Level is the level at which logging is done for this component. These levels are identical to the logging levels defined for JDK1.4 logging package. The levels in descending order are: SEVERE (highest value) WARNING INFO CONFIG FINE FINER FINEST (lowest value).
GTL Properties:			Generic Transport Layer. This library provides an API to different jobs (such as provisioning, collection etc.) to access Device Configuration Service (DCS). The jobs do not interface with DCS directly (to access the devices), but work with the API provided by GTL.
/GTL/CSL/			Configuration Services Layer
ios/			IOS related properties.

Table B-1 DCPL Properties (continued)

cmdsRequiringDelay		string	List of the IOS commands that execute asynchronously and require time to be processed before they are reflected in the running configuration. Matching rules: case insensitive, .matches any char except newline, * means zero or more, + means one or more, ? means zero or one.
delayAfterDownloadingCmd		command name**:** integer, 0-1800 seconds	List of the IOS commands that require a delay after they are downloaded using a terminal session protocol, such as Telnet. The character **;** delimits the list elements. The IOS command in each list element must be followed by the character **:** followed by a maximum integer of 1800, which indicates the number of seconds to delay, thus indicating 0-1800 seconds (0-30 minutes). The command matching rules: case insensitive, .matches any char except newline, * means zero or more, + means one or more, ? means zero or one. The default is a blank field.
delayBeforeDownloadingCmd			List of the IOS commands that require a delay before they are downloaded using a terminal session protocol, such as Telnet. The character **;** delimits the list elements. The IOS command in each list element must be followed by the character **:** followed by a maximum integer of 1800, which indicates the number of seconds to delay, thus indicating 0-1800 seconds (0-30 minutes). The command matching rules: case insensitive, .matches any char except newline, * means zero or more, + means one or more, ? means zero or one.
delayBeforeUpload		integer, 0-30 seconds	The delay in seconds to wait after downloading a configlet that contains asynchronous commands before uploading the new configuration.
delayBeforeWriteMem	0	integer, 0-300 seconds	The delay in seconds to wait after downloading a configlet before performing a write memory command.
/GTL/PAM/			
args		string	Invocation argument to be used.
className		string	PAM Class name.

Table B-1 DCPL Properties (continued)

usePAM	false	The valid values are **true** and **false**.	When the value is **true**, the selected PAM is used for device authentication. When the value is **false**, the standard authentication credentials are used in the Prime Fulfillment repository for each device.
/GTL/device-config-access-protocol	1	integer, 1-3	Protocol to use for device configuration uploads and downloads. 1= TERMINAL (Use the device-terminal-session-protocol for config access) 2= TFTP 3= FTP.
/GTL/device-terminal-session-protocol	1	integer, 1-2	Protocol to use for device terminal sessions. 1= TELNET 2= SSH.
/GTL/echo-mode	false	The valid values are **true** and **false**.	Flag indicating whether to run GTL in **ECHO** mode or **DCS** mode. Setting Prime Fulfillment to run in echo mode allows Prime Fulfillment to perform Service provisioning tasks without downloading the resulting commands to the physical hardware. The resulting Service Provisioning is stored only in the Repository and no attempt is made to connect to the target devices. When echo mode is enabled (set to **true**), no attempt to audit the Service Request is performed. From a production environment, you are able to perform service provisioning on devices that are either temporarily offline or not yet commissioned. Once these devices become active, you can Force Deploy the already provisioned Service Requests and Prime Fulfillment downloads the configurations.
/GTL/ios/			IOS related GTL properties.
copy-running-to-startup	true	The valid values are **true** and **false**.	Flag indicating whether to copy running config to startup config when downloading configlets. Write Mem flag.
/GTL/logLevel	CONFIG	selection	The log Level is the level at which logging is done for this component. These levels are identical to the logging levels defined for JDK1.4 logging package. The levels in descending order are: SEVERE (highest value) WARNING INFO CONFIG FINE FINER FINEST (lowest value).
GUI Properties:			The component for GUI-based properties.

Table B-1 DCPL Properties (continued)

/GUI/Common/			Generic GUI component. Use it if you do not have any specific component requirements, such as L2VPN.
PeSelectionCategory	DEVICE	selection	When required to select a PE device for tasks such as Prime Diagnostics, there are various ways to filter the devices that are shown. This option allows you to decide the default filter to apply, Device, Region, or Provider.
logFileViewThreshold	10000000	integer	The maximum log file size in bytes that can be viewed in the GUI Log Viewer.
logLevel	FINE	selection	The log Level is the level at which logging is done for this component. These levels are identical to the logging levels defined for JDK1.4 logging package. The levels in descending order are: SEVERE (highest value) WARNING INFO CONFIG FINE FINER FINEST (lowest value).
migrationUpdateCount	1000	integer	The maximum number of interface names updated in the service request and to be committed to the database during the IOS XR migration. This maximum count specifies the maximum number of records to be committed to the database in a cycle during the IOS XR migration so that the database does not overload.
/GUI/EVC/			L2VPN related GUI component. Use it with L2VPN related operations only.
logLevel	SEVERE	selection	The log Level is the level at which logging is done for this component. These levels are identical to the logging levels defined for JDK1.4 logging package. The levels in descending order are: SEVERE (highest value) WARNING INFO CONFIG FINE FINER FINEST (lowest value).
/GUI/L2VPN/			L2VPN related GUI component. Use it with L2VPN related operations only.
logLevel	SEVERE	selection	The log Level is the level at which logging is done for this component. These levels are identical to the logging levels defined for JDK1.4 logging package. The levels in descending order are: SEVERE (highest value) WARNING INFO CONFIG FINE FINER FINEST (lowest value).

Table B-1 DCPL Properties (continued)

/GUI/MPLSOAM/			The MPLS OAM component.
logLevel	FINEST	selection	The log Level is the level at which logging is done for this component. These levels are identical to the logging levels defined for JDK1.4 logging package. The levels in descending order are: SEVERE (highest value) WARNING INFO CONFIG FINE FINER FINEST (lowest value).
/GUI/MplsVPN/			MPLS VPN related GUI component. Use it with MPLS VPN related operations only.
logLevel	SEVERE	selection	The log Level is the level at which logging is done for this component. These levels are identical to the logging levels defined for JDK1.4 logging package. The levels in descending order are: SEVERE (highest value) WARNING INFO CONFIG FINE FINER FINEST (lowest value).
UniqueRTFeatureEnable	false	The valid values are **true** and **false**.	The default value for this property is false. To use the independent RTs for IPv4 or IPv6 feature, you must set the DCPL property to true.
/GUI/Performance/			For monitoring GUI performance.
logLevel	INFO	selection	The log Level is the level at which logging is done for this component. These levels are identical to the logging levels defined for JDK1.4 logging package. The levels in descending order are: SEVERE (highest value) WARNING INFO CONFIG FINE FINER FINEST (lowest value).
GUI/Ping			Ping related GUI component. Use it with Ping related operations only.
logLevel	CONFIG	selection	The log Level is the level at which logging is done for this component. These levels are identical to the logging levels defined for JDK1.4 logging package. The levels in descending order are: SEVERE (highest value) WARNING INFO CONFIG FINE FINER FINEST (lowest value).
/GUI/Topology/			Component related to the web start topology application.

Table B-1 DCPL Properties (continued)

logLevel	CONFIG	selection	The log Level is the level at which logging is done for this component. These levels are identical to the logging levels defined for JDK1.4 logging package. The levels in descending order are: SEVERE (highest value) WARNING INFO CONFIG FINE FINER FINEST (lowest value).
/GUI/VPLS/			VPLS related GUI component. Use it with VPLS related operations only.
logLevel	SEVERE	selection	The log Level is the level at which logging is done for this component. These levels are identical to the logging levels defined for JDK1.4 logging package. The levels in descending order are: SEVERE (highest value) WARNING INFO CONFIG FINE FINER FINEST (lowest value).
/GUI/disableSelectAllForSR	false	The valid values are **true** and **false**.	Disable the select all for the SR list.
/GUI/srRefreshRate	30000	integer	The refresh rate (in milliseconds) for the SR List screen.
/GUI/workflowSteps	<*vpnsc_home*>/etc/workflowSteps.csv	string	The predefined workflow steps.
/GUI/workflows	<*vpnsc_home*>/etc/workflows.csv	string	The predefined workflows.
JavaWebStart Properties:			Java Web Start components.
/JavaWebStart/InventoryManager/			Component to create and manage Devices.
MaxDevicesPerSaveTransaction	25	integer, 1-500	Specifies the maximum number of devices per transaction when performing save operation.
/JavaWebStart/TaskManager/			Component to create and monitor scheduled tasks.
MaxDevicesPerCollectionTask	25	integer, 1-500	Specifies the maximum number of devices per Collect Config task. More devices can be specified for a single task and they will be managed as such from a user perspective. However, there may be more than on Collect Config task created and executed in the repository.

Table B-1 DCPL Properties (continued)

LDAP Properties			LDAP properties is used to create the users, set password, and roles for the user in the LDAP server.
/LDAP/LdapAuthentication/	false	The valid values are **true** and **false**.	Authentication required for using the LDAP server. If the property is set to true then authentication will be done using LDAP server. If the property is set to false then authentication will be done using Prime Fulfillment.
/LDAP/HostName/		string (any number of LDAP servers can be added)	This property is used for establishing the communication with the LDAP server using the IP Address of the LDAP server and the port.
/LDAP/DistinguishedName/		string	This property is used for Authentication of user Credentials. The Distinguished Name is same for multiple LDAP servers.
/LDAP/UserDefinedException/		string (User has the option to give a customized exception else it displays system defined exception)	This property will be used to customize the Exception whenever Ldap server is down.
Logging Properties:			This contains different properties needed by the logging framework. There are a set of default values for logging parameters. These values can be overridden for a specific server.
/Logging/Defaults/			This contains the default values for the logging framework.
logFileNumber	2	integer, 1-10	Maximum number of log files for a process. Each of these files can be of size **logFileSize**. When the maximum number for log files is reached for a process, the log files are rotated by deleting the oldest log file for that process.
logFileSize	2000000	integer, 1000000-10000000 bytes	Size in bytes of a single log file for a process. Each process will have a number of log files (see **logFileNumber** property), where each of these files can grow to this size.
logFormatter	java.util.logging.XMLFormatter	string	Class name for the default formatter of log records.

Table B-1 DCPL Properties (continued)

logLevel	CONFIG	selection	NOTE: This log Level is used only if there is no log Level defined for a component. The log Level is the level at which logging is done for this component. These levels are identical to the logging levels defined for JDK1.4 logging package. The levels in descending order are: SEVERE (highest value) WARNING INFO CONFIG FINE FINER FINEST (lowest value).
logLocation	<*vpnsc_tmp*>	string	The directory name where log files are kept.
/Logging/TaskLogs/			This contains logging properties for task logs.
logLocation	<*vpnsc_tmp*>/ TaskLogs	string	The directory name where all the task logs are kept.
logMessageSize	100	integer, 100-300	This property sets the number of lines of message to be displayed for each log entry.
Provisioning Properties:			Contains properties and components for service provisioning like MPLS VPNs.
/Provisioning/Engine/			Contains properties for the XML driven provisioning engine.
logLevel	CONFIG	selection	The log Level is the level at which logging is done for this component. These levels are identical to the logging levels defined for JDK1.4 logging package. The levels in descending order are: SEVERE (highest value) WARNING INFO CONFIG FINE FINER FINEST (lowest value).
serviceSchema	service.xsd	string	Specifies the XML schema definition file for defining new services.
/Provisioning/NOM/			Network Object Model for parsing and delta generation of configs.
DocumentBuilderFactory/			This contains the properties for the DOM builder factory.
ignoreComments	true	The valid values are **true** and **false**.	Flag.
ignoreWhiteSpace	false	The valid values are **true** and **false**.	Flag for DOM builder factory.
validation	false	The valid values are **true** and **false**.	Flag for validation of xml files.

Table B-1 DCPL Properties (continued)

catSyntaxFile	catSyntax.xml	string	Contains the XML for Catalyst command syntax.
explicitlyRemoveRouteTargets	false	The valid values are **true** and **false**.	Normally (**false**), the "no ip vrfname" automatically cleans up all its subcommands in IOS. There is no need to clean up each one of the subcommands before taking away the parent command. By setting this value to **true**, Prime Fulfillment explicitly cleans up all router target subcommands before removing the "ip vrfname".
iosSyntaxFile	iosSyntax.xml	string	Contains the xml syntax for IOS command.
logLevel	CONFIG	selection	The log Level is the level at which logging is done for this component. These levels are identical to the logging levels defined for JDK1.4 logging package. The levels in descending order are: SEVERE (highest value) WARNING INFO CONFIG FINE FINER FINEST (lowest value).
/Provisioning/PasswordManagement/			User generated Password generation
PasswordFormula/			User generated Password formula generation class
class		string	User generated class file
/Provisioning/ProvDrv/			Contains properties for the XML driven provisioning ProvDrv.
AuditJITUpload	true	The valid values are **true** and **false**.	If the value of this property is set to **false,** the provisioning server does NOT upload a copy of the configuration file from the routers when it processes the Service Request for auditing purpose. Instead, it uses copies of the configuration files that were collected and stored in the Repository earlier. If the value of this property is set to **true**, the provisioning server uploads a copy of the configuration file from the routers when it processes the Service Request for auditing purpose. The default value of this property is **true**.
CleanStagedConfigletWhenForceDeploy	false	The valid values are **true** and **false**.	If this value is **true**, when a service request is force deployed, the staged configlet is removed before provisioning. If this value is the default of **false**, the staged configlet is considered as part of the base configuration during provisioning.

Table B-1 DCPL Properties (continued)

DownloadTemplateToUnmanagedDevice	false	The valid values are **true** and **false**.	If this value is **true**, for an unmanaged device, Prime Fulfillment attempts to download just the template. The configlet generated by the provision is not part of the download. By default, this value is **false** and then there is no attempt to download to an unmanaged device.
ForceTemplateDeploy	false	The valid values are **true** and **false**.	Templates are downloaded in the first/initial service request (SR) deployment. During **edit/modification of SRs**, in case templates are attached to the SR, the following is true: a) Templates are updated if this property is set to **true**, in case the physical interface or VLAN ID (or other Prime Fulfillment repository variables) is modified or changed. b) Templates attached to a service policy are not downloaded to new LINKS added to an existing SR if this property is set to **true** or **false**. Templates are not updated if this property is set to **false**, if the physical interface or VLAN ID (or other Prime Fulfillment repository variables) is modified. If the templates do not include any Prime Fulfillment repository variables, the recommendation is to set this property to **false**. If the templates include any Prime Fulfillment repository variables, the recommendation is to set this property to **true**.
MaxNumberOfDevicesPerDownload	100	integer	Prime Fulfillment will try to bundle as much devices as possible during a download attempt. This value set the max number of devices allowed during such an attempt. If the number of devices exceeds this limit, multiple download attempts will take place. You should decrease this limit if the download involves many devices with huge configlets in order to conserve memory usage.

Table B-1 DCPL Properties (continued)

NegateTemplateDeploy	true	The valid values are **true** and **false**.	If the value of this property is set to true, then the negate template will append or prepend, depending on the template association in the service request. If the value of this property is set to false, then the negate template will always prepend.
ProvisionJITUpload	true	The valid values are **true** and **false**.	If the value of this property is set to **false**, the provisioning server does NOT upload a copy of the configuration file from the routers when it processes the Service Request for provisioning purpose. Instead, it uses copies of the configuration files that were collected and stored in the Repository earlier. If the value of this property is set to **true**, the provisioning server uploads a copy of the configuration file from the routers when it processes the Service Request for provisioning purpose.
ProvisioningBatchSize	10	integer, 0-2147483647	Provisioning Driver divides the requested Service Requests into batches while performing the deployment. This parameter specifies the number of Service Requests that will be processed as a batch.
SaveConfigletsFromAllSRs	true	The valid values are **true** and **false**.	If the value of this property is set to true, for each device in a SR, the provisioning server will save the configlet contributed from all SRs that are processed in the same provisioning run. If the value is set to false, only the configlet contributed by the current SR is saved for this device in this SR even though this same device may be in multiple SRs that are processed by the same provisioning run.
logLevel	CONFIG	selection	The log Level is the level at which logging is done for this component. These levels are identical to the logging levels defined for JDK1.4 logging package. The levels in descending order are: SEVERE (highest value) WARNING INFO CONFIG FINE FINER FINEST (lowest value).
/Provisioning/Service/			Contains different services and their properties.
TE/			Traffic Engineering Provisioning Service related properties section.

Table B-1 DCPL Properties (continued)

enableLogging	true	The valid values are **true** and **false**.	When the value is the default of **true**, debugging of logging is enabled for this service. When the value is **false**, debugging of logging is not enabled for this service.
platform/			Used by ProvDrv
CISCO_ROUTER/			Used by ProvDrv
serviceBladeClass	com.cisco.vpnsc.prov.te.ServiceBlade.TeServiceBlade	string	Identifies ServiceBlade class name for ProvDrv.
sendAuditEvent	true	The valid values are **true** and **false**.	Set **true** to enable sending audit event for this service.
Uds/			User defined services.
platform/			Service platform
CISCO_ROUTER/			Cisco router
serviceBladeClass	com.cisco.vpnsc.prov.uds.UdsServiceBlade	string	Uds Service Blade.
deviceConfig/			
logLevel	CONFIG	selection	The log Level is the level at which logging is done for this component. These levels are identical to the logging levels defined for JDK1.4 logging package. The levels in descending order are: SEVERE (highest value) WARNING INFO CONFIG FINE FINER FINEST (lowest value).
parseConfigAfterProvisioning	false	The valid values are **true** and **false**.	This property controls the parsing of the configuration file after the provisioning is completed in order to make sure that device inventory is in sync with network.
saveDebugData	true	The valid values are **true** and **false**.	If this property is set to **true**, whenever an SR is provisioned, the uploaded config files and input XML data are saved to a temporary directory for debugging purposes.
sendAuditEvent	true	The valid values are **true** and **false**.	Set true to enable sending audit event for this service.
serviceFile	l2vpnService.xml	string	Layer 2 VPN Service definition file.
platform/			Contains properties for L2VPN for different platforms.

Table B-1 DCPL Properties (continued)

CATOS/			Service blade parameters for CATOS.
serviceBladeClass	com.cisco.vpnsc.prov.l2vpn.L2VPNServiceBlade	string	ServiceBladeClass location.
CISCO_ROUTER/			
iosXRConfigType	XML		Config type for IOS XR devices for MPLS service blade
serviceBladeClass	com.cisco.vpnsc.prov.l2vpn.L2VPNServiceBlade	string	ServiceBladeClass location.
logLevel/	SEVERE	selection	The log Level is the level at which logging is done for this component. These levels are identical to the logging levels defined for JDK1.4 logging package. The levels in descending order are: SEVERE (highest value) WARNING INFO CONFIG FINE FINER FINEST (lowest value).
fsm/			MPLS Finite State Machine (FSM) Provisioning.
DownloadWeights/			Specifies the download weights for different devices in a FSM service request. The higher the weight, the sooner we download to that device. By default the weights are set to 0, so that all devices get downloaded at the same time during service deployment.
weightForCE	0	integer	Download weight for CE devices.
weightForPE	0	integer	Download weight assigned to PE devices.
weightForPE_CLE	0	integer	Download weight for PE_CLE devices.
platform/			Contains properties for L2VPN for different platforms.
CATOS/			Service blade parameters for CATOS.
serviceBladeClass	com.cisco.vpnsc.prov.fsm.FSMServiceBlade	string	ServiceBladeClass location.
CISCO_ROUTER/			
IosXRConfigType	XML		Config type for IOS XR devices for MPLS service blade
serviceBladeClass	com.cisco.vpnsc.prov.fsm.FSMServiceBlade	string	ServiceBladeClass location.

Table B-1 DCPL Properties (continued)

dataFileSchema	l2vpnData.xsd	string	Specifies the schema for the data XML file for VPLS.
logLevel	CONFIG	selection	The log Level is the level at which logging is done for this component. These levels are identical to the logging levels defined for JDK1.4 logging package. The levels in descending order are: SEVERE (highest value) WARNING INFO CONFIG FINE FINER FINEST (lowest value).
parseConfigAfterProvisioning	false	The valid values are **true** and **false**.	This property controls the parsing of the configuration file after the provisioning is completed in order to make sure that device inventory is in sync with network.
saveDebugData	true	The valid values are **true** and **false**.	If this property is set to **true**, whenever an SR is provisioned, the uploaded config files and input XML data are saved to a temporary directory for debugging purposes.
sendAuditEvent	true	The valid values are **true** and **false**.	Set true to enable sending audit event for this service.
serviceFile	fsmPwService.xml	string	Specifies the XML file containing the service definition for VPLS. The schema for this file is specified by Provisioning.Engine.serviceSchema.
l2vpn/			MPLS Layer 2 VPN Provisioning.
DownloadWeights/			Specifies the download weights for different devices in an L2VPN service request. The higher the weight, the sooner we download to that device. By default the weights are set to 0, so that all devices get downloaded at the same time during service deployment.
weightForCE	0	integer	Download weight for CE devices.
weightForPE	0	integer	Download weight assigned to PE devices.
weightForPE_CLE	0	integer	download weight for PE_CLE devices.
platform/			Contains properties for L2VPN for different platforms.
CATOS/			Service blade parameters for CATOS.
serviceBladeClass	com.cisco.vpnsc.prov.l2vpn.L2VPNServiceBlade	string	ServiceBladeClass location.
CISCO_ROUTER/			

Table B-1 DCPL Properties (continued)

iosXRConfigType	XML		Config type for IOS XR devices for MPLS service blade
serviceBladeClass	com.cisco.vpnsc.prov.l2vpn.L2VPNServiceBlade	string	ServiceBladeClass location.
dataFileSchema	l2vpnData.xsd	string	Layer 2 VPN Data File schema.
logLevel	CONFIG	selection	The log Level is the level at which logging is done for this component. These levels are identical to the logging levels defined for JDK1.4 logging package. The levels in descending order are: SEVERE (highest value) WARNING INFO CONFIG FINE FINER FINEST (lowest value).
parseConfigAfterProvisioning	false	The valid values are **true** and **false**.	This property controls the parsing of the configuration file after the provisioning is completed in order to make sure that device inventory is in sync with network.
saveDebugData	true	The valid values are **true** and **false**.	If this property is set to **true**, whenever an SR is provisioned, the uploaded config files and input XML data are saved to a temporary directory for debugging purposes.
sendAuditEvent	true	The valid values are **true** and **false**.	Set true to enable sending audit event for this service.
serviceFile	l2vpnService.xml	string	Layer 2 VPN Service definition file.
logLevel/	SEVERE	selection	The log Level is the level at which logging is done for this component. These levels are identical to the logging levels defined for JDK1.4 logging package. The levels in descending order are: SEVERE (highest value) WARNING INFO CONFIG FINE FINER FINEST (lowest value).
mpls/			Contains properties for MPLS/BGP Layer 3 VPN service.
DownloadWeights/			Specifies the download weights for different devices in an MPLS-VPN service request. The higher the weight, the sooner we download to that device. By default the weights are set to 0, so that all devices get downloaded at the same time during service deployment.
weightForCE	0	integer	Download weight for CE devices.

Table B-1 DCPL Properties (continued)

weightForMVRFCE	0	integer	Download weight for MVRFCE. The higher the weight the sooner we download to this device while deploying a service request.
weightForPE	0	integer	Download weight assigned to PE devices.
weightForPE_CLE	0	integer	Download weight for PE_CLE devices.
platform/			Platform related classes.
CATOS/			Service blade parameters for CATOS.
serviceBladeClass	com.cisco.vpnsc.prov.mpls.MplsServiceBlade	string	ServiceBladeClass location.
CISCO_ROUTER/			IOS.
iosXRConfigType	XML		Config type for IOS XR devices for MPLS service blade
serviceBladeClass	com.cisco.vpnsc.prov.mpls.MplsServiceBlade	string	ServiceBladeClass location
allowDuplicateIpAddressForPPPoATM	false	The valid values are **true** and **false**.	Provision PPPoATM by allowing duplicate IP addresses for MPLS Service Requests. Ignore duplicate IP address on Loopback and Multilink interfaces.
allowOverwriteManualAssignedAddress	false	The valid values are **true** and **false**.	Allow manually-assigned IP address in Service Request overwrite the pre-existing interface IP address. **False** means if an MPLS service request tries to provision a manually-assigned IP address to an interface that already has a different IP address on it, Prime Fulfillment detects that and reports the error. **True** means Prime Fulfillment allows the new IP address to overwrite the existing IP address.
allowShared VLAN Modification	false	The valid values are **true** and **false**.	For residential services, if the flag is on, **true**, shared VLAN attributes are available for modify in edit mode. If the flag is off, **false**, attributes are in read only mode.
auditIpAddressViaUnnumbered	false	The valid values are **true** and **false**.	When the value is the default of **false**, the auditor only looks for the IP address of a provisioned interface. When the value is **true**, the auditor tries to match the IP address of the unnumbered interface, if one exists.

Table B-1 DCPL Properties (continued)

auditMaxrouteThreshold	true	The valid values are **true** and **false**.	This property controls whether an audit will be run on the Max Route Threshold for a Service Request. This is needed to maintain backward compatibility.
auditPartialCommands	false	The valid values are **true** and **false**.	This property is set for the autodiscovered systems containing a superset of the commands that Prime Fulfillment supports.
dataFileSchema	l3vpnData.xsd	string	Specifies the schema for the data XML file for MPLS/BGP layer3 VPNs.
excludeNoKeepaliveConfigOnPort Channel	false	The valid values are **true** and **false**.	Exclude the no keepalive command on the port channel trunk port.
forceRemoveNonBroadcastStatic RouteOnPE	false	The valid values are **true** and **false**.	The default value is **false**. When the value is set to **true**, Prime Fulfillment removes the non-broadcast type static route command that has a pre-existing long syntax, even if the command was not provisioned by Prime Fulfillment. The non-broadcast type static route command is removed from a PE router prior to provisioning. Long syntax contains both an outgoing interface name and a next hop IP address.
ignoreLoopbackWhileRemovingVRF	false	The valid values are **true** and **false**.	Remove a VRF, even when some Loopback interfaces are still pointing to it.
ignoreMajorInterfaceCheck	false	The valid values are **true** and **false**.	This property controls the check for a proper major interface name in an unmanaged CE. If set to **true**, Prime Fulfillment bypasses the check for a proper major interface name. Note: This will work only for Unmanaged CE devices
ignoreStatusMessagesForUnmanaged CEs	false	The valid values are **true** and **false**.	If set to **true**, this property prevents the generation of status messages for unmanaged CEs
logLevel	CONFIG	selection	The log Level is the level at which logging is done for this component. These levels are identical to the logging levels defined for JDK1.4 logging package. The levels in descending order are: SEVERE (highest value) WARNING INFO CONFIG FINE FINER FINEST (lowest value).

Table B-1 DCPL Properties (continued)

ospfProcessLimit	2	integer	If the number of OSPF processes reaches or exceedes this limit, a warning will be produced.
parseConfigAfterProvisioning	false	The valid values are **true** and **false**.	This property controls the parsing of the configuration file after the provisioning is completed in order to make sure that device inventory is in sync with network.
passAuditForNonBroadcastStaticRouteOnPE	false	The valid values are **true** and **false**.	When this property is set to **true**, the Prime Fulfillment auditor does not generate an error message if the static route was found with a different format (such as, a PE interface name instead of a CE IP address).
passIpAddressAuditWhenNoAddressDetected	false	The valid values are **true** and **false**.	Pass the IP address command auditing if uploaded router config does not contain an IP address. This is to prevent the audit failure from appended template blob overwriting the provisioned IP address command.
reapplyIpAddress	false	The valid values are **true** and **false**.	Re-apply the same IP address to the interface when decommission a service request. This option is only applicable to manually-assigned IP addresses. It does not work for automatically-assigned IP addresses. When this property is in effect, the interface negate command will not be generated.
removeSubInterface	true	The valid values are **true** and **false**.	Removing the Prime Fulfillment generated subinterface commands in decommission service requests.
routeMapDeletedAfterLastLinkDeletion	true	The valid values are **true** and **false**.	If this property is set to **true**, the route map configuration is automatically removed from the device after the last link is deleted. If **false**, the route map configuration is left as it is in the device.
saveDebugData	true	The valid values are **true** and **false**.	If this property is set to **true**, whenever an SR is provisioned, the uploaded config files and input XML data are saved to a temporary directory for debugging purposes.
sendAuditEvent	true	The valid values are **true** and **false**.	Set true to enable sending audit event for this service.

Table B-1 DCPL Properties (continued)

serviceFile	l3vpnService.xml	string	Specifies the XML file containing the service definition for MPLS/BGP layer3 VPNs. The schema for this file is specified by Provisioning.Engine.serviceSchema
skipIpAddressValidationOn UnmanagedCE	false	The valid values are **true** and **false**.	When the value is **false**, the IP addresses between a PE and an unmanaged CE are validated to ensure they are in the same subnetwork and valid host addresses. When the value is **true**, this validation is bypassed.
useNextHopAddressForStaticRoutes	false	The valid values are **true** and **false**.	For Static Routes, use local router outbound interface or IP address of the next hop to reach the destination network.
useOnlyExtraCEloopbackForGrey AccessList	false	The valid values are **true** and **false**.	With Extra CE loopback, the user can select this option to add only the loopback address instead of the interface ip address and extra CE loopback.
shared/			Properties shared by MPLS VPN, L2VPN and VPLS.
FeatureQuery/			Prime Fulfillment components that check if certain features are available for certain devices based on their software version and platform information.
enableValidation	true	The valid values are **true** and **false**.	If enabled, FeatureQuery will check if the features are available based on the feature matrix and device OS version (IOS Version or PIX Version). If disable it will assume that all features are available on all platforms (should be used for testing only).
IosXrVersionFilesDir		string	Path to IOS XR version XML files.
actionTakenOnUNIVlanList	prune	string	Action taken when switch port **allowed vlan** cmd is absent for ERS service.
leaveSystemMTUUnset	false	The valid values are **true** and **false**.	If this property is set as **true**: U-PE system MTU is not set as default, or set as value given by user; N-PE SVI MTU is set as 9216 for VPLS(EWS and ERS) and L2VPN(EWS). If this property is set as **false**: U-PE system MTU is set as minimum value 1522, or set as value given by user; N-PE SVI MTU is not set as default, or set as value given by user.

Table B-1 DCPL Properties (continued)

overwriteInterfaceDescription	true	The valid values are **true** and **false**.	By default, Prime Fulfillment generates a **description** subcommand for all the physical interfaces it provisioned. Set this property to false if this behavior is not desirable. This property does not apply to logical interfaces or other CLI objects that have a **description** subcommand (Example: crypto map entries, gre Interfaces, and so on).
transferUNIDescToVlanName	false	The valid values are **true** and **false**.	Controls provisioning of the VLAN name on the PE-POP. If set to **true**, the VLAN name is assigned from the description for the UNI. If set to the default of **false**, no VLAN name is assigned.
useSRDescriptionToGenerateDebug Data	false	The valid values are **true** and **false**.	This property is used to generate more intuitive debug data for easy fixing of issues.
staging/			
platform/			Platform related classes.
CATOS/			Service blade parameters for CATOS.
serviceBladeClass	com.cisco.vpnsc.prov.staging.StagingServiceBlade	string	ServiceBladeClass location.
CISCO_ROUTER/			IOS.
serviceBladeClass	com.cisco.vpnsc.prov.staging.StagingServiceBlade	string	ServiceBladeClass location.
logLevel	CONFIG	selection	The log Level is the level at which logging is done for this component. These levels are identical to the logging levels defined for JDK1.4 logging package. The levels in descending order are: SEVERE (highest value) WARNING INFO CONFIG FINE FINER FINEST (lowest value).
parseConfigAfterProvisioning	false	The valid values are **true** and **false**.	This property controls the parsing of the configuration file after the provisioning is completed to make sure that device inventory is in sync with network.
saveDebugData	true	The valid values are **true** and **false**.	If this property is set to **true**, whenever an SR is provisioned, the uploaded config files and input XML data are saved to a temporary directory for debugging purposes.

Table B-1 DCPL Properties (continued)

sendAuditEvent	true	The valid values are **true** and **false**.	Set **true** to enable sending audit event for this service.
serviceFile	stagingService.xml	string	Specifies the XML file containing the service definition for staging service. The schema for this file is specified by Provisioning.Engine.serviceSchema.
vpls/			Contains properties for Virtual Private LAN Service.
DownloadWeights/			Specifies the download weights for different devices in a VPLS service request. The higher the weight, the sooner we download to that device. By default the weights are set to 0, so that all devices get downloaded at the same time during service deployment.
weightForCE	0	integer	Download weight for CE devices.
weightForPE	0	integer	Download weight assigned to PE devices.
weightForPE_CLE	0	integer	Download weight for PE_CLE devices.
dataFileSchema	vplsData.xsd	string	Specifies the schema for the data XML file for VPLS.
logLevel	CONFIG	selection	The log Level is the level at which logging is done for this component. These levels are identical to the logging levels defined for JDK1.4 logging package. The levels in descending order are: SEVERE (highest value) WARNING INFO CONFIG FINE FINER FINEST (lowest value).
parseConfigAfterProvisioning	false	The valid values are **true** and **false**.	This property controls the parsing of the configuration file after the provisioning is completed to make sure that device inventory is in sync with network.
platform/			Platform related classes.
CATOS/			Service blade parameters for CATOS.
serviceBladeClass	com.cisco.vpnsc.prov.vpls.VplsServiceBlade	string	ServiceBladeClass location.
CISCO_ROUTER/			IOS.
serviceBladeClass	com.cisco.vpnsc.prov.vpls.VplsServiceBlade	string	ServiceBladeClass location.

Table B-1 DCPL Properties (continued)

saveDebugData	true	The valid values are **true** and **false**.	If this property is set to **true**, whenever an SR is provisioned, the uploaded config files and input XML data are saved to a temporary directory for debugging purposes.
sendAuditEvent	true	The valid values are **true** and **false**.	Set **true** to enable sending audit event for this service.
serviceFile	vplsService.xml	string	Specifies the XML file containing the service definition for VPLS. The schema for this file is specified by Provisioning.Engine.serviceSchema.
SLA Properties:			Service Level Agreement. This component deals with creating SAA probes between different devices and to collect/aggregate the data corresponding to those probes, in order to provide different SLA reports.
copyRunningToStartup	true	The valid values are **true** and **false**.	If **true** and if showInRunningConfig is **true** - the running configuration will be copied to startup after the router SA Agent configuration has been changed.
daysToKeepDailyStats	365	integer, 30-3650 days	Specifies how many days should the SLA database keep the daily statistics. Specifying a low number keeps the database small but you will not be able to access daily reports beyond this period.
daysToKeepHourlyStats	60	integer, 7-1000 days	Specifies how many days should the SLA database keep the hourly statistics. Specifying a low number keeps the database small but you will not be able to access hourly reports beyond this period.
logLevel	CONFIG	selection	The log Level is the level at which logging is done for this component. These levels are identical to the logging levels defined for JDK1.4 logging package. The levels in descending order are: SEVERE (highest value) WARNING INFO CONFIG FINE FINER FINEST (lowest value).
rowAgeOut	3600	integer, 0-2073600 seconds	The time after which a probe is completely removed after its life is over. In seconds.
showInRunningConfig	true	The valid values are **true** and **false**.	If true, the configured SLAs appear in the router's running configuration.

Table B-1 DCPL Properties (continued)

SYSTEM Properties:			The properties common to all sub-systems in Prime Fulfillment can be found under this component. Most of the values here are set at the time of installation.
/SYSTEM/app_dir	<*vpnsc_home*>	string	Location of the Prime Fulfillment installation.
/SYSTEM/ciscoURL	http://www.cisco.com	string	The Cisco URL.
/SYSTEM/databaseServer	<*db_server*>	string	The database server fully qualified name.
/SYSTEM/email/			Properties related to e-mails sent out by Prime Fulfillment.
from	<*mailfrom*>	string	The **from** field in the e-mail header of the mails sent out by Prime Fulfillment.
smtpHost	<*mailhost*>	string	The server using which e-mail messages from Prime Fulfillment should be sent out.
/SYSTEM/fullyManaged/			Properties related to e-mails sent out by Prime Fulfillment in case of fully managed devices.
auditableCommandsFileLocation		string	This property specifies the full path to the file containing the list of prefixes of auditable commands used in the Fully Managed feature.
enforcementAuditScript		string	Script to be invoked when failure of enforcement audit is detected.
externalEventsEmailRecipients	<*mailto*>	string	The comma or space separated list of email addresses to which notification should be sent out when receiving a config-change event originated outside Prime Fulfillment.
/SYSTEM/license/			Properties related to Prime Fulfillment Licensing.
emailRecipients	<*mailto*>	string	The comma separated list of e-mail addresses to which the License Threshold e-mails should be sent out.
refreshInterval	1	integer, 1-24 hours	License refresh interval in hours.
threshold	90	integer, 1-100%	VPN and ACTIVATION Threshold in percent for e-mail notification.
/SYSTEM/masterServer	<*master_server*>	string	The master server fully qualified name.
/SYSTEM/maxTaskLimit	500	integer	maxTaskLimit.
/SYSTEM/role	master	string	The possible value is: master.
/SYSTEM/tibco/			TIBCO related properties.

Table B-1 DCPL Properties (continued)

port	*<tibco_port>*	integer	The port on which TIBCO Rendezvous listens for events.
prefix	cisco.vpnsc.	string	Prefix for all TIBCO messages originating from Prime Fulfillment.
rva-http-port	*<rva_http_port>*	integer	The http port for TIBCO Rendezvous agent web interface.
rva-port	*<rva_port>*	integer	The port on which TIBCO Rendezvous agent listens for events.
/SYSTEM/tmpdir	*<vpnsc_tmp>*	string	Location for temporary files.
Scheduler Properties:			Scheduler reads the task repository and schedules tasks on every minute boundary. Each scheduled task is passed to Task manager for execution.
/Scheduler/logLevel	CONFIG	selection	The log Level indicates the level at which logging is done for this component. These levels are identical to the logging levels defined for JDK1.4 logging package. The levels in descending order are: SEVERE (highest value) WARNING INFO CONFIG FINE FINER FINEST (lowest value).
/Scheduler/syncInterval	5	integer, 0-10 minutes	When scheduler starts up for the first time, it reads all the scheduling information from the task repository. After that, it depends on the events generated by task repository for receiving changes to the scheduling information. It can also periodically synchronize with the task repository by re-reading it at regular intervals. This property specifies, in minutes, that interval. If the value for the interval is 0, scheduler will not synchronize with the task repository and only depends on the events.
Services Properties:			Common services.
/Services/Common/			
/SharedUNI			
logLevel	CONFIG	selection	The log Level indicates the level at which logging is done for this component. These levels are identical to the logging levels defined for JDK1.4 logging package. The levels in descending order are: SEVERE (highest value) WARNING INFO CONFIG FINE FINER FINEST (lowest value).

Table B-1 DCPL Properties (continued)

taskScheduleDelay	5	integer, -1-120	Specify the number of minutes to wait after changing shared UNI attributes and before starting to execute a synchronization task. Note: A scheduled task is created to execute after a specified delay. range (-1 to 120) minutes -1 (do not schedule automatically) 0 (schedule immediately, ASAP) 120 (highest value)
allowForcePurge	true	The valid values are **true** and **false**.	With the default value of **true**, you can force purge a Service Request. If the value is **false**, you cannot force purge a Service Request.
disableFallBack	false	The valid values are **true** and **false**.	This property is used to set a default value for the DisableFallBack property of PseudowireClass. This property is effective only for IOS XR L2VPN services.
disallowVlan1	true	The valid values are **true** and **false**.	This prevents allocating VLAN ID 1 for services configured by Prime Fulfillment. This is applicable for both auto allocation of VLAN from VLAN resource pool and manual allocation. Set this property to **true** to block Prime Fulfillment from deploying services with VLAN ID 1
interfaceDescRegEx		string	Interface Configuration Regular Expression.
interfaceNameRegEx		string	Interface Name Regular Expression.
l2vpnGroupNameOptions	Prime Fulfillment,VPNSC	string	This property is used to set a comma separated list of a maximum of 10 L2VPN Group Names. This property is effective only for IOS XR L2VPN services.
pseudoWireVlanMode	false	The valid values are **true** and **false**.	This property is effective only for IOS XR L2VPN services. The default is **false**. When set to **true**, this configures pseudowire transport mode to VLANs.
SnmpService Properties:			The Snmp Service package provides APIs to perform SNMP get() and set() operations.
/SnmpService/misc			Advanced settings.

Table B-1 DCPL Properties (continued)

enableDebug	false	The valid values are **true** and **false**.	Enables the AdventNet SNMP stack debug messages. Messages are written to the TaskLogs directory in files stdout and stderr. Warning: These log files grow quickly and are not managed by the Prime Fulfillment logger. Requires WatchDog restart.
rcvPktBuffSize	96	integer, 64-512	Buffer size in K bytes, for SNMP stack receive buffer.
/SnmpService/defaultSNMPVersion	1	integer, 1-2	The default SNMP version used to connect to Cisco router. Used if the SNMP version is not specified per router. Valid Values: SNMPv1/SNMPv2c - 1 SNMPv3 - 2.
/SnmpService/defaultSecurityLevel	3	integer, 1-3	The default security level used to connect to Cisco router. Used if the security level is not specified per router. Values: authentication no encryption - 1 authentication encryption - 2 no authentication no encryption - 3.
/SnmpService/logLevel	CONFIG	selection	The log Level is the level at which logging is done for this component. These levels are identical to the logging levels defined for JDK1.4 logging package. The levels in descending order are: SEVERE (highest value) WARNING INFO CONFIG FINE FINER FINEST (lowest value).
/SnmpService/maxTaskDuration	5	integer, 1-30	Maximum duration in minutes for collecting device interface information. A longer duration is required for devices with large numbers of interfaces. This period must be longer than 2^(retries+1) * timeout.
/SnmpService/retries	3	integer, 0-10	The number of retries to be used by the SNMP protocol.
/SnmpService/timeout	5	integer, 0-300 seconds	Timeout value to be used by the SNMP protocol. Unit: seconds
TE Properties:			Traffic Engineering Management (TEM) Properties
/TE/Deployment			Control the operation of TEM Provisioning
maxCacheSize	60	integer, >0	Maximum cache size.
oneDeviceEachTimeThreshold	500	integer, >0	When the total number of tunnels to be provisioned exceeds this threshold number, provision one device at a time.

Table B-1 DCPL Properties (continued)

partialConfigAudit	false	The valid values are **true** and **false**.	When the value is the default of **false**, the config audit is not limited. When the value is set to **true**, only a partial config audit (audit of only the PENDING tunnels) occurs for primary and backup tunnel deployment.
/TE/repository			TEM Repository-related Properties
checkPermissionEnabled	false	The valid values are **true** and **false**.	This property enables or disables Role-Based Access Control (RBAC) checking during particular TEM operations, such as topology population, discovery, and service deployment. When the value is the default of **false**, RBAC permission checking is not enabled. When the values is set to **true**, RBAC permission checking is enabled and performance degrades.
TE Topology Properties:			TEM Topology-related Properties
/TE Topology/TrafficData			Color Control for Traffic Data Displays
Green	0-25	integer, 0-100 (percentage)	Topology representations for a link performance utilization range, specified as a percentage (default: 0-25), are displayed in the color green.
Orange	51-75	integer, 0-100 (percentage)	Topology representations for a link performance utilization range, specified as a percentage (default: 51-75), are displayed in the color orange.
Red	76-100	integer, 0-100 (percentage)	Topology representations for a link performance utilization range, specified as a percentage (default: 76-100), are displayed in the color red. Greater than 100% is also displayed in red.
Yellow	26-50	integer, 0-100 (percentage)	Topology representations for a link performance utilization range, specified as a percentage (default: 26-50), are displayed in the color yellow.
TaskManager Properties:			Task manager executes tasks that are scheduled by scheduler. Task execution consists of executing different actions that comprise the task. Task manager manages the dependencies between these actions.
/TaskManager/CollectConfig			The Collect Config task uploads the running configuration.

Table B-1 DCPL Properties (continued)

logLevel	CONFIG	selection	The log Level indicates the level at which logging is done for this component. These levels are identical to the logging levels defined for JDK1.4 logging package. The levels in descending order are: SEVERE (highest value) WARNING INFO CONFIG FINE FINER FINEST (lowest value).
/TaskManager/logLevel	CONFIG	selection	The log Level indicates the level at which logging is done for this component. These levels are identical to the logging levels defined for JDK1.4 logging package. The levels in descending order are: SEVERE (highest value) WARNING INFO CONFIG FINE FINER FINEST (lowest value).
TemplateManager Properties:			
userTemplateAttrFile	<*vpnsc_home*>/UserTemplateAttr.xml	string	User template attribute file path and name.
VpnInvServer Properties:			Corba Server for VpnInvServer IDL backward compatibility.
/VpnInvServer/logLevel	SEVERE	selection	The log Level is the level at which logging is done for this component. These levels are identical to the logging levels defined for JDK1.4 logging package. The levels in descending order are: SEVERE (highest value) WARNING INFO CONFIG FINE FINER FINEST (lowest value).
aagent Properties:			AAgent component related defines.
/aagent/defaultVersion	3.6.3	string	The default 3k firmware version for AAgent.
/aagent/directories/			Various directories for aagent.
dmd	<*vpnsc_home*>/resources/AAgent/DMDFiles	string	File path and name.
input	<*vpnsc_home*>/resources/java/classes/common/AAgent/com/cisco/vpnscagent	string	File path and name.
working	<*vpnsc_home*>/resources/java/archives	string	File path and name.

Table B-1 DCPL Properties (continued)

cfr Properties:			The Command Flow Runner component. This currently runs within the Tomcat server (in the Prime Fulfillment web application) and is responsible for running MPLSOAM troubleshooting workflows.
/cfr/Diagnostics/			
disableTunnelDiagnostics	false	The valid values are **true** and **false**.	Set to **true** to disable tunnel diagnostics, in order to avoid errors when running Prime Diagnostics across networks with non-Cisco devices in the tunnel LSPs.
/cfr/LogHandler	com.cisco.mgmt.workflow.util.IscLogHandler		Set the CFR to use a custom handler for logging. The handler should log to a separate file and format the log messages using the java.util.logging.SimpleFormatter instead of the Prime Fulfillment default XML formatting.
/cfr/logLevel	INFO		The level of logging information the Command Flow Runner will log (it will log from the set level upwards). The logging levels are as defined in the java.util.logging package.
lockmanager Properties:			Component that handles device locking. When different jobs (such as provisioning) try to update the config on the device, they obtain **software** locks so that two different jobs do not update the config at the same time. LockManager provides a way to obtain and later release such software locks.
/lockmanager/collectConfigLock	false	The valid values are **true** and **false**.	Determines if a software lock is to be applied to the devices in the CollectConfig task. If **true**, a software lock is applied to all devices prior to executing the CollectConfig operation, and is released upon completion of the CollectConfig operation. Note that a software lock is not applied to the optional device attributes and interfaces operations. This flag is read by the CollectConfig task upon execution.
/lockmanager/lockTimeoutInHours	8	integer, 1-168 hours	Timeout in hours for a lock held by a lock holder. If the lock holder does not free a lock within this time the lockmanager will automatically release the device lock.

Table B-1 DCPL Properties (continued)

/lockmanager/logLevel	SEVERE	selection	The log Level is the level at which logging is done for this component. These levels are identical to the logging levels defined for JDK1.4 logging package. The levels in descending order are: SEVERE (highest value) WARNING INFO CONFIG FINE FINER FINEST (lowest value).
/lockmanager/queueServicingInterval	100	integer, 10-2000 milliseconds	How often in milliseconds to service pending lock requests. A lower value decreases the average time it takes to get a lock at the expense of CPU processing overhead.
nbi Properties:			Northbound API (Nbi) component related defines.
/nbi/BackwardCompatible			Path for execQuery requests.
RecordNumber	false	The valid values are **true** and **false**.	For execQuery requests, the number embedded in the output class name include **Record** for the default, **false**, or **Record#1** for **true**.
/nbi/CompositeDir	<*vpnsc_home*>/ resources/java/ xml/com/cisco/ vpnsc/repository/ meta/xml/ composite	string	Path to composite XML files. Do not change it or the composite meta XML files will not be backed up.
/nbi/CustomerReportMetaDir	<*vpnsc_home*>/ resources/java/ xml/com/cisco/ vpnsc/repository/ meta/xml	string	Path to user defined report meta XML files. Do not change it or the report meta XML files will not be backed up.
/nbi/Formatter	com.cisco.vpnsc. nbi.io.NbiSimple Formatter	string	File path and name.
/nbi/Logger	com.cisco.vpnsc. nbi.util.NbiVpns cLogger	string	File path and name.
/nbi/MetaCheckInterval	300000	string	Set the time for next meta check to happen.
/nbi/MetaDir	<*vpnsc_home*>/ resources/java/ xml/com/cisco/ vpnsc/repository/ meta/xml	string	Path to meta XML files. Do not change it or the meta XML will not be backed up.

Table B-1 DCPL Properties (continued)

/nbi/ProvidedReportMetaDir	<*vpnsc_home*>/resources/java/xml/com/cisco/vpnsc/repository/meta/xml	string	Path to Prime Fulfillment provided report meta XML files. Do not change it or the report meta xml files will not be backed up.
/nbi/Reader	com.cisco.vpnsc.nbi.io.NbiSoapReader	string	File path and name.
/nbi/RequestParserMgr	com.cisco.vpnsc.nbi.parser.NbiRequestParserMgr	string	File path and name.
/nbi/SSLfilepath	<*vpnsc_home*>/bin/client.keystore	string	Path to client.keystore file for NBI SSL connections.
/nbi/SessionTimeout	1200000	string	Amount of time the session is valid. A session is the socket connection between the client and the NBI server through the Tomcat server.
/nbi/TransactionParser	com.cisco.vpnsc.nbi.parser.NbiWsdlParser	string	File path and name.
/nbi/Validation	true	The valid values are **true** and **false**.	Variable to enable validation of incoming Nbi API XML attributes.
/nbi/WaitTimeout	1200	integer	The time in seconds to wait for a Service Request to deploy.
/nbi/Writer/			
SoapEncapsulation	false	The valid values are **true** and **false**.	SoapEncapsulation.
/nbi/Writer	com.cisco.vpnsc.nbi.io.NbiSoapWriter	string	File path and name.
/nbi/logHandler	com.cisco.vpnsc.nbi.util.VpnscLogHandler	string	Custom log handler for nbi. This handler allows NBI to use alternate formatter from default one used by rest of Prime Fulfillment. In this case, NBI defaults to using SimpleFormatter which dumps simple output as opposed to XML output.

Table B-1 DCPL Properties (continued)

/nbi/logLevel	WARNING	selection	The log Level is the level at which logging is done for this component. These levels are identical to the logging levels defined for JDK1.4 logging pack age. The levels in descending order are: SEVERE (highest value) WARNING INFO CONFIG FINE FINER FINEST (lowest value).
notification Properties:			Event notification related defines.
/notification/Logger	com.cisco.vpnsc.nbi.util.NbiVpnscLogger	string	File path and name.
/notification/clientEnabled	false	The valid values are **true** and **false**.	Set to true for enabling the example event receiving servlet.
/notification/clientHost	<*master_server*>	string	TIBCO event client host.
/notification/clientMethod	/notification/servlet eventListener	string	TIBCO event client method.
/notification/clientPort	<*http_port*>	string	TIBCO event client port.
/notification/clientRegFile	<*vpnsc_home*>/resources/nbi/notification clientReg.txt	string	Client TIBCO event registration file name.
/notification/logFormatter	java.util.logging.SimpleFormatter	string	File path and name.
/notification/logHandler	com.cisco.vpnsc.nbi.util.VpnscLogHandler	string	Custom log handler.
/notification/logLevel	WARNING	selection	The log Level is the level at which logging is done for this component. These levels are identical to the logging levels defined for JDK1.4 logging package. The levels in descending order are: SEVERE (highest value) WARNING INFO CONFIG FINE FINER FINEST (lowest value).
/notification/password	cisco	string	Both username and password are same as the ones used for GUI login.
/notification/remotePassword		string	User password for remote system authentication, if required, for example, when LDAP is in use.
/notification/remoteUsername		string	Username for remote system authentication, if required, for example, when LDAP is in use.

Table B-1 DCPL Properties (continued)

/notification/username	admin	string	Both username and password are s same as the ones used for GUI login.
pal Properties:			The PAL Device interaction component. This runs within the Tomcat server and is responsible for running device interaction for the CFR to run the OAM troubleshooting workflows.
/pal/failureScenario			The system parameter that represents the current failure scenario. For use with the Canned Response mechanism for testing.
/pal/logHandler	com.cisco.mgmt.workflow.util.IscLogHandler		Set the PAL to use a custom handler for logging. The handler should log to a separate file and will format the log messages using the java.util.logging.SimpleFormatter instead of the Prime Fulfillment default XML formatting.
/pal/logLevel	INFO		The log Level is the level at which logging is done for this component. These levels are identical to the logging levels defined for JDK1.4 logging package. The levels in descending order are: SEVERE (highest value) WARNING INFO CONFIG FINE FINER FINEST (lowest value).
/pal/responseDir	/vob/ntg/dev/resources/pal/testnetwork		The base directory where the failure scenarios are held. Used by the canned response mechanism and transport for failure scenario testing.
repository Properties:			The component for Database related properties.
/repository/Concurrency/			To setup properties for re-try loop to avoid deadlock
NOICE_FACTOR	500	integer	Add random noise to each process that is being retried.
NO_OF_RETRIES	3	integer	Number of retries before throwing deadlock exception.
TIME_BASE	2	integer	The base number to calculate the wait time. For example, a value of 2 for this property and 3 retries means, the process will be retried every 2^0, 2^1, and 2^2 seconds.
/repository/IPAddressPool/			IP Address Pool Constants.
AGE_TIME	1440	integer	The Aging interval for released IP Address, in minutes. The default is 24 hours (1440 minutes).

Table B-1 DCPL Properties (continued)

RecoverIPAddrSleepInterval	60	integer, 10 - 144000 minutes	The time in minutes for recovering Aged IP addresses recovery service to wait between recovery cycles. The default is 60 minutes. Changing this value initiates the recovery process.
releaseAndReuseAgedAddresses	true	The valid values are **true** and **false**.	The default value is **false**. When the value is set to **true**, the user wants a manual allocation of the address in the aged address to succeed. When the value is set to **true**, the address is released from the Aged Pool and moved to the Allocated pool when manually allocated.
/repository/common			Repository common constants.
MCAST_SUBSUME_ALL_SRS	true	The valid values are **true** and **false**.	This property set at **true** indicates that the user wants all the MPLS VPN links of a VPN to be subsumed when Multicast is enabled for that VPN.
releaseAndReuseAgedAddresses	true	The valid values are **true** and **false**.	The default value is **false**. When the value is set to **true**, the address will be released from the Aged Pool and moved to the Allocated pool when manually allocated.
/repository/deviceConfig/		null	Configuration file related properties.
maxVersions	10	integer, 1-50	Maximum number of configuration files to be stored per device in the repository before older versions automatically get purged.
/repository/mlshare/			Share directory for both MPLS and L2VPN.
allowLoopbackIntfInNPC	false	The valid values are **true** and **false**.	Allows the selection of loopback interfaces in NPC.
logLevel	SEVERE	selection	The log Level is the level at which logging is done for this component. These levels are identical to the logging levels defined for JDK1.4 logging package. The levels in descending order are: SEVERE (highest value) WARNING INFO CONFIG FINE FINER FINEST (lowest value).
/repository/persistence/			Properties for database.
Versions	5	integer	The number of maximum versions for a Versioning Persistent Objects.
catalog	directory	string	Catalog.
driver	<*db_driver*>	string	The class name for the driver.
initialConnections	1	integer, 1-20	Number of initial connections.

Table B-1 DCPL Properties (continued)

location	<*repository_home*>	string	The directory containing the repository.db and repository.log files.
password	sql	string	Password for opening a DB connection.
schema	DBA	string	Schema.
slaurl	jdbc:sybase:Tds:tl-dev-v240-16.cisco.com:2638/?JCONNECT_VERSION=6&serviceName=<server_name>	string	The url for opening a JDBC connection to the SLA database.
url	<*db_url*>	string	The url for opening a JDBC connection.
username	dba	string	User id to open a db connection.
/repository/rbac/			The component for RBAC User Access Model, user Authentication.
cache/isEternal	false	The valid values are **true** and **false**.	Specifies whether the elements in the RBAC cache are eternal, never expire. The value **true** indicates the elements in the cache are eternal and never expire. The default value **false** indicates the elements in the cache can expire.
cache/maxElementsInMemory	5000	integer, 1000 to 10000	Specifies the maximum number of elements in cache memory. Default: 5000.
overflowToDisk	false	The valid values are **true** and **false**.	Specifies whether to use disk to store cache.
cache/timeToIdleSeconds	120	integer, 60 to 1800 seconds	Specifies the default number of seconds for an element to live in cache from its last accessed or modified date. Default: 120 seconds.
cache/timeToLiveSeconds	300	integer, 100 to 3600 seconds	Specifies the default number of seconds for an element to live in cache from its creation date. Default: 300 seconds.
/repository/rbac/checkCreatorPermissionEnabled	true	The valid values are **true** and **false**.	The creator of objects can give the permissions of Modify or Delete to others. If this flag is false, enable RBAC permission checkin.
/repository/rbac/checkPermissionEnabled	true	The valid values are **true** and **false**.	The creator of objects can give the permissions of Modify or Delete to others. If this flag is false, enable RBAC permission checkin.

Table B-1 DCPL Properties (continued)

/repository/rbac/enableAutologin	true	The valid values are **true** and **false**.	The property controls whether user may store login information in form of cookies on the computer from which the user connects. If enabled, automatic login, based on the cookie information is permitted. Also user is presented with a screen in which he or she can elect to store login information on the local user's computer. With this property set to false no autologin or options associated with it are available.
/repository/rbac/logLevel	CONFIG	selection	The log Level is the level at which logging is done for this component. These levels are identical to the logging levels defined for JDK1.4 logging package. The levels in descending order are: SEVERE (highest value) WARNING INFO CONFIG FINE FINER FINEST (lowest value).
/repository/rbac/partialQueryResult Expected	true	The valid values are **true** and **false**.	When checking Permission on a list of Persistent Objects, and the current user does not the specified permission to all the objects in the result list, partial results will be returned if this flag is true; Insufficient Permission exception will be generated if the flag is **false**.
/repository/rbac/webSessionTimeoutSec	1800	integer, 1 - 2,147,483,647	Timeout of inactive web client session in seconds. Default is 30 minutes.
/repository/ual/			User Access/Audit Log
cleanUALogs	true	The valid values are **true** and **false**.	Indicates whether to let the system automatically clean up UAL log entries based on ual.maxAgeInDays.
maxAgeInDays	30	integer	Maximum age of the User Access/Audit Logs in days after which the UALog Cleanup Service will delete them. if 0 then UALogs deletion is disabled even if cleanUALogs is set to true.
watchdog Properties:			All the servers in Prime Fulfillment are launched and managed by the Watchdog.
/watchdog/byRole			
/watchdog/byRole/cs			
servers	httpd nspoller worker dbpoller		Names of servers to be run.
/watchdog/byRole/db			
servers	dbpoller		Names of servers to be run on an installation with the role "db"

Table B-1 DCPL Properties (continued)

/watchdog/byRole/is			
servers	httpd dbpoller		Names of servers to be run on an installation with the role "is"
/watchdog/byRole/master			
servers	httpd nspoller dbpoller dispatcher worker scheduler lockmanager cnsserver discovery rgserver		Names of servers to be run.
/watchdog/byRole/ps			
servers	httpd nspoller worker dbpoller		Names of servers to be run.
/watchdog/criticalServers		string	If any of these servers enters the disabled state, then it would mean that the system is NOT healthy. If this value is null/empty then every single server is critical.
/watchdog/diskspace/			Contains properties related to disk space monitoring.
dirsToMonitor		string	The directories (and ultimately the disks that contain them) to be monitored.
disksToMonitor		string	The disks to be monitored for space constraints.
emailRecipients	<*mailto*>	string	The comma separated list of e-mail addresses to which the disk space related e-mails should be sent out.
highWatermark	<*highwater*>	string	High watermark for the directories (disks) being monitored. The value should be a number followed by a < (for percent) or m or M (for Mbytes). These values should correspond to the available/free space on the disk. If the available disk space stabilizes above this value (after falling below the low watermark), an e-mail is sent to the addresses specified in the property watchdog.diskspace.emailRecipients.

Table B-1 DCPL Properties (continued)

lowWatermark	*<lowwater>*	string	Low watermark for the directories (disks) being monitored. The value should be be a number followed by a % (for percent) or m or M (for Mbytes). These values should correspond to the available/free space on the disk. If the available disk space falls below this value, an e-mail is sent to the addresses specified in the property watchdog.diskspace.emailRecipients.
sleepInterval	60000	integer, 30000-300000 milliseconds	Time between two status checks for disk space limits in milliseconds.
/watchdog/group/			Group.
database_users	scheduler httpd	string	The servers that access database.
/watchdog/groups	database_users	string	The space separated list of different groups in the system.
/watchdog/heartbeat/			Properties related to watchdog heartbeat mechanism are specified here.
period	120000	integer, 30000-86400000 milliseconds	The minimum time between each heartbeat request in milliseconds.
period_poller	60000	integer, 30000-86400000 milliseconds	The minimum time between each heartbeat request for dbpoller and nspoller in milliseconds.
sendEvents	false	The valid values are **true** and **false**.	If set to true, watchdog sends out TIBCO events every time a heartbeat succeeds or fails. If set to false, no such events will be sent.
startDelay	5000	integer, 0-60000 milliseconds	Time to wait before making the first heartbeat request in milliseconds.
timeout	3000	integer, 1000-600000 milliseconds	The period of time before which response for heartbeat request should be received by the watchdog, in milliseconds.
wds/			Heartbeat properties for intra-watchdog communication.
delay	5000	integer, 1000-60000 milliseconds	The period in between heartbeats. (from master watchdog to slave watchdog and vice-versa) in milliseconds.
initDelay	1000	integer, 1000-5000 milliseconds	The initial period of time for which the heartbeat thread waits before trying for a heartbeat after a watchdog registers with the MasterWatchdog, in milliseconds.

Table B-1 DCPL Properties (continued)

masterReconnectAttemptDelay	2000	integer, 100-60000 milliseconds	The sleep time between two successive attempts by a slave watchdog to reconnect to master watchdog, in milliseconds.
maxAllowedMisses	3	integer	The maximum number of consecutive misses that a watchdog should miss for the master to consider it inactive or unregistered.
maxAttemptsForMasterReconnect	500	integer	After the slave watchdog loses connection with the master, it will try this many times to try and establish the connection. If it cannot re-establish a connection with the master even after making these many attempts, it shuts itself down. Between attempts, it sleeps **watchdog.heartbeat.wds.masterReconnectAttemptDelay** time. The value for this property should be specified in milliseconds. A value of 0 indicates that the slave watchdog has no upper limit on the number of reconnect attempts.
/watchdog/java/			Java.
flags	-XX:+UseAltSigs	string	Any other flags to be passed on to **java.**
vmtype	-server	string	The flag to be passed on to java (-server or -client).
/watchdog/logLevel	FINEST	selection	The log Level is the level at which logging is done for this component. These levels are identical to the logging levels defined for JDK1.4 logging package. The levels in descending order are: SEVERE (highest value) WARNING INFO CONFIG FINE FINER FINEST (lowest value).
/watchdog/server/	httpd nspoller dbpoller dispatcher worker scheduler lockmanager cornerstonebridge	string	Server.
cnsserver/			Monitors CNS events from IE2100 boxes. Communication between client and server is completely handled using TIBCO events.
heartbeat/			Heartbeat related properties.

Table B-1 DCPL Properties (continued)

startDelay	10000	integer, 0-60000 milliseconds	Time to wait before making the first heartbeat request in milliseconds.
timeout	3000	integer, 1000-600000 milliseconds	The period of time before which response for heartbeat request should be received by the watchdog, in milliseconds.
java/			Java attributes for this server.
flags		string	Any additional java flags specific to this server. If the value is changed, watchdog restart is required for the new value to take effect.
class	com.cisco.vpnsc.watchdog.servers.WDCnsServer	string	Heartbeat Handler - Checks for valid TIBCO Connection.
cmd	java com.cisco.vpnsc.cns.CnsServer	string	Implementation to monitor CNS events from IE2100 boxes.
dependencies	dbpoller	string	Dependencies.
logLevel	CONFIG	selection	The log Level is the level at which logging is done for this component. These levels are identical to the logging levels defined for JDK1.4 logging package. The levels in descending order are: SEVERE (highest value) WARNING INFO CONFIG FINE FINER FINEST (lowest value).
dbpoller/			This server keeps polling the database to see if it is functional.
class	com.cisco.vpnsc.watchdog.servers.WDDatabase	string	Name of class responsible for getting heartbeats.
connectionextend	5	integer, 1-15	For Oracle RAC failover, increase this value to make sure the failover happens before dbpoller stops.
logLevel	CONFIG	selection	The log Level is the level at which logging is done for this component. These levels are identical to the logging levels defined for JDK1.4 logging package. The levels in descending order are: SEVERE (highest value) WARNING INFO CONFIG FINE FINER FINEST (lowest value).
select	select id from vpnsc_host	string	SQL select statement to issue when pinging the database.
discovery/			Handles various Prime Fulfillment Discovery workflow related tasks.

Table B-1 DCPL Properties (continued)

class	com.cisco.vpnsc.discovery.engine.server.DiscoveryServer	string	Heartbeat Handler.
cmd	java com.cisco.vpnsc.discovery.engine.server.DiscoveryImpl	string	Implementation of the Discovery work interface.
dependencies	dbpoller	string	dependencies
heartbeat/			Heartbeat related properties.
startDelay	10000	integer, 0-60000 milliseconds	Time to wait before making the first heartbeat request in milliseconds.
timeout	3000	integer, 1000-60000 milliseconds	The period of time before which response for heartbeat request should be received by the watchdog, in milliseconds. To discover large networks with a complex topology, we recommend you reset this property to 180000 milliseconds (3 minutes).
java/			Java attributes for this server
flags		string	Any additional java flags specific to this server. If the value is changed, watchdog restart is required for the new value to take effect. To discover large networks with a complex topology, we recommend you reset this property to -Xmx3072m -XX:PermSize=256m -XX:MaxPermSize=512m.
logLevel	CONFIG	selection	The log Level is the level at which logging is done for this component. These levels are identical to the logging levels defined for JDK1.4 logging package. The levels in descending order are: SEVERE (highest value) WARNING INFO CONFIG FINE FINER FINEST (lowest value).
dispatcher/			Dispatcher service of the Distribution framework.
app_args	Dispatcher com.cisco.vpnsc.dist.vpnsc.VpnscDispatcherImpl	string	Args to the class that starts this service.
class	com.cisco.vpnsc.watchdog.servers.WDDispatcher	string	The class that proxies this service for the watchdog.

Table B-1 DCPL Properties (continued)

cmd	java com.cisco.vpnsc.watchdog.ext.ServiceLauncherImpl	string	Command to start the server.
dependencies	dbpoller nspoller	string	The other services that this service depends on Heartbeat related properties.
heartbeat/			
startDelay	45000	integer, 0-60000 milliseconds	Time to wait before making the first heartbeat request in milliseconds.
timeout	3000	integer, 1000-60000 milliseconds	The period of time before which response for heartbeat request should be received by the watchdog, in milliseconds.
java/			Java attributes for this server
flags		string	Any additional java flags specific to this server. If the value is changed, watchdog restart is required for the new value to take effect.
logLevel	CONFIG	selection	The log Level is the level at which logging is done for this component. These levels are identical to the logging levels defined for JDK1.4 logging package. The levels in descending order are: SEVERE (highest value) WARNING INFO CONFIG FINE FINER FINEST (lowest value).
httpd/		httpd	httpd
class	com.cisco.vpnsc.watchdog.servers.WDHttpd	string	Class.
cmd	<*vpnsc_home*>/bin/tomcat.sh start fg	string	The command to start httpd on this host.
dependencies	dbpoller	string	Dependencies.
dependenciesByRole/			
cs		string	Dependencies on a cs.
ps	dbpoller	string	Dependencies on a ps.
heartbeat/			Heartbeat.
port	<*http_port*>	integer	The port on which httpd should run.
startDelay	45000	integer, 0-60000 milliseconds	Time to wait before making the first heartbeat request in milliseconds.

Table B-1 DCPL Properties (continued)

timeout	10000	integer, 1000-600000 milliseconds	The period of time before which response for heartbeat request should be received by the watchdog, in milliseconds.
url	http://localhost:<*http_port*>/isc/about.htm	string	url
logLevel	CONFIG	selection	The log Level is the level at which logging is done for this component. These levels are identical to the logging levels defined for JDK1.4 logging package. The levels in descending order are: SEVERE (highest value) WARNING INFO CONFIG FINE FINER FINEST (lowest value).
lockmanager/			Component that handles locking.
class	com.cisco.vpnsc.watchdog.servers.WDLockManager	string	Class that keeps track of lockmanager heartbeats.
cmd	java com.cisco.vpnsc.lockmanager.LockManagerImpl	string	Command that starts up the lockmanager.
dependencies	nspoller	string	Lock Manager depends on the NS.
heartbeat/			Heartbeat related properties.
startDelay	10000	integer, 0-60000 milliseconds	Time to wait before making the first heartbeat request in milliseconds.
timeout	3000	integer, 1000-600000 seconds	The period of time before which response for heartbeat request should be received by the watchdog, in milliseconds.
java/			Java attributes for this server.
flags		string	Any additional java flags specific to this server. If the value is changed, watchdog restart is required for the new value to take effect.
logLevel	CONFIG	selection	The log Level is the level at which logging is done for this component. These levels are identical to the logging levels defined for JDK1.4 logging package. The levels in descending order are: SEVERE (highest value) WARNING INFO CONFIG FINE FINER FINEST (lowest value).

Table B-1 DCPL Properties (continued)

maxQuickDieCount	3	integer	The maximum number of times a server can die consecutively without having a successful heartbeat. If this number is exceeded, the server is marked as disabled.
nspoller/			This server polls the NameServer to see if it is running.
class	com.cisco.vpnsc.watchdog.servers.WDNameServer	string	Class.
logLevel	CONFIG	selection	The log Level is the level at which logging is done for this component. These levels are identical to the logging levels defined for JDK1.4 logging package. The levels in descending order are: SEVERE (highest value) WARNING INFO CONFIG FINE FINER FINEST (lowest value).
rgserver/			TEM server for the TEM tunnel generation algorithm.
heartbeat/			
rgport		string	The port on which rgserver should run.
startDelay	45000	integer, 0-60000 milliseconds	Time to wait before making the first heartbeat request in milliseconds.
timeout	3000	integer, 1000-600000 milliseconds	The period of time before which response for heartbeat request should be received by the watchdog, in milliseconds.
class	com.cisco.vpnsc.watchdog.servers.WDRGServer	string	Class.
cmd	rgserver.sh	string	Command to start the rgserver.
dependencies	httpd	string	Servers that must be functioning for this server to function normally.
logLevel	CONFIG	selection	The log Level is the level at which logging is done for this component. These levels are identical to the logging levels defined for JDK1.4 logging package. The levels in descending order are: SEVERE (highest value) WARNING INFO CONFIG FINE FINER FINEST (lowest value).
scheduler/			Scheduler.
class	com.cisco.vpnsc.watchdog.servers.WDScheduler	string	Class.

Table B-1 DCPL Properties (continued)

cmd	java com.cisco.vpnsc.scheduler.Scheduler	string	Command to start the scheduler.
dependencies	dbpoller worker	string	Dependencies.
heartbeat/			Heartbeat related properties.
startDelay	30000	integer, 0-60000 milliseconds	Time to wait before making the first heartbeat request in milliseconds.
timeout	3000	integer, 1000-600000 milliseconds	The period of time before which response for heartbeat request should be received by the watchdog, in milliseconds.
java/			Java attributes for this server.
flags		string	Any additional java flags specific to this server. If the value is changed, watchdog restart is required for the new value to take effect.
logLevel	CONFIG	selection	The log Level is the level at which logging is done for this component. These levels are identical to the logging levels defined for JDK1.4 logging package. The levels in descending order are: SEVERE (highest value) WARNING INFO CONFIG FINE FINER FINEST (lowest value).
startTimeout	240000	integer, 5000-600000	The timeout for the initial heartbeat response. The first heartbeat should happen within this time.
worker/			Worker service of the distribution framework.

Table B-1 DCPL Properties (continued)

app_args	Worker com.cisco.vpnsc.dist.WorkerImpl, com.cisco.vpnsc.sla.sql.SlaMaintenanceService, com.cisco.vpnsc.repository.ual.UALCleanupServiceImpl, com.cisco.vpnsc.license.LicenseSynchronize, com.cisco.vpnsc.cleanup.TaskLogCleanupService, com.cisco.vpnsc.cleanup.TempFileCleanupService, com.cisco.vpnsc.cleanup.RuntimeTaskCleanupService”	string	Arguments to the class specified in the **cmd** property.
class	com.cisco.vpnsc.watchdog.servers.WDWorker	string	The server class that proxies Worker service for the watchdog.
cmd	java com.cisco.vpnsc.watchdog.ext.ServiceLauncherImpl	string	Command to start the worker.
dependencies	nspoller	string	Servers that have to be functioning for this server to function normally.
dependenciesByRole/			
cs		string	Dependencies on a cs.
ps	dbpoller	string	Dependencies on a ps.
heartbeat/			Heartbeat related properties.
startDelay	45000	integer, 0-60000 milliseconds	Time to wait before making the first heartbeat request in milliseconds.
timeout	3000	integer, 1000-600000 milliseconds	The period of time before which response for heartbeat request should be received by the watchdog, in milliseconds.
java/			Java attributes for this server.

Table B-1 DCPL Properties (continued)

flags	-Xmx512m -Xbootclasspath/p:<*vpnsc_home*>/thirdparty/jar/AdventNetSnmp3_3.2.jar:<*vpnsc_home*>/thirdparty/jar/cryptix32.jar -Dcom.cisco.insmbu.templatemgr.backend.PropFile=<*vpnsc_home*>/resources/templatesystem/Template.properties	string	Any additional java flags specific to this server. If the value is changed, watchdog restart is required for the new value to take effect.
logLevel	CONFIG	selection	The log Level is the level at which logging is done for this component. These levels are identical to the logging levels defined for JDK1.4 logging package. The levels in descending order are: SEVERE (highest value) WARNING INFO CONFIG FINE FINER FINEST (lowest value).
/watchdog/serverStatus/			The properties related to the server status monitoring function provided by the watchdog are specified here.
emailRecipients	<*mailtoRestart*>	string	Comma separated list of e-mail addresses to which notices about server state changes should be e-mailed
stableTime	60000	integer, 20000-300000 milliseconds	Time in milliseconds that has to pass before a server's status can be considered stable (for the purpose of sending out a server status e-mail notification).
/watchdog/servers	httpd nspoller dbpoller dispatcher worker scheduler lockmanager cornerstonebridge	string	Server.
/watchdog/waitDelay	3000	integer, 20000-300000 milliseconds	The time period for which the wait() calls in watchdog wait, before checking the wait condition, in milliseconds.
xml Properties:			The component for XML-based properties.

Table B-1 DCPL Properties (continued)

/xml/ValidatorRule			
filepath	<*vpnsc_home*>/resources/java/classes/common/com/cisco/vpnsc/util/validator/xml	string	Validator rules file path and name.
/xml/queries/			Properties for RepQueryLoader.
filepath	<*vpnsc_home*>/resources/java/xml/com/cisco/vpnsc/repository/Queries.xml	string	File path and name.

parseConfigAfterProvisioning	false	The valid values are **true** and **false**.	This property controls the parsing of the configuration file after the provisioning is completed in order to make sure that device inventory is in sync with network.

APPENDIX C

WatchDog Commands

The WatchDog is responsible for bootstrapping Cisco Prime Fulfillment and starting the necessary set of server processes. In addition, the WatchDog monitors the health and performance of each server to ensure it is functioning properly. In the event of a software error that causes a server to fail, the WatchDog automatically restarts the errant server.

The WatchDog is a background daemon process that is automatically installed as part of the installation procedure for Prime Fulfillment. After the installation procedure has completed, WatchDog is started automatically. You can execute the **startwd** command to run the WatchDog after the installation. The WatchDog can be configured to automatically start any time the machine is rebooted.

In addition to the commands that are specified in this chapter, in the product you can choose **Administration > Control Center > Hosts** and from there you can start, stop, restart, and view log files for the individual Prime Fulfillment servers.

This chapter provides the description, syntax, and arguments (listed alphabetically) for the following WatchDog commands:

startdb Command

This section provides the description and syntax for the **startdb** command.

Description

The **startdb** command starts the database.

Syntax

startdb

The **startdb** command has no arguments and starts the database.

The location of **startdb** is: <*ISC Directory*>**/bin**.

Note Do *not* run **startdb** in the background. Do *not* enter **startdb &**.

startns Command

This section provides the description and syntax for the **startns** command.

Description

The **startns** command starts the name server. The **orbd** process provides the name server functionality. **orbd** (from JDK) is required, but **startwd** starts it if it is not already running. The **startns** and **stopns** commands deal with **orbd**.

Syntax

startns

The **startns** command has no arguments and starts the name server.

The location of **startns** is: <*ISC Directory*>**/bin**.

startwd Command

This section provides the description and syntax for the **startwd** command.

Description

The **startwd** command starts the WatchDog and all Prime Fulfillment processes. The **startwd** command includes the functionality of **startdb** (see the "startdb Command" section on page C-1) and **startns** (see the "startns Command" section on page C-2). Executing this command is a necessary procedure and occurs automatically as part of the installation. Use this **startwd** command after issuing a **stopwd** command to restart the WatchDog.

If for some reason the Prime Fulfillment host is stopped, either inadvertently or by issuing the **stopwd** command, it can be restarted by using the **startwd** command.

Syntax

startwd

The **startwd** command has no arguments and starts the WatchDog only for the machine where it is executed.

The location of **startwd** is: <*ISC Directory*>**/bin**.

Note Do *not* run **startwd** in the background. Do *not* enter **startwd &**.

stopall Command

This section provides the description and syntax for the **stopall** command.

Description

The **stopall** command stops the database, name server, and WatchDog on the machine on which it is run. The **stopall** command includes the functionality of **stopdb -y** (see the "stopdb Command" section on page C-3), **stopns -y** (see the "stopns Command" section on page C-4), and **stopwd -y** (see the "stopwd Command" section on page C-4). Normally this is only necessary before installing a new version of Prime Fulfillment.

Syntax

stopall

Caution There is no **-y** parameter. Therefore, everything stops without the ability to cancel.

The location of **stopall** is: <*ISC Directory*>**/bin**.

stopdb Command

This section provides the description and syntax for the **stopdb** command.

Description

The **stopdb** command stops the database.

Syntax

stopdb [**-y**]

where:

-y indicates not to prompt before shutdown. If **-y** is not specified, you are prompted with the following message: "Are you absolutely sure you want to stop the database?" You are then prompted to reply **yes** or **no**.

The location of **stopdb** is: <*ISC Directory*>**/bin**.

stopns Command

This section provides the description and syntax for the **stopns** command.

Description

The **stopns** command stops the name server. The **startns** and **stopns** commands deal with **orbd**.

Syntax

stopns [**-y**]

where:

-y indicates not to prompt before shutdown. If **-y** is not specified, you are prompted with the following message: "Are you absolutely sure you want to stop the nameserver?" You are then prompted to reply **yes** or **no**.

The location of **stopns** is: <*ISC Directory*>**/bin**.

stopwd Command

This section provides the description and syntax for the **stopwd** command.

Description

The **stopwd** command stops the WatchDog and all Prime Fulfillment processes other than the name server and the database.

Syntax

stopwd [**-y**]

where:

-y indicates not to prompt before shutdown. If **-y** is not specified, you are prompted with the following message: "Are you absolutely sure you want to stop the watchdog and all of its servers? Other users may be using this system as well. No activity (for example: collections, performance monitoring, provisioning) occurs until the system is restarted." You are then prompted to reply **yes** or **no**.

The location of **stopwd** is: <*ISC Directory*>**/bin**.

wdclient Command

This section provides the description, syntax, and options (listed alphabetically) for the **wdclient** subcommands. These subcommands are diagnostic tools. This section also describes the column format of the output of each of the subcommands.

Note The location of **wdclient** is: <*ISC Directory*>**/bin**.

The following are the **wdclient** subcommands:

Note If you enter **wdclient -help**, you receive a listing of all the wdclient subcommands.

wdclient disk Subcommand

This section provides the description and syntax for the **wdclient disk** subcommand.

Description

The **wdclient disk** subcommand gives the disk space statistics for the directories where Prime Fulfillment is installed.

Syntax

wdclient disk

wdclient group *<group_name>* Subcommand

This section provides the description and syntax for the **wdclient group *<group_name>*** subcommand.

Description

The **wdclient group *<group_name>*** subcommand lists the servers in the specified server group. Server groups provide a convenient way to start or stop a group of servers with a single command.

Syntax

wdclient group *<group_name>*

where:

<group_name> is the name of a server group chosen from the list displayed by the **wdclient groups** command.

wdclient groups Subcommand

This section provides the description and syntax for the **wdclient groups** subcommand.

Description

The **wdclient groups** subcommand lists all the active server groups.

Syntax

wdclient groups

wdclient health Subcommand

This section provides the description and syntax for the **wdclient health** subcommand.

Description

The **wdclient health** subcommand indicates whether all the servers are stable.

Syntax

wdclient health

wdclient restart Subcommand

This section provides the description and syntax for the **wdclient restart** subcommand.

Description

The **wdclient restart** subcommand restarts one or more servers. Any dependent servers are also restarted.

Note It is not necessary to restart servers in a properly functioning system. The **wdclient restart** command should only be run under the direction of Cisco Support.

Syntax

wdclient restart [**all** | <*server_name*> | **group** <*group_name*>]

where you can choose one of the following arguments:

all is all servers. This is the default if no argument is specified.

<*server_name*> is the name of a server chosen from the list displayed by the **wdclient status** command. See Table C-1, "Servers and Their Functions," for server descriptions.

group <*group_name*> where, <*group_name*> is the name of a server group chosen from the list displayed by the **wdclient groups** command.

wdclient start Subcommand

This section provides the description and syntax for the **wdclient start** subcommand.

Description

The **wdclient start** subcommand starts one or more servers. Other servers that depend on the specified server(s) might also start.

Note It is not necessary to stop and start servers in a properly functioning system. The **wdclient start** command should only be run under the direction of Cisco Support.

Syntax

wdclient start [**all** | <*server_name*> | **group** <*group_name*>]

where you can choose one of the following arguments:

all is all servers. This is the default if no argument is specified.

<*server_name*> is the name of a server chosen from the list displayed by the **wdclient status** command. See Table C-1, "Servers and Their Functions," for server descriptions.

group <*group_name*> where, <*group_name*> is the name of a server group chosen from the list displayed by the **wdclient groups** command.

wdclient status Subcommand

This section provides the description, syntax, and information produced for the **wdclient status** subcommand.

Description

The **wdclient status** subcommand lists all the servers and their states. See Table C-1 on page C-8, "Servers and Their Functions," for server descriptions. See Table C-2 on page C-9, "Valid States," for the list of all the states.

Syntax

wdclient [**-poll** <*seconds*>] **status**

where:

-poll <*seconds*> is an optional parameter. <*seconds*> is the number of seconds. A number other than zero indicates that when new status data is available it is displayed every <*seconds*> seconds, where <*seconds*> is the specified number of seconds. The default **-poll** value is zero (0), which shows the status just once.

Information Produced: Name Column

The **Name** column provides the name of each of the servers. Table C-1 provides a list of the servers and a description of the function that each server provides.

Table C-1 Servers and Their Functions

Server	Function
cnsserver	Handles TIBCO messages from Cisco Configuration Engine servers and takes appropriate actions.
dbpoller	Monitors database server.
discovery	Devices and Service Discovery Engine.
dispatcher	Manages workers. Distributes work to other hosts (if any).
httpd	Web server.
lockmanager	Handles device locking so a router's configuration is not modified by multiple service requests at the same time.
nspoller	Monitors name service.
rgserver	Executes various Prime Fulfillment traffic engineering computations, such as tunnel repairing.

Table C-1 Servers and Their Functions (continued)

Server	Function
scheduler	Enables you to schedule tasks immediately or later in time, for one-time or repeated execution.
worker	Executes various Prime Fulfillment tasks/jobs such as Provisioning.

Information Produced: State Column

The **State** column provides the current state of the server. Table C-2 provides a description of each of the states in normal progression order.

Table C-2 Valid States

State	Description
start_depends	This server has been asked to start, but is waiting for servers it depends on to start. After all dependent servers have started, this server transitions to the state of starting.
starting	This server is currently starting. After a successful heartbeat occurs, this server transitions to the state of started.
started	This server is currently started and running.
stop_depends	This server is supposed to be stopped, but it is waiting for servers it depends on to be stopped first.
stopping_gently	This server is in the process of stopping in a gentle fashion. That is, it was notified that it is to stop.
stopping_hard	This server is in the process of being killed because either it did not have a way to stop gently or because the gentle stop took too long.
stopped	This server is stopped. The WatchDog either starts it again or disables it if it has been frequently dying.
disabled_dependent	This server is disabled because one or more servers it depends on are disabled. If all servers it depends on are started, this server automatically starts.
disabled	This server is disabled and must be manually restarted.
restart_delay	This server is delaying before restarting. There is a short delay after a server stops and before it is restarted again.

Information Produced: Gen Column

The **Gen** column provides the generation of the server. Each time the server is started, the generation is incremented by 1.

Information Produced: Exec Time Column

The **Exec Time** column provides the date and time the server was last started.

Information Produced: PID Column

The **PID** column provides the UNIX process identifier for each server, except for dbpoller and nspoller.

Information Produced: Success Column

The **Success** column provides the number of successful heartbeats since the server was last started. Heartbeats are used to verify that servers are functioning correctly.

Information Produced: Missed Column

The **Missed** column provides the number of missed heartbeats since the server was last started.

A few missed heartbeats could simply indicate the system was busy. However, more than a couple of missed heartbeats per day could indicate a problem. See the logs to diagnose the reason.

Three missed heartbeats in a row is the default for restarting the server.

wdclient stop Subcommand

This section provides the description and syntax for the **wdclient stop** subcommand.

Description

The **wdclient stop** subcommand stops one or more servers. Other servers that depend on the specified servers also stop.

Note It is not necessary to stop servers in a properly functioning system. The **wdclient stop** command should *only* be run under the direction of Cisco Support.

Syntax

wdclient stop [**all** | <*server_name*> | **group** <*group_name*>]

where you can choose one of the following arguments.

all is all servers. This is the default if no argument is specified.

<*server_name*> is the name of a server chosen from the list displayed by the **wdclient status** command. See Table C-1, "Servers and Their Functions," for server descriptions.

group <*group_name*> is the name of a server group chosen from the list displayed by the **wdclient groups** command.

APPENDIX D

Prime Fulfillment XML Reference

This appendix contains an alphabetical listing of the XML rules, tags, and attributes that are used in the XML files used for Cisco Prime Fulfillment Discovery.

For a detailed description of the XML files and XML examples, see Appendix G, "Inventory - Discovery."

Table D-1 IPrime Fulfillment XML Rules, Tags, and Attributes

Tag	Description
<as-number>	Specifies the autonomous system (AS) number for the provider. The AS number can be an integer between 1 and 65535.
<CDP>	Starts a **<CDP>** tag. The **<CDP>** tag specifies an seed IP address and a hop count. The **<CDP>** tag must contain the following attributes: • **ipaddress** • **hop**
<connection>	Starts a **<connection>** tag. The **<connection>** tag must specify the following attributes: • **discovery-protocol** • **fromDevice** • **FromIP** • **FromInterface** • **toDevice** • **toIP** • **toIF**
<create-customer>	Starts a **create-customer** rule. The **create-customer** rule creates a region object. the create-customer rule must contain the following tags: • **<customer-name>** • **<create-site>**

Table D-1 IPrime Fulfillment XML Rules, Tags, and Attributes (continued)

Tag	Description
<create-provider>	Starts a **create-provider** rule. The create-provider rule creates a service provider object. The **create-provider** rule must contain the following tags: • **<provider-name>** • **<as-number>** • **<create-region>**
<create-region>	Starts a **create-region** rule. The **create-region** rule creates a region object. The **create-region** rule must contain a **region-name** tag.
<create-site>	Starts a **create-site** rule. The create-site rule must contain a **<site-name>** tag.
<customer-name>	Specifies a customer name. Required within the **create-customer** rule.
<device>	Starts a **<device>** tag. The **<device>** tag must contain the following tags: • **<device-name>** • **<ip-address>** The following tags are optional within the **<device>** tag: • **<system-object-id>** • **<snmp-info>**
<device-name>	Specifies the name of the device. Required within the **<device>** tag.
<DISCOVERY_METHOD>	Starts a **<DISCOVERY_METHOD>** tag. The **<DISCOVERY_METHOD>** tag must contain a **<CDP>** tag.
discovery-protocol	Specifies the Discovery protocol used to discover the network topology. Normally, this is "CDP."
fromDevice	Specifies the name of the device from which the Named Physical Circuit starts. Required attribute for the **<connection>** tag.
FromInterface	Specifies the name of the device interface from which the Named Physical Circuit starts. Required attribute for the **<connection>** tag.
FromIP	Specifies the management IP address of the device from which the Named Physical Circuit starts. Required attribute for the **<connection>** tag.

Table D-1 IPrime Fulfillment XML Rules, Tags, and Attributes (continued)

Tag	Description
hop	Specifies the number of hops from the device identified by the **ipaddress** attribute to go in discovering devices. Required attribute for the **<CDP>** tag.
ipaddress	Specifies the IP address of a seed device. Required attribute for the **<CDP>** tag.
<ip-address>	Specifies the IP address of the device. Required within the **<device>** tag.
<provider-name>	Specifies the name of the provider.
<region-name>	Specifies the name of a region.
<ro-community>	Specifies the level of SNMP access for the device. Normally, this should be "public." Required within the **<snmp-info>** tag.
<site-name>	Specifies a site name.
<snmp-info>	Specifies SNMP information for the device. The **<snmp-info>** tag must contain a **<ro-community>** tag. Optional within the **<device>** tag.
<system-object-id>	(optional) Can be included to specify the SNMP Object ID (OID) for the device. If this is provided, it is specified within the **<device>** tag.
toDevice	Specifies the name of the device to which the Named Physical Circuit connects. Required attribute for the **<connection>** tag.
toIF	Specifies the device interface on the device to which the Named Physical Circuit connects. Required attribute for the **<connection>** tag.
toIP	Specifies the management IP address of the device from which the Named Physical Circuit connects Required attribute for the **<connection>** tag.

APPENDIX E

Terminating an Access Ring on Two N-PEs

This appendix describes how to terminate an access ring on two N-PEs for redundancy in case an access link goes down. It contains the following sections:

Overview

Prime Fulfillment supports device-level redundancy in the service topology. This allows the service to remain active in case one access link should drop. This is accomplished through support for provisioning termination of access links against two different N-PEs. This is implemented by allowing an access ring to terminate on two different N-PEs. This may also be described as a "dual-homed access ring." The N-PEs are connected by a logical link using loopback interfaces on the N-PEs. The redundant link starts from a U-PE device and may, optionally, include PE-AGG devices. One attachment link is primary and one is secondary. The selection is made when the Named Physical Circuit (NPC) is created. The terminating device on the NPC acts as the primary N-PE, while the other N-PE on the same ring acts as the secondary N-PE.

For backward compatibility, Prime Fulfillment continues to support provisioning services without redundant links, as in previous releases.

N-PE redundancy is supported for FlexUNI/EVC and MPLS services. As many of the basic concepts are shared for both services, both are covered in this appendix.

Figure E-1 and Figure E-2 show two network topologies which illustrate redundancy, starting from a U-PE access node. Both topologies provide open segments for each uplink, starting from the U-PE and terminating on the N-PE devices. The N-PEs are logically connected via loopback interfaces. Services are configured on both of these Ethernet access links starting from the U-PE to two different N-PEs.

Figure E-1 N-PE Redundancy, Starting at the U-PE

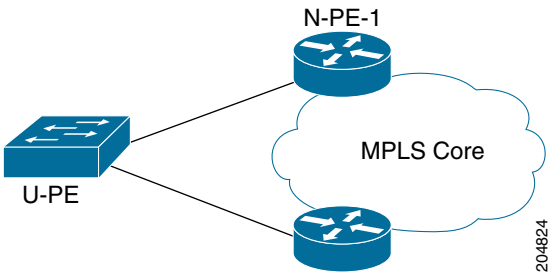
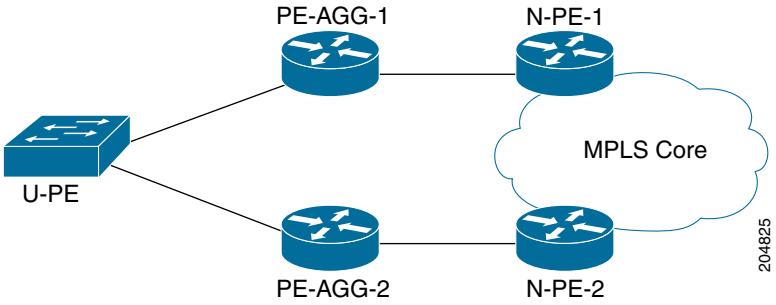


Figure E-2 N-PE and PE-AGG Redundancy, Starting at the U-PE



The first topology (N-PE redundancy starting at the U-PE, as shown in Figure E-1) provides the model of fault recovery for the N-PE device. As shown in the diagram, there are two different outgoing interfaces starting from the U-PE device. Each terminates at a different N-PE.

The second topology (N-PE and PE-AGG redundancy starting at the U-PE, as shown in Figure E-2) provides fault recovery for both the PE-AGG and N-PE devices. The service switches over from the primary to the secondary link when either the PE-AGG or the N-PE of the primary link fails.

For other network scenarios illustrating more complex topologies, see Additional Network Configurations and Sample Configlets, page E-5.

The following list provides additional details about the implementation:

- Using one U-PE and two N-PEs consumes one access link (AL).
- When creating a service on a U-PE, the user specifies an NPC to be used. If the topology includes an access ring with two N-PEs, then the service is configured on both N-PEs.
- For Ethernet over MPLS (EoMPLS) pseudowire (PW) services, if there is N-PE redundancy on both sides of the service provider network, two pseudowires are created. One N-PE is defined as primary and the other as secondary, in order to determine the how the pseudowires connect. If the user enables the PW Redundancy option, the primary and secondary on either end are also connected with pseudowire redundancy.
- For point-to-point (P2P) configurations, the two N-PEs use two separate pseudowires.
- Prime Fulfillment supports the case in which the service is configured identically (except for the access interface) on both N-PEs. This saves the user from having to enter data twice because the link attributes in the service request workflow are common for both N-PEs that are part of the attachment circuit.
- This feature is supported for both Cisco 7600 and Cisco ASR 9000 platforms. However, a single service cannot include both 7600 and ASR 9000 platforms.
- For the Cisco ASR 9000 platform, IOS XR version 3.7.3 and 3.9.0 are supported.

Note Check the on-line version of *Release Notes for Cisco Prime Fulfillment 6.2,* for the most current information on device and platform support, in case updates have occurred since the publication of this guide.

The implementation of this feature is covered in more detail in the following sections.

Setting Up an NPC Access Ring with Two N-PEs

Terminating an NPC access ring on two N-PEs is achieved by using the standard method of setting up an NPC ring in Prime Fulfillment. The basic steps for doing this are described Setting Up Logical Inventory, page 2-53. Additional information is provided in this guide in the section Creating Named Physical Circuits, page 3-11.

In normal cases, a ring would be closed by connecting the devices via physical interfaces. When terminating an access ring on two different N-PEs, there is no need for a physical connection between the N-PEs. However, Prime Fulfillment requires that a virtual link must be created between the N-PEs, in order to close the ring. The virtual link is set up through the use of loopback interfaces.

In order to use loopback interfaces in a ring in this manner, you must enable the DCPL property allowLoopbackIntfInNPC, which is accessed in the Host Configuration window under the folder /repository/mlshare. When this DCPL property is set to true, Prime Fulfillment allows the use of loopback interfaces in a ring.

Note Note that Prime Fulfillment does not generate any configlets onto the loopback interfaces during deployment of the service request.

Using N-PE Redundancy in FlexUNI/EVC Service Requests

Using a dual-homed access ring in a FlexUNI/EVC service request does not require any change in the usual workflow in the Prime Fulfillment GUI. During creation of the FlexUNI/EVC service request, you select the NPC which is associated with an NPC access ring terminating on two N-PEs.

Usage notes:

- The service is configured on both N-PEs of the access ring.
- Though there are two different N-PEs, only one access link is consumed.
- You can modify the configuration redundant N-PEs before or after deploying the service request. Modified configlets will be generated according to the changes made in service request.
- The destined N-PE device on the NPC used in the service request is treated as the primary N-PE. The other N-PE on the same ring is treated as the secondary N-PE. To change the primary and secondary N-PE, you must modify the attachment circuits in the service request.
- Configlets are generated according to the configuration specified in the service request. Prime Fulfillment generates identical configlets on both of the N-PEs in the attachment circuit (AC). The Link Attributes sections are common for both N-PEs.
- For FlexUNI/EVC services, N-PE redundancy is supported for PSEUDOWIRE and VPLS core connectivity types.

- In case of VPLS core connectivity, all N-PEs in NPC rings are configured to have Layer 2 Virtual Forwarding Interface (VFI), and all N-PEs on the same VPLS VPN participate in the VPLS service at the same time.
- In the case of PSEUDOWIRE core connectivity, the following notes apply:
  - If there is N-PE redundancy on both sides, a point-to-point pseudo wire (PW) will be configured between the N-PEs that were specified as the terminating N-PE devices during the NPC creation (between primary N-PEs). One more point-to-point PW will be configured between the N-PEs that were not specified as the terminating N-PE devices during NPC creation. The VC IDs of these pseudowires will be common.
  - If there is N-PE redundancy on only one side, then the Pseudowire Redundancy option must be checked in the GUI (in the Service Request Details section of the of the FlexUNI(EVC) Service Editor window). The primary PW will connect the primary N-PE of the dual-homed ring with the N-PE of the single-homed ring, and the secondary PW will connect the secondary N-PE of the dual-homed ring with the N-PE of the single-homed ring. Prime Fulfillment will issue a warning message if you try to save the service request without enabling the Pseudowire Redundancy option.

Using N-PE Redundancy in MPLS Service Requests

Access ring termination on two N-PEs is supported for MPLS/L3 services for the Regular PE-CE policy type. The process of creating the NPC rings and associating them into the MPLS service is similar to that covered in Using N-PE Redundancy in FlexUNI/EVC Service Requests, page E-3. There are not any changes to the standard MPLS service request workflow.

Usage notes:

- The service is configured on both N-PEs of the access ring in the PE_NO_PE case. However, in the PE_CE case, the service request is configured on the primary N-PE of the access ring.
- Though there are two different N-PEs, only one access link is consumed.
- You can modify the configuration-redundant N-PEs before or after deploying the service request. Modified configlets will be generated according to the changes made in the service request.
- The destined N-PE device on the NPC used in the service request is treated as the primary N-PE. The other N-PE on the same ring is treated as the secondary N-PE.
- To change the primary N-PE, delete and recreate the NPC, provided the NPC is not associated with any service requests. To change the secondary N-PE, you have to modify the secondary N-PE at the ring level.
- During MPLS service request creation using the PE_NO_CE policy, the secondary NPE device can be configured through the second link. Separate link attributes such as VLAN ID, PE Interface
- Address/Mask, VPN and RD and others can be configured separately for both primary and secondary N-PEs. This way you can manually add a different IP address on primary and secondary N-PEs. UNI device information will be available only in the link of the primary N-PE.
- During MPLS service request creation using the PE-CE policy, only one MPLS VPN link would be created even though the selected NPC has two N-PEs. Service can be associated only to the primary N-PE, no additional link will be provided for the secondary N-PE. Configlets will be generated and pushed to all the devices in the ring except the secondary N-PE.
- VPNs and VRF objects are supported for MPLS service requests using access ring termination on two N-PEs.

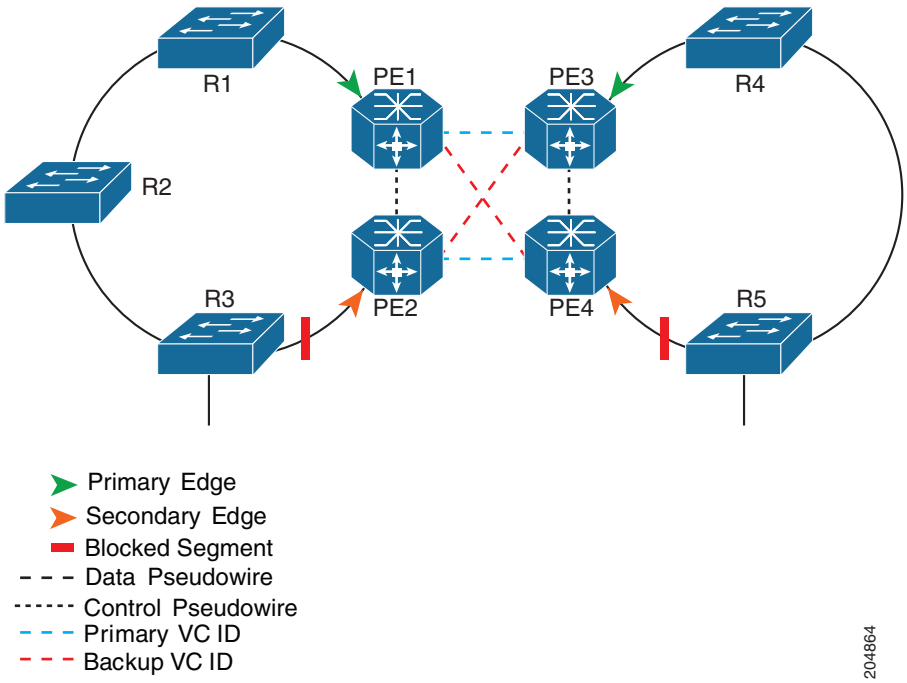
Additional Network Configurations and Sample Configlets

This section provides additional network scenarios for reference, along with sample configlets for associated network devices.

Example 1: Pseudowire Connectivity (A)

Figure E-3 illustrates a network configuration with pseudowire connectivity with dual-homed N-PEs on both sides of the network and with pseudowire redundancy.

Figure E-3 Pseudowire Connectivity, Dual-Homed N-PEs on Both Sides of the Network, with Pseudowire Redundancy



Sample configlets for the devices are provided below.

PE1

```
vlan <S-Vlan>
!
interface <UNI-to-R1>
    switchport
    switchport trunk encapsulation dot1q
    switchport mode trunk
    switchport trunk allowed vlan add <S-Vlan>
!
interface vlan <S-Vlan>
    xconnect <PE3 loopback> <PrimaryVcId> encapsulation mpls
    backup peer <PE4 loopback> <BackupVcId>
```

PE2

```
vlan <S-Vlan>
!
interface <UNI-to-R3>
    switchport
    switchport trunk encapsulation dot1q
    switchport mode trunk
    switchport trunk allowed vlan add <S-Vlan>
!
interface vlan <S-Vlan>
    xconnect <PE4 loopback> <PrimaryVcId> encapsulation mpls
    backup peer <PE3 loopback> <BackupVcId>
```

PE3

```
vlan <S-Vlan>
!
interface <UNI-to-R4>
    switchport
    switchport trunk encapsulation dot1q
    switchport mode trunk
    switchport trunk allowed vlan add <S-Vlan>
!
interface vlan <S-Vlan>
    xconnect <PE1 loopback> <PrimaryVcId> encapsulation mpls
    backup peer <PE2 loopback> <BackupVcId>
```

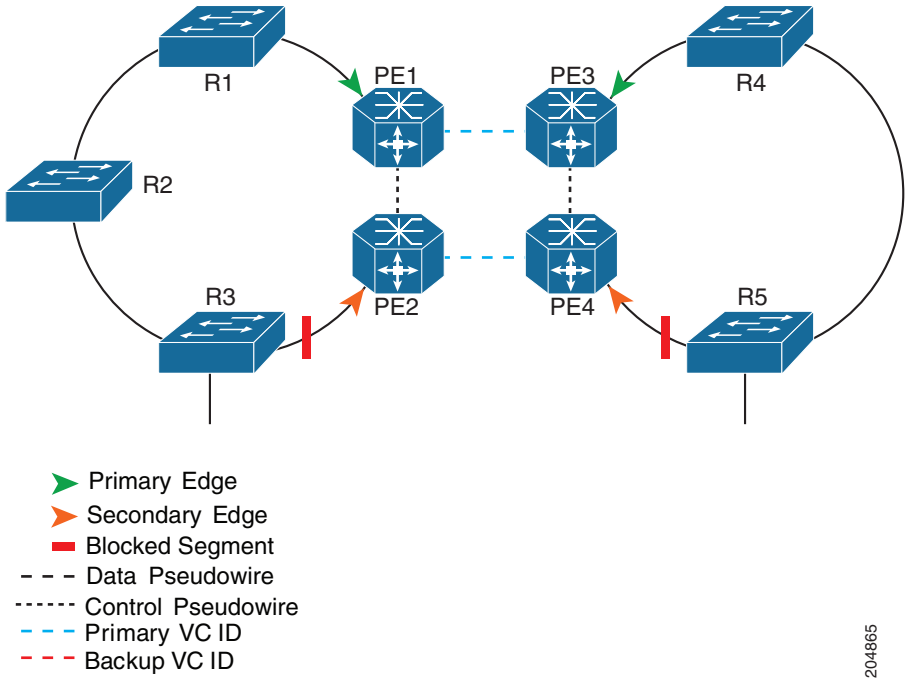
PE4

```
vlan <S-Vlan>
!
interface <UNI-to-R5>
    switchport
    switchport trunk encapsulation dot1q
    switchport mode trunk
    switchport trunk allowed vlan add <S-Vlan>
!
interface vlan <S-Vlan>
    xconnect <PE2 loopback> <PrimaryVcId> encapsulation mpls
    backup peer <PE1 loopback> <BackupVcId>
```

Example 2: Pseudowire Connectivity (B)

Figure E-4 illustrates a network configuration using pseudowire connectivity, with dual-homed N-PEs on both sides of the network without pseudowire redundancy.

Figure E-4 Pseudowire Connectivity, Dual-Homed N-PEs on Both Sides of the Network, with No Pseudowire Redundancy



Sample configlets for the devices are provided below.

PE1

```
vlan <S-Vlan>
!
interface <UNI-to-R1>
    switchport
    switchport trunk encapsulation dot1q
    switchport mode trunk
    switchport trunk allowed vlan add <S-Vlan>
!
interface vlan <S-Vlan>
    xconnect <PE3 loopback> <PrimaryVcId> encapsulation mpls
```

PE2

```
vlan <S-Vlan>
!
interface <UNI-to-R3>
    switchport
    switchport trunk encapsulation dot1q
    switchport mode trunk
    switchport trunk allowed vlan add <S-Vlan>
!
interface vlan <S-Vlan>
    xconnect <PE4 loopback> <PrimaryVcId> encapsulation mpls
```

PE3

```
vlan <S-Vlan>
!
interface <UNI-to-R4>
    switchport
    switchport trunk encapsulation dot1q
    switchport mode trunk
    switchport trunk allowed vlan add <S-Vlan>
!
interface vlan <S-Vlan>
    xconnect <PE1 loopback> <PrimaryVcId> encapsulation mpls
```

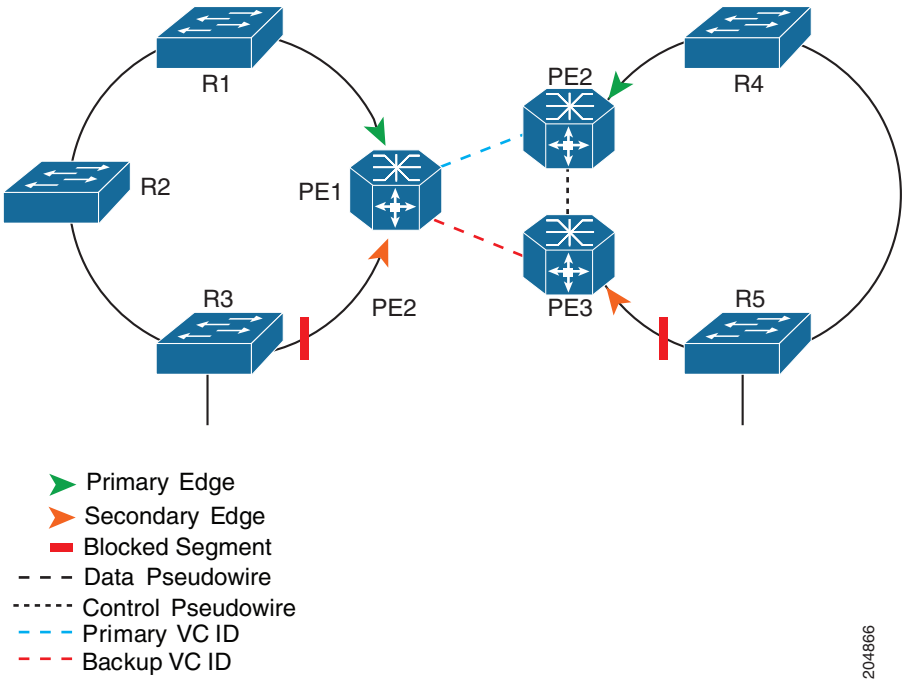
PE4

```
vlan <S-Vlan>
!
interface <UNI-to-R5>
    switchport
    switchport trunk encapsulation dot1q
    switchport mode trunk
    switchport trunk allowed vlan add <S-Vlan>
!
interface vlan <S-Vlan>
    xconnect <PE2 loopback> <PrimaryVcId> encapsulation mpls
```

Example 3: Pseudowire Connectivity (C)

Figure E-5 illustrates a network configuration using pseudowire connectivity with dual-homed N-PEs at one side of the network and with pseudowire redundancy.

Figure E-5 Pseudowire Connectivity, Dual-Homed N-PEs on One Side of the Network, with Pseudowire Redundancy



Sample configlets for the devices are provided below.

PE1

```
vlan <S-Vlan>
!
interface <UNI-to-R1>
    switchport
    switchport trunk encapsulation dot1q
    switchport mode trunk
    switchport trunk allowed vlan add <S-Vlan>
!
interface vlan <S-Vlan>
    xconnect <PE2 loopback> <PrimaryVcId> encapsulation mpls
    backup peer <PE3 loopback> <BackupVcId>
```

PE2

```
vlan <S-Vlan>
!
interface <UNI-to-R4>
    switchport
    switchport trunk encapsulation dot1q
    switchport mode trunk
    switchport trunk allowed vlan add <S-Vlan>
!
interface vlan <S-Vlan>
    xconnect <PE1 loopback> <PrimaryVcId> encapsulation mpls
```

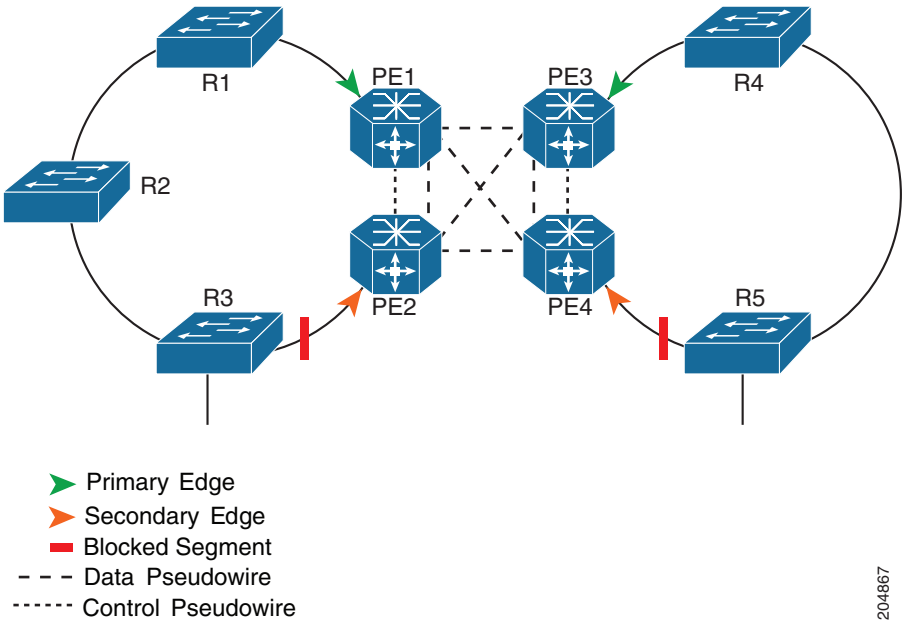
PE3

```
vlan <S-Vlan>
!
interface <UNI-to-R5>
    switchport
    switchport trunk encapsulation dot1q
    switchport mode trunk
    switchport trunk allowed vlan add <S-Vlan>
!
interface vlan <S-Vlan>
    xconnect <PE1 loopback> <BackupVcId> encapsulation mpls
```

Example 4: VPLS Connectivity

Figure E-6 illustrates a network configuration using VPLS connectivity with dual-homed N-PEs on both sides of the network.

Figure E-6 VPLS Connectivity, Dual-Homed N-PEs on Both Sides of the Network



Sample configlets for the devices are provided below.

PE1

```
vlan <S-Vlan>
!
l2 vfi <VFI-ID> manual
    vpn id <S-Vlan>
    neighbor <PE2> encapsulation mpls
    neighbor <PE3> encapsulation mpls
    neighbor <PE4> encapsulation mpls
!
interface vlan <S-Vlan>
    xconnect vfi <VFI-ID>
!
interface <NNI-to-R1>
    switchport trunk allowed vlan add <S-Vlan>
```

PE2

```
vlan <S-Vlan>
!
l2 vfi <VFI-ID> manual
    vpn id <S-Vlan>
    neighbor <PE1> encapsulation mpls
    neighbor <PE3> encapsulation mpls
    neighbor <PE4> encapsulation mpls
!
interface vlan <S-Vlan>
    xconnect vfi <VFI-ID>
!
interface <NNI-to-R3>
    switchport trunk allowed vlan add <S-Vlan>
```

PE3

```
vlan <S-Vlan>
!
l2 vfi <VFI-ID> manual
    vpn id <S-Vlan>
    neighbor <PE1> encapsulation mpls
    neighbor <PE2> encapsulation mpls
    neighbor <PE4> encapsulation mpls
!
interface vlan <S-Vlan>
    xconnect vfi <VFI-ID>
!
interface <NNI-to-R5>
    switchport trunk allowed vlan add <S-Vlan>
```

PE4

```
vlan <S-Vlan>
!
l2 vfi <VFI-ID> manual
    vpn id <S-Vlan>
    neighbor <PE1> encapsulation mpls
    neighbor <PE2> encapsulation mpls
    neighbor <PE3> encapsulation mpls
!
interface vlan <S-Vlan>
    xconnect vfi <VFI-ID>
!
interface <NNI-to-R4>
    switchport trunk allowed vlan add <S-Vlan>
```

APPENDIX F

Repository Views

A view is a stored query accessible as a virtual table composed of the result set of a query. Unlike ordinary tables (base tables) in a relational database, a view does not form part of the physical schema; it is a dynamic and virtual table computed or collated from data in the database. Changing the data in a table alters the data shown in subsequent invocations of the view.

The advantages of repository views are as follows:

- Data security: Provides an additional level of table security by restricting access to a pre-determined set of rows and/or columns of a table.
- Provides an easy way to query data from different data sources like a single table.
- Useful when developing complex reports based on multiple tables.

This appendix contains the following sections:

Creating Repository Views

This section describes how to create views in Sybase repository and Oracle repository.

Creating Views Sybase Repository

New and Upgrade Installation

All the views available in Cisco Prime Fulfillment (see the Using Views in Prime Fulfillment, page F-2) are created as part of the new and upgrade installation of Prime Fulfillment 6.2.

Creating Views in Oracle Repository

New and Upgrade Installation

To create repository views (see the Using Views in Prime Fulfillment, page F-2) in new and upgrade installation of Cisco Prime Fulfillment 6.2, follow these steps:

Step 1 Copy the **schema.tar** file to the Oracle server and then extract all files into a directory.

Note The schema information is held in the schema.tar file in the software package. Obtain the correct package (schemas can change between packages) and extract the **schema.tar** file from the package.

Step 2 Navigate to the directory containing the expanded schema, then go to the **ddl/6.0** sub-directory.

Step 3 Run the command **sqlplus**.

Step 4 Log in as sysdba and provide the DBA privileges to the Prime Fulfillment user using the command: **GRANT DBA, CONNECT, RESOURCE TO <isc_user>;**

Step 5 Log in with the username and password previously created.

Step 6 Enter the SQL command **start DBViews.sql;**

This will create all the views in Oracle repository.

Using Views in Prime Fulfillment

The different views available in Prime Fulfillment are as follows:

- Summary View, page F-2
- Site View, page F-4
- Customer View, page F-5
- Region View, page F-5

Summary View

You can query using the column name for summary view. Table F-1 describes the column name and its type name.

Table F-1 Summary view column names

Column Name	Type Name
SR_Number	Integer
SR_STATE	Integer
SR_Last_Modified_Time	Varchar

Table F-1 Summary view column names (continued)

Column Name	Type Name
PE_Name	Varchar
PE_Interface	Varchar
PE_Interface_IPAddress	Varchar
CE_Name	Varchar
CE_Interface	Varchar
CE_Interface_IPAddress	Varchar
CE_Type	Integer
CE_Site_ID	Integer
CE_Site_Name	Varchar
VPN_Name	Varchar
VRF_Name	Varchar
Customer_ID	Integer
Customer_Name	Varchar
JOB_DESCRIPTION	Varchar

The description of the column name is as follows:

- SR_Number—Service Request Number, represents the service request JOB ID that is available on the Service Request page in the Prime Fulfillment GUI
- SR_STATE—State of the Service Request and the following table maps the value in the database and its associated state:

Database Value	Associated State
-1	UNKNOWN
0	All States
1	Requested
2	Pending
3	Failed Deploy
4	InValid
5	Deployed
6	Broken
7	Functional
8	Lost
9	Closed
10	Failed Audit
11	Wait Deploy
12	In Progress

- SR_Last_Modified_Time—last modified time of SR based on the current state of the SR
- PE_Name—PE Host Name
- PE_Interface—PE Interface Name associated with SR.
- PE_Interface_IPAddress—IP address of the PE interface
- CE_Name—CE Host Name
- CE_Interface—CE interface name associated with SR
- CE_Interface_IPAddress—IP address of the CE interface
- CE_Type—Management type of the CE Device, the following table maps the value in the database and the CE Management Type:

Database Value	CE Management Type
-1	UNKNOWN
0	Managed
1	UnManaged
2	Managed - Management LAN
3	UnManaged - Management LAN
4	Directly Connected
5	Directly Connected Management Host
6	Multi-VRF
7	Un Managed Multi-VRF

- CE_Site_ID—Site ID of the CE
- CE_Site_Name—Site name of the CE
- VPN_Name—VPN name associated with SR
- VRF_Name—VRF name associated with SR
- Customer_ID—Customer ID
- Customer_Name—Customer Name
- JOB_DESCRIPTION—Job description of MPLS SR

An example for the summary view query is as follows:

```
select SR_Number, PE_Name, CE_Name, VPN_Name from Summary_View;
```

Site View

You can query using the column name for site veiw. Table F-2 describes the column name and its type name.

Table F-2 Site view column names

Column Name	Type Name
SITE_ID	Integer
SITE_NAME	Varchar
CPE_Name	Varchar
LINK_ID	Integer

The description of the column name is as follows:

- SITE_ID—Site ID
- SITE_NAME—Site Name
- CPE_Name—CPE name associated with the site
- LINK_ID—Link ID of the CPE associated to a SR

An example for the site view query is as follows:

```
select Site_Id, Site_Name, CPE_Name, Link_ID from Site_View;
```

Customer View

You can query using the column name for customer view. Table F-3 describes the column name and its type name.

Table F-3 Customer view column names

Column Name	Type Name
CUSTOMER_ID	Integer
CUSTOMER_CONTACT	Varchar

The description of the column name is as follows:

- CUSTOMER_ID—Customer ID
- CUSTOMER_CONTACT—Information about the customer

An example for the customer view query is as follows:

```
select * from Customer_View;
```

Region View

You can query using the column name that is available for region view. Table F-4 describes the column name and its type name.

Table F-4 Region view column name

Column Name	Type Name
PROVIDER_ID	Integer
REGION_ID	Integer
PE_NAME	Varchar

The description of the column name is as follows:

- PROVIDER_ID—Provider ID
- REGION_ID—Region ID of the provider
- PE_NAME—PE Host Name associated to this Region

An example for the region view query is as follows:

```
select Region_Id, PE_Name from Region_View;
```

APPENDIX G

Inventory - Discovery

This appendix describes how to use the Discovery feature to discover devices, connections, and services for the Cisco Prime Fulfillment provisioning process. It contains the following sections:

Overview of Prime Fulfillment Discovery

Prime Fulfillment discovery is designed to be an aid in installing Prime Fulfillment in a brown-field network, that is a network that has pre-existing services before Prime Fulfillment is introduced. Prime Fulfillment discovery is not a mechanism to repeatedly discover services and synchronize with the database. After the initial discovery has been completed, all provisioning must then be done using Prime Fulfillment. If services are provisioned directly on the network without using Prime Fulfillment, these services are not known by Prime Fulfillment and Prime Fulfillment might overwrite or conflict with these services. Therefore, any services provisioned outside of Prime Fulfillment must be brought into Prime Fulfillment, by means of provisioning them in Prime Fulfillment through the Graphical User Interface (GUI) or the Application Program Interface (API). This can be done in echo mode, so that a subsequent configuration audit confirms that the network and Prime Fulfillment database are synchronized.

In general, Prime Fulfillment can only discover a subset of what it can provision, so Prime Fulfillment cannot discover a service of a type that it cannot provision.

Prime Fulfillment can expedite the process for building a network device inventory by discovering the devices, connections, and services that your MPLS VPN or L2VPN Metro Ethernet network comprises.

Note Service discovery is a complex operation that can be impacted by many variables within the network. The original network configuration must have been performed in accordance with the same rules that Prime Fulfillment follows when provisioning services. Otherwise, errors might occur during the discovery. As a result of the many possible configurations in a given network, it is strongly recommended that you contact your Cisco account team or Cisco advanced services to provide support, before committing to the service discovery process.

Users who run service discovery should have a thorough understanding of their overall network topology, should be familiar with network terminology, such as: PE, N-PE, U-PE, PE-AGG, and CE, and should understand the definition of NPC and Metro Ethernet/MPLS services in Prime Fulfillment.

Prime Fulfillment supports the discovery process for admin users only.

The Prime Fulfillment Discovery feature can be used to populate repository for three of the applications in the Cisco Prime Fulfillment application suite:

- Cisco Prime Fulfillment MPLS VPN Management
- Cisco Prime Fulfillment L2VPN Management
- Cisco Prime Fulfillment Prime Diagnostics

Note Service discovery does not support Secure Shell version 2 (SSHv2) as a terminal session protocol. MPLS and L2VPN service discovery do not support devices running IOS XR.

When a device in Prime Fulfillment only has a hostname, the Prime Fulfillment device has no IP management address or domain name configured. If in Discovery, a device with the same hostname is discovered with an IP management address or is created manually in the Device Editor, the device might fail to commit to the Prime Fulfillment repository. The failure occurs because a match is determined with the existing Prime Fulfillment device, because both devices do not have a configured domain name.

The workaround is to do either 1. or 2., as follows:

1. Edit the device that exists in Prime Fulfillment and add the management IP address before Discovery. Discovery then treats that device as a duplicate and marks it read-only in the Device Editor.

   or

2. During Discovery, in the Device Editor, enter a domain name for the discovered device. Discovery then treats this as a new device.

The Cisco Prime Fulfillment Traffic Engineering Management has its own Discovery interface and process. This is documented in TE Network Discovery of the *Cisco Prime Fulfillment User Guide 6.2*.

Multiple service discovery processes are supported and you can restart from any of the previous steps. Support for multiple discovery processes allows you to do incremental discovery of the network. The ability to restart from previous steps helps you roll back the discovery process to a selected previous step. You can then resume discovery from that step instead of needing to restart the entire discovery process from the beginning. Restarting from discovery data collection prompts the user to select devices for which data needs to be collected.

Incremental discovery occurs for existing VPN links. The existing VPNs are not editable in the discovery GUI and the existing VPN links are by-passed during commit.

There is no synchronization in MPLS and L2VPN service discovery. Any modification must be done manually through the Prime Fulfillment user interface. Only new VPNs are discovered. Also, services on existing modified NPCs and conflicting NPCs are not discovered.

The commit to Prime Fulfillment happens only at the end of the discovery phase, not after each step. The Discovery process does not change the state of Prime Fulfillment during discovery workflow. It is only at the end of the workflow that a user can commit the discovered devices and services to Prime Fulfillment.

The Discovery process provides you with several choices on how to discover your network topology.

3. If you are running Discovery to provision Cisco Prime Fulfillment MPLS VPN Management or Cisco Prime Fulfillment L2VPN Management, you can choose between three Discovery methods:
   a. CDP Discovery

      You can use the Cisco Discovery Protocol (CDP) to discover devices connected to an initial device that has an IP address you provide in a **policy.xml** file.

   b. Device/Topology Based Discovery

      You can use a Device/Topology-based method. This method uses XML files that specify device and NPC topology information.

   c. Import Configuration File Based

      You can use an Import Configuration Files-based method. This method uses a directory on the server that contains configuration files for the devices to be discovered and an XML file that contains device connectivity information that is used to automatically create NPCs.
4. You can choose the network topology to discover an MPLS VPN topology, an L2VPN (Metro Ethernet) topology, or both.

   If you choose L2VPN (Metro Ethernet) Discovery, you can discover either a Metro Ethernet with an MPLS core, a Metro Ethernet with an Ethernet core, or a combination of the two, a mixed core. In a mixed core, the L2VPN services can span across the MPLS core or they can be confined to a local Ethernet domain alone (local switched services). Local switched services that do not traverse N-PE devices across an ethernet domain can also be discovered. Figure G-1, shows a mixed core.

Figure G-1 Mixed Core

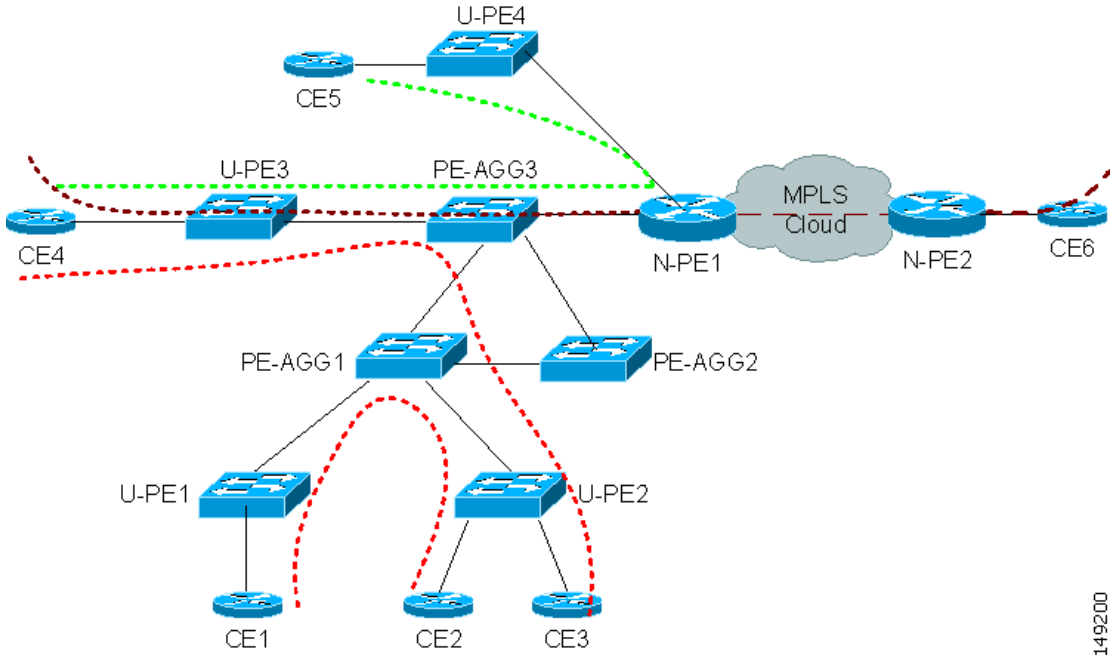


Figure G-2 illustrates the phases in the Discovery process.

Figure G-2 Prime Fulfillment Discovery Steps

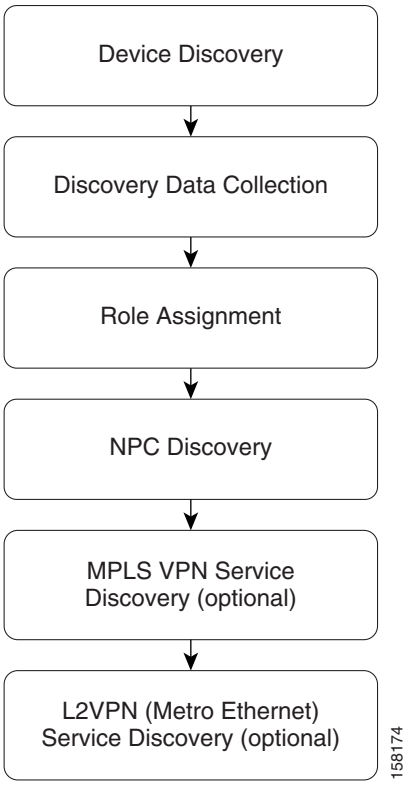


Table G-1 describes the phases in the Discovery process.

Table G-1 Steps in the Discovery Process

Step	Description
Device Discovery	Discovers devices in the MPLS VPN and/or Metro Ethernet topology.
Discovery Data Collection	Collects the IOS configuration for the devices discovered.
Role Assignment	Does the role assignment for the discovered devices based on rules.xml, and prompts you to edit the device roles as N-PE, U-PE, or CE. **Note** A sample is found at: $PRIMEF_HOME/resources/discovery/data/ rules.xml, where the rules.xml file must be kept.
NPC Discovery	Displays discovered NPCs and allows addition or removal of NPCs.
MPLS VPN Discovery	Discovers the topology for your MPLS VPN network and allows you to change it as required. **Note** The MPLS VPN Discovery step is not required if you are using Prime Fulfillment Discovery with Prime Diagnostics.
(L2VPN) Metro Ethernet Discovery	Discovers the topology for your Metro Ethernet network and allows you to change it as required. **Note** The (L2VPN) Metro Ethernet Discovery step is not required if you are using Prime Fulfillment Discovery with Prime Diagnostics.

The following discovery operations are not supported by Prime Fulfillment:

- Initial discovery of MPLS VPN services on devices running IOS XR
- Initial discovery of Ethernet access (U-PE, PE-AGG) into MPLS VPN services
- Initial discovery of VRF-lite/MVRF
- Initial discovery of VRF with multicast configuration
- Initial discovery of MPLS VPN services with unique route distinguishers
- Initial discovery of VRF that is not associated with an interface
- Initial discovery of VRF associated with one interface and no other VRF is in service among discovered devices
- Discovered extranet, unless the user manually splits VPNs
- Discovery of MPLS VPN/MLS VRF which is attached to a Loopback interface
- Initial discovery of services using the new ethernet service instance (EVC) syntax that was introduced in 12.2(33) SRB on the Cisco 7600 Series Routers

- Initial discovery of L2VPN services on devices running IOS XR
- Discovery of ERS and ERMS services with access port (only discovery with VLAN access is supported)
- Discovery of ATM or Frame Relay services (only discovery of Ethernet services is supported)
- Incremental NPC synchronization
- Re-synchronization, including discrepancy management

Technical Notes for Prime Fulfillment Discovery

This section presents technical tips, general information, and limitations about the Prime Fulfillment Discovery process.

The Prime Fulfillment Discovery feature can be used to populate repository for three of the applications in the Cisco Prime Fulfillment application suite:

- Cisco Prime Fulfillment MPLS VPN Management
- Cisco Prime Fulfillment L2VPN Management
- Cisco Prime Fulfillment Prime Diagnostics

Although the general steps are similar, there are some differences in the workflow for the various types of Discovery. These are described in the section covering each Prime Fulfillment application:

- Using Prime Fulfillment Discovery with Cisco Prime Fulfillment MPLS VPN Management, page G-7
- Using Prime Fulfillment Discovery With Cisco Prime Fulfillment L2VPN Management, page G-7
- Using Prime Fulfillment Discovery with Cisco Prime Fulfillment Prime Diagnostics, page G-8
- Using Prime Fulfillment Discovery With Cisco Prime Fulfillment Traffic Engineering Management, page G-9

Note Cisco Prime Fulfillment Traffic Engineering Management has its own Discovery interface and process.

This is documented in TE Network Discovery of the *Cisco Prime Fulfillment User Guide 6.2*.

For technical notes on using Prime Fulfillment Discovery in installations that include both Cisco Prime Fulfillment Traffic Engineering Management and Cisco Prime Fulfillment MPLS VPN Management, see Using Prime Fulfillment Discovery With Cisco Prime Fulfillment Traffic Engineering Management, page G-9.

General Notes

Note the following points before running Prime Fulfillment Discovery:

- You can use the Prime Fulfillment GUI to create providers, customers, and resource pools before doing Discovery.
- Only one user can control the Discovery workflow interface at a given time.

- The procedures in the chapter show a “generic” procedure. If you do not have licenses for a particular application, you will not see the selections for that application on the start screen for Prime Fulfillment Discovery.
- Perform “manual” device collection after discovery is over.
- After you have started the Discovery process, a **Restart** button appears on the Discovery Workflow window. You can click the **Restart** button, a drop-down list of completed steps pops up and you can select a step and restart from that step.
- Restarting from initialization aborts the current discovery process.
- Discovery using Role Based Access Control (RBAC) is not supported.

Using the Discovery Log Files

A log file is written for each phase of the Discovery process. You can view a log file by clicking the **View** selection in the Log column next to each discovery phase summary on the Discovery Workflow window.

The log file provides useful information in the event a discovery step fails.

Using Prime Fulfillment Discovery with Cisco Prime Fulfillment MPLS VPN Management

If you are running the Discovery process to discover an MPLS VPN network for use with Cisco Prime Fulfillment MPLS VPN Management, note the following points:

- You must perform all of the main steps in the Discovery process.
- You can use either CDP Discovery, Device/Topology, or Import Configuration Files-based Discovery. The recommendation is to use either Device/Topology or Import Configuration Files-based Discovery.
- Prime Fulfillment does not support partial mesh VPN topologies. If the Discovery process discovers a Partial Mesh VPN, you must split the partial mesh VPN into smaller units (usually a combination of full mesh VPNs and Hub and Spoke VPNs).
- After completion of the automated Discovery process, you must schedule and run a **Operate > Tasks > Task Manager > Collect Config** task for all discovered devices.

Note There is no synchronization in MPLS service discovery. Any modification must be done manually through the Prime Fulfillment user interface. Only new VPNs are discovered. Also, services on existing modified NPCs and conflicting NPCs are not discovered.

Using Prime Fulfillment Discovery With Cisco Prime Fulfillment L2VPN Management

If you are running the Discovery process to discover an L2VPN network that will be provisioned and managed using Cisco Prime Fulfillment L2VPN Management, note the following points:

- You must perform all of the main steps in the Discovery process.

- You can use either CDP Discovery, Device/Topology, or Import Configuration Files-based Discovery. The recommendation is to use either Device/Topology or Import Configuration Files-based Discovery.
- A new L2VPN service is discovered when any of the following are found compared to the services existing in Prime Fulfillment:
  - A new Virtual LAN Identifier (VLAN ID) in an Ethernet core (Ethernet access domain)
  - A new Virtual Circuit Identifier (VC ID) for virtual private wire service (VPWS) services on an MPLS core.
  - A new VPLS Forwarding Instance Identifier (VFI ID) for virtual private LAN service (VPLS) services on an MPLS core.
- The Discovery process for Cisco Prime Fulfillment L2VPN Management can discover Metro Ethernets with an MPLS core, an Ethernet core, or both.
- Prior to performing the NPC Discovery step for Cisco Prime Fulfillment L2VPN Management, you must specify the Access Domain for N-PE devices.
- Any new links that are configured on NPCs marked as Existing Modified or Conflicting are not discovered.
- After completion of the automated Discovery process, you must schedule and run a **Task Manager > Collect Config** task for all discovered devices.

Note There is no synchronization in L2VPN service discovery. Any modification must be done manually through the Prime Fulfillment user interface. Only new VPNs are discovered. Also, services on existing modified NPCs and conflicting NPCs are not discovered.

Using Prime Fulfillment Discovery with Cisco Prime Fulfillment Prime Diagnostics

If you are running the Discovery process to discover an MPLS VPN network for use with Prime Diagnostics, note the following points.

- You can use either CDP Discovery, Device/Topology, or Import Configuration Files-based Discovery. The recommendation is to use either Device/Topology or Import Configuration Files-based Discovery.
- For Cisco Prime Fulfillment Prime Diagnostics, you only need to perform the Device Discovery, Discovery Data Collection, and Role Assignment Steps. You do not need to perform the NPC Discovery step or the Service Discovery step. However, you can let the NPC Discovery process run.

  See Figure G-5 for a flowchart of the required steps for Prime Fulfillment Discovery with Cisco Prime Fulfillment Prime Diagnostics.
- If you are using Cisco Prime Fulfillment Prime Diagnostics, then you normally only need to discover P and PE devices. Therefore, when you perform the Role Assignment step for discovered devices, you only need to assign roles to the P and PE devices.

Note If you do discover any CE devices, you must assign them CE roles.

- After completion of the automated Discovery process, you must schedule and run a **Task Manager > Collect Config** task for all discovered devices.

Using Prime Fulfillment Discovery With Cisco Prime Fulfillment Traffic Engineering Management

Normally you do not have to run the Prime Fulfillment Discovery process if you are using Cisco Prime Fulfillment Traffic Engineering Management. Cisco Prime Fulfillment Traffic Engineering Management has its own discovery process,. This process is documented in TE Network Discovery of the *Cisco Prime Fulfillment User Guide 6.2*.

However, if you are running *both* Cisco Prime Fulfillment Traffic Engineering Management (TEM) and Cisco IP solution Center MPLS VPN Management, you must run the Discovery process for Cisco Prime Fulfillment MPLS VPN Management.

Note the following points:

- One region (default region) is used for TEM.
- If you are also running Prime Fulfillment Discovery for MPLS VPN Management, make sure you run the Discovery workflow described in this chapter *first*, and then run the Cisco Prime Fulfillment Traffic Engineering Management process later.

Summary of Tasks for Discovery (Cisco Prime Fulfillment MPLS VPN Management and L2VPN Management)

Figure G-3 provides a general workflow diagram for the Discovery process used with the Cisco Prime Fulfillment MPLS VPN Management or Cisco Prime Fulfillment L2VPN Management application.

Note Figure G-5 provides a general workflow diagram for the Discovery process as used with the Prime Diagnostics application.

Figure G-3 Basic Workflow for Discovery with Cisco Prime Fulfillment MPLS VPN Management or Cisco Prime Fulfillment L2VPN Management

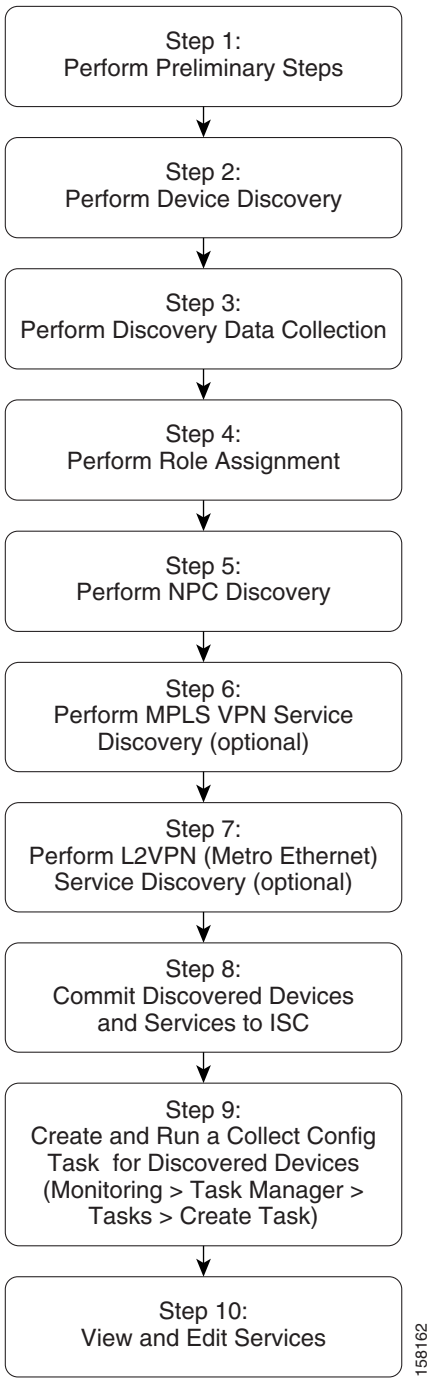


Table G-2 describes each task in the Discovery workflow for Cisco Prime Fulfillment MPLS VPN Management and Cisco Prime Fulfillment L2VPN Management.

Table G-2 Description of Discovery Steps for MPLS VPN and L2VPN Management

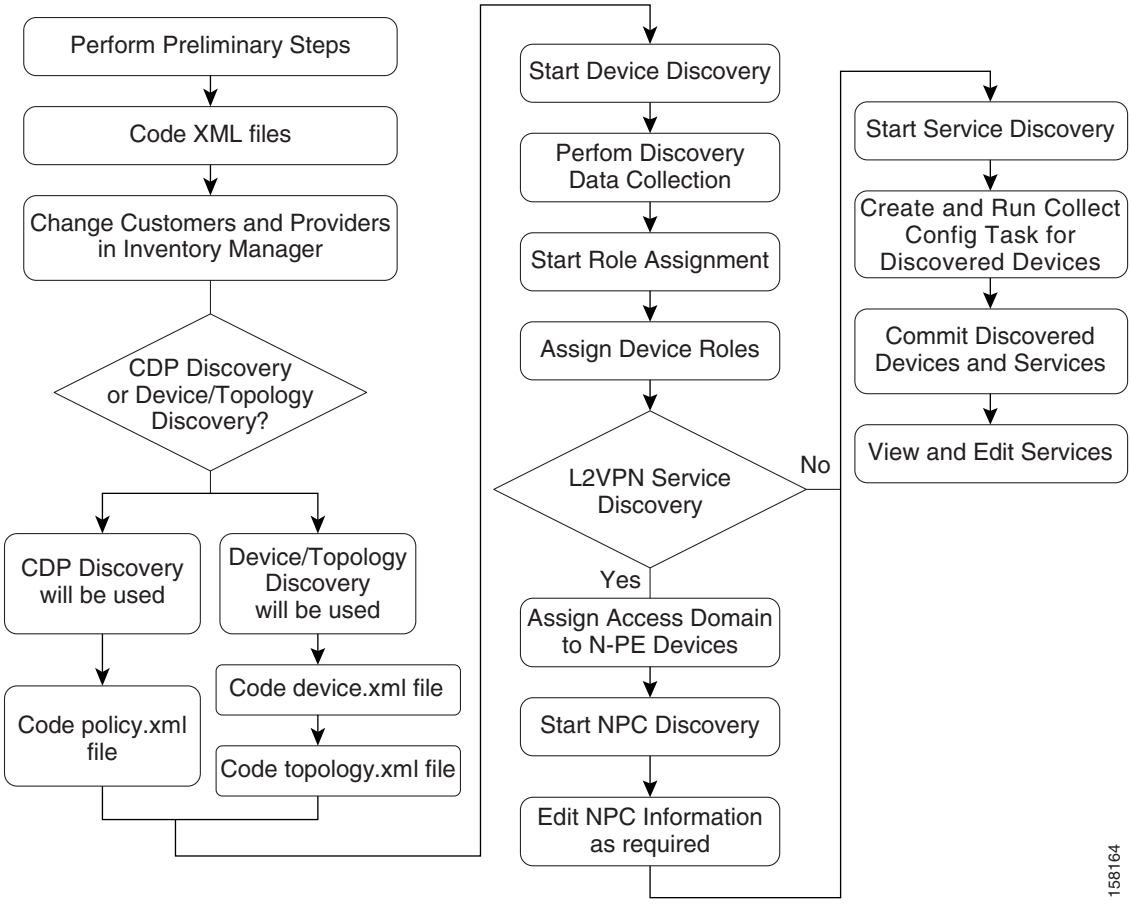
Step	Description
Step 1: Perform Preliminary Steps	Perform preliminary steps that are required for Prime Fulfillment Discovery. See Step 1: Perform Preliminary Steps, page G-16. • Review System Requirements See Review System Requirements, page G-17. • Install Licenses See Install Licenses, page G-18. • (CDP Discovery Only) Verify That a Unique TIBCO Port Is Defined See (CDP Discovery Only) Verify That a Unique TIBCO Port Is Defined, page G-18 • (CDP Discovery Only) Verify That CDP Is Running on Devices To Be Discovered See (CDP Discovery Only) Verify That CDP Is Running on Devices To Be Discovered, page G-19. • Code XML Files Required for Discovery See Code XML Files Required for Discovery, page G-20.
Step 2: Perform Device Discovery	• Start Device Discovery See Starting Device Discovery, page G-27. • After Device Discovery is complete, enter device passwords For information on entering device passwords, see Setting Password Attributes (Required Step), page G-31. • Enter additional device information as required See Setting General Device Attributes, page G-32 and Setting Cisco CNS Attributes, page G-32.
Step 3: Perform Discovery Data Collection	Start configuration collection. No input is required for this step. See Step 3: Perform Discovery Data Collection, page G-33.
Step 4: Perform Role Assignment	Assign device roles to each device. See Step 4: Perform Role Assignment, page G-33.
Step 5: Perform NPC Discovery	If you are discovering a Metro Ethernet Network with an Ethernet Core, perform the required preliminary steps. See Preliminary Steps Before Completing NPC Discovery for Metro Ethernet Networks, page G-41 • Start NPC Discovery See Step 5: Perform NPC Discovery, page G-41. • Modify and/or add NPCs as required. See Adding a Device for an NPC, page G-44, Adding a Ring, page G-45, Inserting a Device, page G-45, Inserting a Ring, page G-45, or Deleting a Device or a Ring, page G-46.

Table G-2 Description of Discovery Steps for MPLS VPN and L2VPN Management (continued)

Step	Description
Step 6: Perform MPLS VPN Service Discovery (optional)	Start MPLS VPN Service Discovery. See Step 6: Perform MPLS VPN Service Discovery (Optional), page G-46. This step is required for the Cisco Prime Fulfillment MPLS VPN Management application, **Note** This step is not required for the Cisco Prime Fulfillment L2VPN Management application or the Cisco Prime Fulfillment Prime Diagnostics application.
Step 7: Perform L2VPN Service Discovery (optional)	Start L2VPN Service Discovery. See Step 7: Perform L2VPN (Metro Ethernet) Service Discovery (Optional), page G-51. This step is required for the Cisco Prime Fulfillment L2VPN Management application. **Note** This step is not required for the Cisco Prime Fulfillment MPLS VPN Management application or the Cisco Prime Fulfillment Prime Diagnostics application.
Step 8: Commit Discovered Devices and Services to Prime Fulfillment Repository	Commit the discovered devices and services to the Prime Fulfillment repository. Prior to this step, discovery workflow stores the discovered devices and services in a temporary repository, which gets committed to Prime Fulfillment only at the last step of discovery workflow.
Step 9: Create and Run a Collect Config Task for Discovered Devices	From the Prime Fulfillment Start Page, choose **Operate > Tasks > Task Manager**. Select the **Collect Config** task and select all of the devices discovered in the Device Discovery step; then submit the task. See Step 9: Create and Run a Collect Config Task for the Discovered Devices, page G-58.
Step 10: View and Edit Services	The discovered services will be in Pending state and you need to do a config audit to move them to Deployed state. See Step 10: View and Edit Services, page G-59.

Within each step, additional tasks must be performed and choices must be made. Figure G-4 shows a detailed flowchart that illustrates all of the steps in the Discovery workflow.

Figure G-4 Detailed Diagram of Discovery Steps (Cisco Prime Fulfillment MPLS VPN Management and Cisco Prime Fulfillment L2VPN Management)



Summary of Prime Fulfillment Discovery Steps for Prime Diagnostics

Figure G-5 shows the basic Discovery steps for Cisco Prime Fulfillment with the Prime Diagnostics application. For Prime Diagnostics, several of the steps required for Cisco Prime Fulfillment MPLS VPN Management and Cisco Prime Fulfillment L2VPN Management are not required.

Figure G-5 Discovery Workflow for the Prime Diagnostics Application

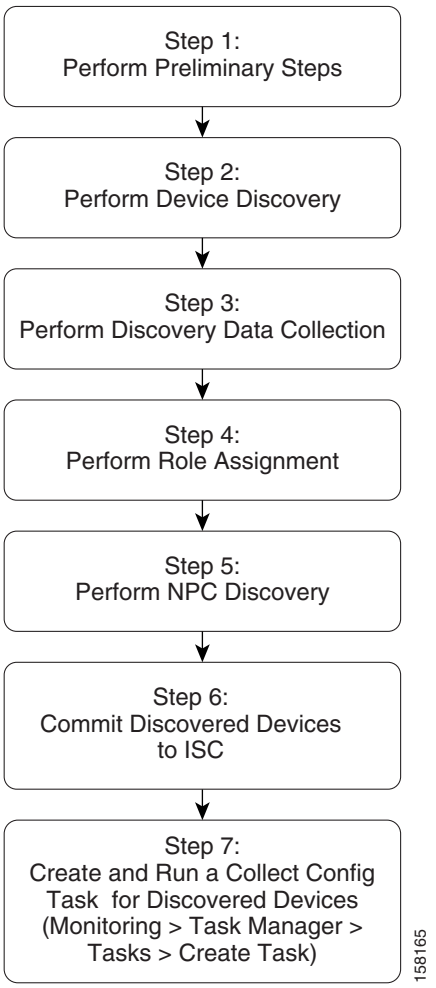


Table G-3 Description of Discovery Steps for Prime Diagnostics

Step	Description
Step 1: Perform Preliminary Steps	Perform preliminary steps that are required for Prime Fulfillment Discovery. • Review System Requirements See Review System Requirements, page G-17. • Install Licenses See Install Licenses, page G-18 • Code XML Files Required for Discovery For specific instructions, see the following section: – Code XML Files Required for Discovery, page G-20.
Step 2: Perform Device Discovery	• Start Device Discovery See Starting Device Discovery, page G-27. • After Device Discovery is complete, enter device passwords For information on entering device passwords, see Setting Password Attributes (Required Step), page G-31. • Enter additional device information as required See Setting General Device Attributes, page G-32 and Setting Cisco CNS Attributes, page G-32.
Step 3: Perform Discovery Data Collection	Start configuration collection. No input is required for this step. See Step 3: Perform Discovery Data Collection, page G-33.

Table G-3 Description of Discovery Steps for Prime Diagnostics (continued)

Step	Description
Step 4: Perform Role Assignment	Assign device roles to each device. See Step 4: Perform Role Assignment, page G-33. For Prime Diagnostics, you normally discover only P and PEs and assign P and PE roles to them However, if you discover CEs, assign CE roles to the CE devices. **Note** Although you do not have to edit NPCs for Prime Diagnostics, after you perform role assignment this step should complete.
Step 5: Create and Run a Collect Config Task for Discovered Devices	From the Prime Fulfillment Start Page, choose **Operate > Tasks > Task Manager**. Select the **Collect Config** task and select all of the devices discovered in the Device Discovery step; then submit the task. See Step 8: Commit Discovered Devices and Services to Prime Fulfillment Repository, page G-58.

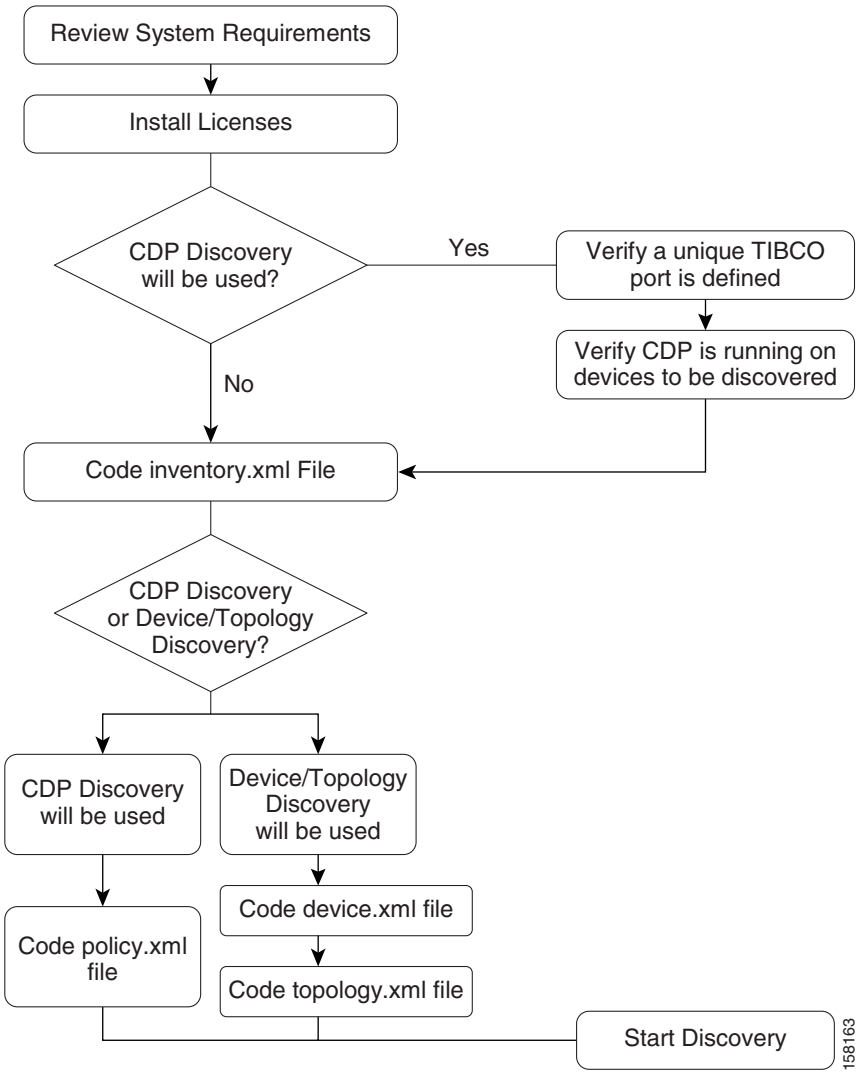
Step 1: Perform Preliminary Steps

Before you initiate the Prime Fulfillment Discovery process, complete the following preliminary steps:

- Review System Requirements
- Install Licenses
- Discovery in Large Networks
- (CDP Discovery Only) Verify That a Unique TIBCO Port Is Defined
- (CDP Discovery Only) Verify That CDP is Running on Devices To Be Discovered
- Code XML Files Required for Discovery

Figure G-6 summarizes the preliminary steps for Prime Fulfillment Discovery.

Figure G-6 Summary of Preliminary Steps for Discovery



Review System Requirements

Cisco recommends that you thoroughly review the system requirements for Prime Fulfillment before planning your installation, to be sure that you have all the hardware and software that you must successfully install.

The system recommendations and requirements for Prime Fulfillment are listed in Chapter 1, "System Recommendations" of the *Cisco Prime Fulfillment Installation Guide 6.2* and in the *Release Notes for Cisco Prime Fulfillment 6.2.*

Install Licenses

Before starting Discovery, the appropriate licenses (both Activation and VPN licenses) must be installed. Also, each license must be large enough to handle all possible discovered objects. For information on installing licenses, see the "Installing License Keys" section of Chapter 2 of the *Cisco Prime Fulfillment Installation Guide 6.2*, "Installing and Logging In to Prime Fulfillment."

Discovery in Large Networks

To discover large networks with a complex topology, we recommend you reset two DCPL properties, as follows:

Step 1 See Appendix B, "Property Settings" for an explanation of how to navigate to the Dynamic Component Properties Library (DCPL) properties.

Step 2 Navigate to the property **watchdog\server\discovery\heartbeat\timeout** and set this property to **180000 milliseconds** (3 minutes).

Step 3 Navigate to the property **watchdog\server\discovery\java\flags** and set this property to **-Xmx3072m -XX:PermSize=256m -XX:MaxPermSize=512m**

Step 4 Restart the Prime Fulfillment server.

Heap is a block of memory segment for the L2VPN and Metro Ethernet, Layer 3 MPLS VPN, and TEM components. It is allocated for use by the Java virtual machine (JVM) process during runtime. It might need to be increased for large deployments. If the **httpd** process restarts, increase the heap size, as follows:

Step 1 **cd $PRIMEF_HOME/bin**

Step 2 **vi tomcat.sh**

Step 3 Search for a line with **-Xmx** and specify a higher value.

Step 4 Set the heap size to 1GB or 2GB by replacing **-Xmx512m** with **-Xmx1024m** or **-Xmx2048m**, respectively.

Step 5 Save the **tomcat.sh** file.

Step 6 Enter **stopall** to stop the Prime Fulfillment server.

Step 7 Enter **startwd** to start the Prime Fulfillment server.

(CDP Discovery Only) Verify That a Unique TIBCO Port Is Defined

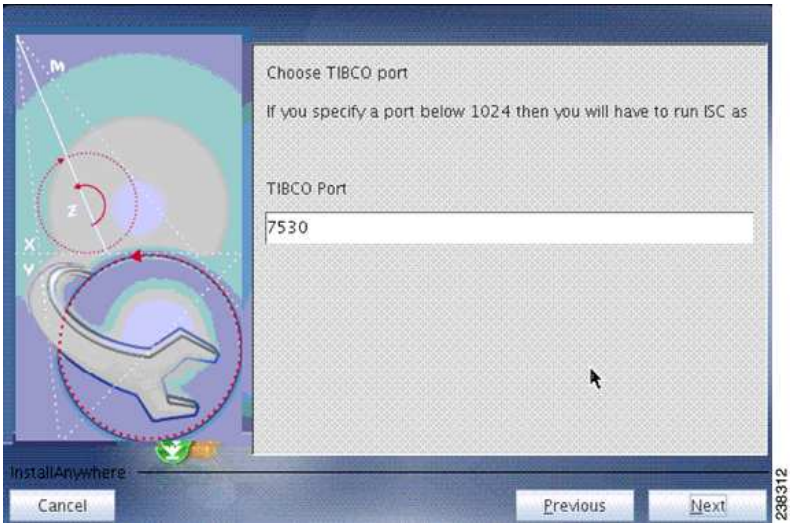
If you are using CDP Discovery to discover the network topology, make sure the TIBCO Port is unique. Otherwise, CDP discovery will fail.

During installation, the TIBCO port can be specified if the "custom" Installation Type is selected at the start of the installation process. Otherwise, the default port installed is 7530. You specify the TIBCO port on the Choose TIBCO Port dialog.

The port number that is specified must be unique throughout the network, and no other Prime Fulfillment installations are allowed with the same port.

Figure G-7 shows the Choose TIBCO Port dialog.

Figure G-7 Choose TIBCO Port



The Tibco port can be changed after installation by modifying the Dynamic Component Properties Library entry /SYSTEM/tibco/port, specified in Appendix B, "Property Settings".

(CDP Discovery Only) Verify That CDP Is Running on Devices To Be Discovered

If CDP Discovery is going to be used, use the **show cdp** command to ensure that CDP is running on all of the devices intended to be discovered.

For each device, enter the **show cdp** command, as shown in Example G-1.

Example G-1 The show cdp Command:

```
Router# show cdp
Global CDP information:
        Sending CDP packets every 120 seconds
        Sending a holdtime value of 180 seconds
        Sending CDPv2 advertisements is enabled
Router#
```

Note When performing CDP Discovery for devices with more than one IP address configured, it is possible that CDP discovery will find an IP address other than the management IP address. If the IP address found is not accessible from the Prime Fulfillment server, then it will not be possible to discover that device using CDP discovery.

Code XML Files Required for Discovery

Before you can run Prime Fulfillment Discovery, you must code XML files that are required for the Discovery process. A different set of files is required, depending on whether you use CDP Discovery or Device/Topology-based Discovery.

Table G-4 describes the XML files and indicates which files are required for each type of discovery method.

Table G-4 XML Files Used with Prime Fulfillment Discovery

XML File	Description	Required for CDP Discovery	Required for Device/Topology Based Discovery
policy.xml	Specifies one or more seed IP addresses that can be reached from the specified seed device and a maximum hop count for the device discovery process.	Yes	No
device.xml	Specifies information used to locate devices, such as device IP addresses and Object IDs (OIDs).	No	Yes
topology.xml	Specifies information used to build NPCs used by MPLS VPN and/or Metro Ethernet topology.	No	Yes

Note Make sure that the coding in your XML files is accurate. If there are errors in the files, you might need to re-run the Discovery process.

Sample XML Files

The initial installation of Prime Fulfillment provides sample XML files that you can use as a starting point in coding your own XML files. The sample XML files are located in the following directory:

<*install_directory***>/resources/discovery/sample**

where <*install_directory*> is the installation directory that you specified when prompted by the Prime Fulfillment installation program.

Coding the policy.xml File

The **policy.xml** file:

- Is required for CDP Discovery.
- Is required for Cisco Prime Fulfillment MPLS VPN Management, Cisco Prime Fulfillment Carrier Ethernet and L2VPN Management, and Prime Diagnostics.
- Is not required for Device/Topology-based Discovery.
- Is not required for Cisco Prime Fulfillment Traffic Engineering Management.
- Provides a seed IP address that the CDP protocol uses to discover devices near the seed device.

Example G-2 shows the sample **policy.xml** file that is provided with the Prime Fulfillment installation.

Example G-2 Sample policy.xml File

```
<?xml version='1.0' encoding='UTF-8'?>
<DISCOVERY_POLICY overwrite_existing_policy="true">
  <DISCOVERY_METHOD>
    <CDP ipaddress="209.168.133.232" hop="1"/>
  </DISCOVERY_METHOD>
  <SNMP_COMMUNITY>
    <RO_COMMUNITY>
      <COMMUNITY community="public"/>
    </RO_COMMUNITY>
    <RW_COMMUNITY>
      <COMMUNITY community="private"/>
    </RW_COMMUNITY>
  </SNMP_COMMUNITY>

</DISCOVERY_POLICY>
```

If there are additional routers that are on the other side of PE routers on the edge of the core segment of the network, you can specify more than one seed IP address in order to discover these devices.

Example G-3 shows a **policy.xml** file that contains two seed IP addresses.

Example G-3 Policy.xml File with Two IP Addresses

```
<?xml version='1.0' encoding='UTF-8'?>
<DISCOVERY_POLICY overwrite_existing_policy="true">
  <DISCOVERY_METHOD>
    <CDP ipaddress="209.168.133.241" hop="8"/>
  </DISCOVERY_METHOD>
  <DISCOVERY_METHOD>
    <CDP ipaddress="209.168.133.244" hop="8"/>
  </DISCOVERY_METHOD>
  <SNMP_COMMUNITY>
    <RO_COMMUNITY>
      <COMMUNITY community="public"/>
    </RO_COMMUNITY>
    <RW_COMMUNITY>
      <COMMUNITY community="private"/>
    </RW_COMMUNITY>
  </SNMP_COMMUNITY>

</DISCOVERY_POLICY>
```

Table G-5 describes the XML tags used in the **policy.xml** file.

Table G-5 XML Tags and Attributes Used in the policy.xml File

Tag	Description
<DISCOVERY_METHOD>	Starts a **<DISCOVERY_METHOD>** tag. The **<DISCOVERY_METHOD>** tag must contain a **<CDP>** tag.
<CDP>	Starts a **<CDP>** tag. The **<CDP>** tag specifies a seed IP address and a hop count. The **<CDP>** tag must contain the following attributes: • **ipaddress** • **hop**

Table G-5 XML Tags and Attributes Used in the policy.xml File (continued)

Tag	Description
ipaddress	Specifies the IP address of a seed device. Required attribute for the **<CDP>** tag.
hop	Specifies the number of hops from the device identified by the ipaddress attribute to go in discovering devices. Required attribute for the **<CDP>** tag.

Follow these steps to edit the sample **policy.xml** file:

Step 1 Edit the sample file and replace the IP address specified with the **ipaddress** XML attribute with an appropriate IP address from your network.

This IP address is a device that can be reached from the Prime Fulfillment host. For each seed device, an accessible interface on the starting point is configured, because the management interface must be provided. The management interface is the address on the device that the Prime Fulfillment host uses to reach the device.

Note You can provide more than one IP address. This is useful in situations where one network domain is on the other side of a PE router on the edge of the core segment of the network.

Step 2 Edit the hop count specified with the **hop** attribute and specify a hop count that will be used when the Discovery process is initialized.

When you choose the seed devices and hop count, pick a seed device that can reach a large section of the network. Pick one or more of them until you think these devices will enable you to reach your entire managed network.

Point-of-presence (POP) routers are usually good choices. If you choose all the POPs in your network as the collection of seed devices and put in the appropriate number of hubs, you discover the entire managed network.

To pick the hop count number, go to the CE that is the furthest from its associated POP, and count the number of devices between them. If this number is N, the hub number is N+1, assuming you are picking the POP as the seed.

Step 3 If you need to add additional IP addresses for seed devices, code additional **<DISCOVERY_METHOD>** tags.

Within the additional **<DISCOVERY_METHOD>** tags, include **<CDP>** tags.

For each **<CDP>** tag, specify an IP address with the **ipaddress** attribute and a hop count with the **hops** attributes.

Step 4 Save the **policy.xml** file to an appropriate directory on the Prime Fulfillment host.

When you run the Discovery process, the process queries the starting point device for its CDP table. From this table, all of those devices are queried for their CDP information. This process continues until the maximum hop count from the starting point is reached. When you use the CDP-based method, note that only devices running CDP are discovered.

Coding the device.xml File

The **device.xml** file:

- Is required for Device/Topology-based Discovery.
- Is not required for CDP-based Discovery.
- Is required for Cisco Prime Fulfillment MPLS VPN Management, Cisco Prime Fulfillment Carrier Ethernet and L2VPN Management, and Prime Diagnostics.
- Is not required for Cisco Prime Fulfillment Traffic Engineering Management.
- Specifies information used to locate devices, such as device IP addresses and Object IDs (OIDs).

Example G-4 shows a sample **device.xml** file. Use the sample file as an example and save your edited file in an appropriate directory.

Example G-4 Sample device.xml file

```
<network>
<device>
<device-name>mlpe8</device-name>
<ip-address>209.168.133.244</ip-address>
<system-object-id>.1.3.6.1.4.1.9.1.509</system-object-id>
<snmp-info>
<ro-community>public</ro-community>
</snmp-info>
</device>

<device>
<device-name>mlsw11</device-name>
<ip-address>209.168.133.170</ip-address>
<system-object-id>.1.3.6.1.4.1.9.1.574</system-object-id>
<snmp-info>
<ro-community>public</ro-community>
</snmp-info>
</device>

<device>
<device-name>mlsw16</device-name>
<ip-address>209.168.133.175</ip-address>
<system-object-id>.1.3.6.1.4.1.9.1.574</system-object-id>
<snmp-info>
<ro-community>public</ro-community>
</snmp-info>
</device>

<device>
<device-name>mlsw17</device-name>
<ip-address>209.168.133.176</ip-address>
<system-object-id>.1.3.6.1.4.1.9.1.574</system-object-id>
<snmp-info>
<ro-community>public</ro-community>
</snmp-info>
</device>

</network>
```

Table G-6 describes the XML tags used in the **device.xml** file.

Table G-6 XML Tags Used in the device.xml File

Tag	Description
<device>	Starts a **<device>** tag. The **<device>** tag must contain the following tags: • **<device-name>** • **<ip-address>** The following tags are optional within the **<device>** tag: • **<system-object-id>** • **<snmp-info>**
<device-name>	Specifies the name of the device. Required within the **<device>** tag.
<ip-address>	Specifies the IP address of the device. Required within the **<device>** tag.
<system-object-id>	(optional) Can be included to specify the SNMP Object ID (OID) for the device. If this is provided, it is specified within the **<device>** tag.
<snmp-info>	Specifies SNMP information for the device. The **<snmp-info>** tag must contain a **<ro-community>** tag. Optional within the **<device>** tag.
<ro-community>	Specifies the level of SNMP access for the device. Normally, this should be "public." Required within the **<snmp-info>** tag.
Note: SNMPv3 is not supported.	

Follow these steps to code the **device.xml** file:

Step 1 Edit the sample **device.xml** file provided with the installation.

Step 2 For each device that is to be discovered by Prime Fulfillment, code a **<device>** entry.

Each **<device>** entry should contain the following tags:

- A **<device-name>** tag specifying the device name.
- An **<ip-address>** tag specifying the IP address for the device.
- A **<system-object-id>** tag specifying the OID for the device (optional).
- An **<snmp-info>** tag specifying **<ro-community>** information

Step 3 Save the **device.xml** file to an appropriate directory on the Prime Fulfillment host.

Coding the topology.xml File

The **topology.xml** file:

- Is required for Device/Topology-based Discovery.
- Is not required for CDP-based Discovery.
- Is required to perform Prime Fulfillment Discovery for Cisco Prime Fulfillment MPLS VPN Management, Cisco Prime Fulfillment Carrier Ethernet and L2VPN Management, and Prime Diagnostics.
- Is not required for Cisco Prime Fulfillment Traffic Engineering Management.
- Specifies information used to locate devices, such as device IP addresses and Object IDs (OIDs).

The **topology.xml** file specifies the discovery protocol that is used in the discovery process, and, for each connection, specifies the starting IP address, the starting interface, the end device, and the end interface

Example G-5 shows a sample **topology.xml** file. Use the sample file as an example and save your edited file in an appropriate directory.

Example G-5 Sample topology.xml File

```
<topology>
<connection discovery-protocol="CDP" fromDevice="mlsw19"  fromIP="209.168.133.178"
fromInterface="GigabitEthernet1/1/2"   toDevice="mlsw21"  toIP="209.168.133.220"
toIF="GigabitEthernet1/1/1" >
</connection>

<connection discovery-protocol="CDP" fromDevice="mlsw19"  fromIP="209.168.133.178"
fromInterface="FastEthernet1/0/23"   toDevice="mlsw21"  toIP="209.168.133.220"
toIF="FastEthernet1/0/24" >
</connection>

<connection discovery-protocol="CDP" fromDevice="mlsw19"  fromIP="209.168.133.178"
fromInterface="FastEthernet
1/0/24"   toDevice="mlsw18"  toIP="209.168.133.177"  toIF="FastEthernet1/0/23" >
</connection>

<connection discovery-protocol="CDP" fromDevice="mlsw19"  fromIP="209.168.133.178"
fromInterface="FastEthernet1/0/22"   toDevice="mlsw22"  toIP="209.168.133.221"
toIF="FastEthernet1/0/24" >
</connection>

</topology>
```

Table G-7 describes the XML tags used in the **topology.xml** file.

Table G-7 XML tags and Attributes Used in the topology.xml File

Tag	Description
<connection>	Starts a **<connection>** tag. The **<connection>** tag must specify the following attributes: • **discovery-protocol** • **fromDevice** • **FromIP** • **FromInterface** • **toDevice** • **toIP** • **toIF**
discovery-protocol	Specifies the Discovery protocol used to discover the network topology. Normally, this is "CDP."
fromDevice	Specifies the name of the device from which the Named Physical Circuit starts. Required attribute for the **<connection>** tag.
FromIP	Specifies the management IP address of the device from which the Named Physical Circuit starts. Required attribute for the **<connection>** tag.
FromInterface	Specifies the name of the device interface from which the Named Physical Circuit starts. Required attribute for the **<connection>** tag.
toDevice	Specifies the name of the device to which the Named Physical Circuit connects. Required attribute for the **<connection>** tag.
toIP	Specifies the management IP address of the device from which the Named Physical Circuit connects. Required attribute for the **<connection>** tag.
toIF	Specifies the device interface on the device to which the Named Physical Circuit connects. Required attribute for the **<connection>** tag.

Follow these steps to code the **topology.xml** file:

Step 1 Edit the sample **topology.xml** file provided with the installation.

Step 2 For each NPC connection that is to be discovered by Prime Fulfillment, code a **<connection >** entry.

Each **<connection>** entry must contain the following tags:

- A **discovery-protocol** attribute specifying the CDP protocol.
- A **fromDevice** attribute specifying the device from which the NPC starts.
- A **FromIP** attribute specifying the management IP address from which the NPC starts.

- A **FromInterface** attribute specifying the device interface from which the NPC starts.
- A **toDevice** attribute specifying the name of the device to which the NPC connects.
- A **toIP** attribute specifying the management IP address of the device to which the NPC connects
- A **toIF** attribute specifying the name of the interface on the device to which the NPC connects

Step 3 Save the **topology.xml** file to an appropriate directory on the Prime Fulfillment host.

Step 2: Perform Device Discovery

This section describes how to start the device discovery process and edit device configuration.

Starting Device Discovery

To start discovery, follow these steps:

Step 1 Log into Prime Fulfillment.

Step 2 Click the **Inventory > Physical Inventory > Discovery**.

The Device Discovery — CDP Fields window appears.

Initially, the CDP Discovery method is selected and the window displays the required input for this method.

The editable **Output Device File** field is optional and defaults to an XML file of the discovered devices. This file can then be an input **Devices File** for rerunning discovery using the **Device/Topology** option, by choosing that radio button.

The editable **Output Connection File** is optional and defaults to an XML file that contains device connectivity information that is written during CDP Device Discovery. This file can then be an input **NPC Topology File** for rerunning discovery using the Device/Topology option, by choosing that radio button.

Step 3 Choose a Discovery method:

- To use the Cisco Discovery Protocol (CDP) method, click the **CDP** radio button.
- To use the Device/Topology method, click the **Device/Topology** button.
- To use the Import Configuration Files method, click the **Import Configuration Files** button.

The required **Directory** field is the directory on the server that contains configuration files for the devices to be discovered. The format of these files *must* be <*filename*>**.cfg**.

The **NPC Topology File** field contains an XML file that contains device connectivity information that is used to automatically create NPCs.

Note During service discovery, Providers, Regions, Customers, and Sites are not automatically created, and therefore you must manually create them before running service discovery. If Resource Pools are used for provisioning in Prime Fulfillment, Access Domains and Resource Pools must be manually created before running service discovery.

Step 4 In the Discovery window, specify the settings indicated in Table G-8.

Table G-8 Discovery Settings

Setting	Description
Name	In this field, enter a unique name of your choice for the Workflow name. If you do not enter a name in this field, the system automatically generates a unique name for you.
CDP	Click this radio button to select Cisco Discovery Protocol (CDP) as the Discovery method.
Policy File	If you click the **CDP** button, specify the path to your **policy.xml** file here. This file is an XML file that indicates the IP address of one or more devices used as a starting point for the discovery process. For more information on the **policy.xml** file, see Coding the policy.xml File, page G-20.
Output Device File	This editable optional field defaults to an XML file of the discovered devices. This file can then be an input **Devices File** for rerunning discovery using the **Device/Topology** option.
Output Connection File	This editable optional field defaults to an XML file that contains device connectivity information that is written during CDP device discovery. This file can then be an input **NPC Topology File** for rerunning discovery using the **Device/Topology** option.
Device/Topology	Click this radio button to select Device/Topology as the Discovery method.
Devices File	If you click the **Device/Topology** button, specify the path to your **device.xml** file here. This file contains information used to locate the devices in your network, such as IP addresses and OIDs. For more information on the **device.xml** file, see Coding the device.xml File, page G-23.
NPC Topology File	If you click this optional **Device/Topology** button, specify the path to your **topology.xml** file here. This file contains information used to determine the NPC topology of your network. For more information on the **topology.xml** file, see Coding the topology.xml File, page G-25.
Import Configuration Files	Click this radio button to select Import Configuration Files as the Discovery method.
Directory	This required field is the directory on the server that contains configuration files for the devices to be discovered. The format of these files *must* be <*filename*>**.cfg**.

Table G-8 Discovery Settings (continued)

Setting	Description
NPC Topology File	This field contains an XML file that contains device connectivity information that is used to automatically create NPCs.
MPLS VPN	To discover devices used in an MPLS VPN service, click the **MPLS VPN** radio button.
L2VPN (Metro Ethernet) Discovery	To discover layer 2 devices used in a Metro Ethernet service, click the **L2VPN (Metro Ethernet) Discovery** radio button.

Step 5 Click the **Start** button.

The discovery process starts and the Discovery Workflow window appears.

The **Workflow** category in the data pane gives the name information about the current discovery request/workflow.

Click the **Restart** button and you receive a drop-down list of completed steps. Select a step and you will restart from that step.

In the left column, **Current Request** gives the discovery request/workflow that is currently running. If there is no currently running discovery request/workflow, an initialization window appears to create a new discovery request/workflow.

In the left column, **Previous Requests** lists all the discovered requests/workflows. You can look at the status and logs for any of these discovery requests/workflows.

Discovery Workflow window indicates the progress of each phase of device discovery:

- When the window first appears, the status indicator is yellow and indicates that the device discovery process is **Initializing**.
- The status indicator then indicates that the process is **In Progress**.
- After the discovery processes has completed, the display indicates how many devices were discovered, and the status indicator changes to orange and indicates that there is **Pending Input.**

The Progress area at the bottom of the window indicates how many devices were discovered.

At the lower right of the window there is a **Restart** button. You can click this button to restart the entire discovery process. However, if you restart the Discovery process, any work that has been done previous to restarting Discovery is lost.

Note After each phase of the Discovery process, make sure that you check the log file to ensure that there were no errors in the process. For specific instructions, see Using the Discovery Log Files, page G-7.

Editing Device Configurations

After the initial discovery of devices in your network, you must edit the information that Prime Fulfillment maintains about the devices. This allows the Discovery process to collect configuration information about the devices that are required to determine the network topology and generate service requests.

Editing device configuration includes these steps:

- Setting Password Attributes (a required step)
- Setting General Device Attributes
- Setting Cisco CNS Attributes

Follow these steps to edit device configurations:

Step 1 When the Discovery Workflow window indicates that the Device Discovery is **Pending Input,** click the **Continue** button.

The General Attributes - Devices window appears.

The General Attributes - Devices window allows you to do the following:

1. Delete devices.

   If devices appear in the device list that you do not want to configure, you can delete them, as explained in Step 4.

2. Set the following groups of attributes for each device:

- **General Attributes**—The general attributes include the hostname of the device, the device type, the management IP address, and other settings.

  You can accept the default attributes shown in the General Attributes - Devices window or change them as required.

  For a list of the general attributes, see Setting General Device Attributes, page G-32.

- **Password Attributes**—The password attributes include the username and password for the device and the enable username and password for the device. You *must* set these attributes.
- **CNS Attributes**—If the device is a CNS device, set the CNS attributes.

Step 2 If you want to filter the devices that appear in the window, enter part of the device name for the devices that you want to view, preceded or followed by the asterisk (*) and then click the **Find** button.

If the Find field displays an asterisk, all devices are displayed.

The setting in the Find field applies to all of the attributes windows.

Step 3 To change the display to show one of the attributes areas, click the **Attributes** button at the bottom of the window and use the pull-down list to select the attributes area to display.

- If you need to change the general attributes for the device, such as the protocol used to configure the device (Config Access Protocol), you can do this in the initial window that appears.

  If the General Attributes - Devices window is not the current window, click the **Attributes** button and select **General Attributes** from the pull-down list.

  See Setting Password Attributes (Required Step), page G-31 for instructions on setting the General Attributes.

- To set the password attributes, click the **Attributes** button and then select Password Attributes from the pull-down list.

For instructions on setting the password attributes, see Setting Password Attributes (Required Step), page G-31.

Note This is a required step. To enable configuration collection, you *must* set the password attributes.

- If you need to change the CNS attributes, see Setting Cisco CNS Attributes, page G-32.

Step 4 If you want to delete one or more devices, follow these steps:

a. Check the check box next to each device that you want to delete.

   If you need to delete more than one device, you can check the check box next to the heading for the list of the devices. This selects all of the devices in the list. You can then uncheck the boxes next to any devices that you do not want to delete.

b. To delete the devices, click the **Delete** button.

Setting Password Attributes (Required Step)

In order for the Configuration Collection phase to succeed, you *must* set the password attributes for each device. Follow these steps to set password attributes:

Step 1 If the Password Attributes window is not the current window, click the **Attributes** button and select **Password Attributes** from the pull-down list.

The Password Attributes window appears.

Step 2 Follow these steps to select the devices and password attributes to configure:

a. Check the check box next to a device that has password attributes you want to configure.

   If several devices have the same password attributes, you can check multiple check boxes. If all of the devices have the same password attributes, you can check the box to the left of the heading row to select all of the devices in the list. If this check box is checked, you can uncheck it to deselect all of the devices.

b. To select the password attributes to configure, check one or more of the check boxes next to the attribute names in the heading row.

Step 3 Click the **Edit** button.

Step 4 Enter the following information for the device:

- **Login Password**—Enter the login password for the device
- **Login User**—Enter the username for the device
- **Enable User**—Enter the name of a user with enable privileges
- **Enable Password**— Enter the enable password for the enable user

Step 5 Click **Save**.

The information that you entered appears in the Password Attributes window.

Setting General Device Attributes

After you complete the device discovery process, the General Attributes - Devices window displays the current general attributes settings for each device.

Follow these steps to change the general attributes for a device:

Step 1 Click on the attribute that you want to change.

An Edit Attributes dialog box appears for the selected attribute.

Step 2 In the dialog box, indicate the new setting for the attribute.

The General Device attributes include the following:

- **Host Name**—Must begin with a letter, digit, or underscore followed by letters, digits, underscores, spaces, hyphens, or dots ending with a letter, digit, or underscore. This field is required and must match the name configured on the target router device. Limited to 256 characters.
- **Device Type**—The device type is the Cisco Router.
- **Device Description (not editable from this window)**—Can contain any pertinent information about the device, such as the type of device, its location, or other information that might be helpful to service provider operators. Limited to 80 characters.
- **Management Address**—Valid IP address of the device that Prime Fulfillment uses to configure the target router device. This IP address must be reachable from the Prime Fulfillment host.
- **Domain Name**—Must begin with a letter, digit, or underscore followed by letters, digits, underscores, spaces, hyphens, or dots ending with a letter, digit, or underscore. The name must match the domain name on the target router device.
- **Config Access Protocol**—Administers the access protocol for config upload and download. Choices include: Telnet, Terminal, TFTP, and RCP.

Setting Cisco CNS Attributes

If one of the devices is a Cisco CNS device, follow these steps to set CNS attributes:

Step 1 If the CNS Attributes window is not the current window, click the **Attributes** button and select **CNS Attributes** from the pull-down list.

The CNS Attributes window appears.

The **Terminal Server** column specifies the devices that represent the workstations that can be used to provision edge routers, and the **Port Number** column specifies the port numbers used by the terminal server.

Step 2 Click an existing Event Identification item.

The Edit Attributes dialog box for Event Identification appears.

Step 3 From the drop-down list for Event Identification attribute, you can select **Event-Identification**, which indicates whether the CNS Identification field contains a HOST NAME or CNS ID. Default: HOST NAME.

Saving the Device Configuration

After you are finished making device configuration changes, click the **Continue** button.

The Device Discovery indicator turns green and indicates that Device Discovery is **Complete**.

The Discovery Data Collection phase begins automatically.

Step 3: Perform Discovery Data Collection

After you save your device configuration settings, the Discovery Data Collection phase of Device Discovery starts automatically.

While Cisco Prime Fulfillment is collecting the device configurations, the Discovery Data Collection indicator is yellow and indicates that the process is **In Progress**.

When the Discovery Data Collection phase is complete, the indicator changes to green and indicates that the process is **Complete**. You are now ready to assign device roles.

Step 4: Perform Role Assignment

After the Discovery Data Collection phase of Device Discovery is complete, the Discovery Workflow window indicates that the Role Assignment phase is **Pending Input**.

Restarting from Discovery Data Collection prompts you to select the devices for which discovery data collection needs to occur.

Follow these steps to assign device roles:

- Initiate Device Role Assignment
- Change the Device Assignment Display
- Change Device Assignments
- Determine Device Roles
- Assign CE Device Roles
- Assign PE Device Roles

The following sections describe each of these steps.

Initiating Device Role Assignment

Follow these steps to initiate device role assignment:

Step 1 In the Discovery Workflow window, click **Continue**.

The Role Assignment - Un-assigned Devices window appears.

On the Role Assignment - Un-assigned Devices window, if you select a single device, you are prompted directly for the device role assignment. However, if you select more than one device, either the Role Assignment - CEs window or the Role Assignment - PEs window appears. On these windows you can specify the desired device roles.

Step 2 If you want to change the way that the devices are displayed, see the following section, Changing the Device Assignment Display, page G-34.

Changing the Device Assignment Display

You can change the way devices are displayed in the Role Assignment window in the following ways:

- You can change the display to show unassigned devices, PE devices, or CE devices using the pull-down list at the bottom of the Role Assignment window.
- You can change the range of devices that are displayed using the **Show devices with** selection at the top of the window in combination with the **matching** field.

Follow these steps to change the category of devices that is displayed:

Step 1 To change the category of devices that is displayed, select a value from the pull-down list at the bottom of the Role Assignment window:

- To view PE devices, select **PEs**.
- To view CE devices, select **CEs**.
- To view unassigned devices, select **Un-assigned Devices**.

Step 2 To change the range of devices that are displayed, use the **Show devices with** selection at the top of the window in combination with the **matching** field.

- To list devices by hostname, select **Device Host Name** and enter a search value in the matching field, then click **Find**.
- To list devices by domain name, select **Device Domain Name** name and enter a search value in the matching field, then click **Find**.
- To list devices by management IP address, select Management IP Address and enter a search value in the matching field, then click **Find**.

The value in the **matching** field specifies a search mask that controls which devices are displayed. An asterisk (*) specifies display of all devices by the selected search criteria. A string followed by an asterisk specifies display of all devices starting with part of a hostname, domain name, or management IP address. And a string preceded by an asterisk specifies display of all devices ending with part of a hostname, domain name, or management IP address.

You can specify more than one wildcard (asterisk) value in a search string. For example, to display all devices that have “ce” in the hostname, enter *ce* in the matching field.

The display changes depending on the selection that you made. For example, if two devices have been assigned the CE role, the Role Assignment - CEs window appears.

Changing Device Assignments

In some instances, the device discovery process assigns the wrong device role to groups of devices. For example, devices that should be PEs can be assigned as CEs.

If this occurs, perform these steps:

- If all the devices you expected would appear as PEs are not listed on the Role Assignment - PEs window, check the Role Assignment - Unassigned Devices window and the Role Assignment - CEs window and assign the devices as PE devices.
  - Go to the Role Assignment - CEs window and select any devices that should be PE devices
  - Click the **Assign as PEs** button

    The Role Assignment - PEs window appears and now lists the devices that you assigned as PEs.
- If other devices are not assigned as desired, change their basic device assignment as required.

Assigning Devices Individually or in Bulk

Using the windows provided for Role Assignment, you can assign device roles one device at a time or using bulk assignment (by selecting several devices and assigning them all the same role).

If you assign device roles for a single device, you can also assign the other device attributes, such as Site, Region, etc. However, if you assign device roles in bulk, then you cannot assign the other attributes at this time. You will have to go to the PEs or CEs window later to assign the other attributes.

Determine Device Roles

The purpose of device assignment is to categorize the devices discovered in the provider's network into two general groups:

- Provider-related devices—Provider Edge (PE) devices.

  See Assigning the PE Role, page G-35 for instructions on assigning the PE roles (U-PE, N-PE, P, or PE-AGG).
- Customer-related devices—Customer Edge (CE) devices

  See Assigning the CE Role, page G-38 for instructions on assigning the CE role.

For PE devices, use the following guidelines to determine device roles:

- Assign a device that is at the center of a core domain as a P device.
- Assign any devices that interface with users of the VPN services as U-PE devices. These are devices that are on the customer facing edge of a domain.
- Assign any devices that are on the edge of the MPLS core domain or L2VPN core domain as N-PE devices.
- Assign any devices that are in device rings or which connect to multiple U-PE devices as PE-AGG devices.

  For CE devices, see the descriptions of the CE roles in the section on assigning CE roles (Assigning the CE Role, page G-38) for specific information.

Assigning the PE Role

Follow these steps to assign a device as a PE device:

Step 1 In the Role Assignment - Un-assigned Devices window, select a device that you want to assign as a PE.

- To select a device, check the check box next to the device name.
- To deselect a device, uncheck the check box next to the device name.

Step 2 Click the **Assign as PE(s)** button.

Step 3 In the Assign as PE window, assign the required information for the PE.

a. To assign a PE Region Name, click the **Select** button.

The PE Region Name window appears.

b. In the PE Region Name window, click the radio button next to the region name that you want to assign and then click **Select**.

The Assign as PE window appears with the region name in the PE Region field.

c. To assign a PE role, select a value from the pull-down list for the PE Role field.

The PE role specifies the architectural role that a PE router performs. Assign the PE role based on the network layer to which the device belongs.

You can select the following PE roles:

- **N-PE**—Assign devices that are at the edge of domains (within the Edge layer) as Network Facing Provider Edge (N-PE) devices.
- **U-PE**—Assign devices within the User Facing Provider Edge as U-PE devices.
- **P**—Assign a device that is at the center of a core domain as a Provider Core (P) device.
- **PE-AGG**—Assign devices within the Aggregation Layer as Provider Edge Aggregation (PE-AGG) devices.

d. Click **OK**.

The Role Assignment - PEs window appears with the specified values shown.

Editing the PE Role

After you have assigned one or more devices as PE devices and they appear in the Role Assignment - PEs window, you can edit the PE role. You can edit the PE role even if no values have been assigned in the Assign as PE window.

Note PE role assignment is not mandatory. However, it is recommended to avoid unexpected behavior.

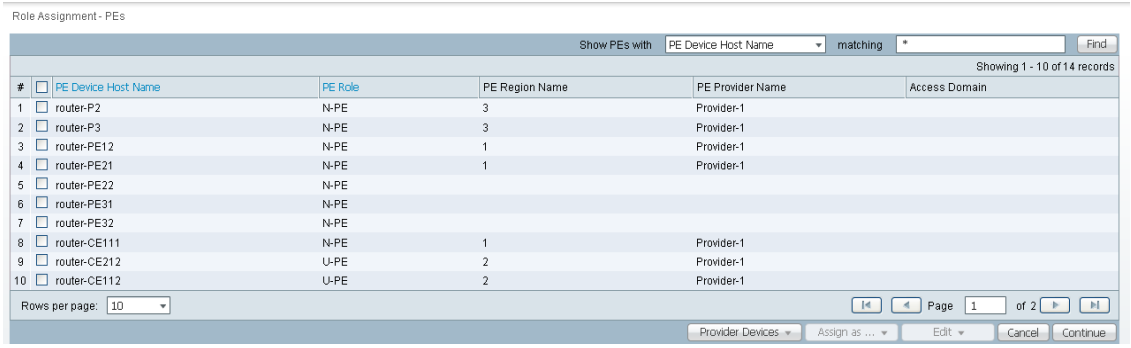
Follow these steps to edit the Role Assignment values for a PE device:

Step 1 While the Role Assignment phase of Device Discovery is active, choose the Role Assignment - PEs window.

If the Role Assignment - Un-assigned Devices or the Role Assignment - CEs window is active, select **Role-Assignment - PEs** from the pull-down list at the bottom of the window.

The Role Assignment - PEs window appears, as shown in Figure G-8.

Figure G-8 Role Assignment - PEs Window



Note that on this window, sorting is disabled for the following columns:

- PE Device Host Name
- PE Provider Name
- PE Region Name.

In the sample window shown in Figure G-8, one of the PEs has role information assigned. The other two PEs have been assigned as PEs but do not have role information assigned. You can edit any of the information for the PEs, whether information has been entered or not.

Step 2 Select one or more PEs to edit.

- To select a specific PE, check the check box next to the device name.
- To select all the PEs shown in the window, check the check box in the heading row.

Step 3 To edit the PE role, follow these steps:

a. Click the **Edit** button at the bottom of the window and choose **PE Role** from the pull-down list.

You are prompted to select a PE role.

b. Select a value from the pull-down list for the PE Role field.

You can select the following PE roles:

- **N-PE**—Assign devices within the Edge layer as Network Facing Provider Edge (N-PE) devices.
- **U-PE**—Assign devices within the User Facing Provider Edge as U-PE devices.
- **P**—Assign devices within the Core layer as Provider Core (P) devices.
- **PE-AGG**—Assign devices within the Aggregation Layer as Provider Edge Aggregation (PE-AGG) devices.

The specified PE role appears in the Role Assignment - PEs window.

Step 4 To edit the PE provider name or PE region name, follow these steps:

a. Click the **Edit** button at the bottom of the window and choose **Region/Provider** from the pull-down list.

You are prompted for a Region name.

b. Click the radio button next to one of the region names listed in the pop-up window and then click the **Select** button.

The specified Region Name and its associated Provider Name appear in the Role Assignment - PEs window.

Assigning the CE Role

Follow these steps to assign a device as a CE device:

Step 1 In the Role Assignment - Un-assigned Devices window, select a device that you want to assign as a CE.

- To select a device, check the check box next to the device name.
- To deselect a device, uncheck the check box next to the device name.

Step 2 Click the **Assign as CE(s)** button. The Assign as CE window appears.

Step 3 In the Assign as CE window, assign the required information for the CE.

a. To assign a Customer Name (required field), click the **Select** button.

The Customer Name window appears.

b. To assign a customer name, click the radio button next to the customer name that you want to assign and then click the **Select** button.

The Assign as CE window appears with the specified customer name displayed.

c. To assign a CE management type, select a value from the pull-down list for the CE Management Type.

The CE Management type specifies the architectural role that a CE router performs. Assign the CE management type based on the network layer to which the device belongs.

You can select the following CE management types:

- **MANAGED-REGULAR—**This is the default CE role assignment. Assign this role to CEs that you want the Provider to manage. The CE must be reachable from an Prime Fulfillment server. When you assign this role, then when you create a router in the Inventory Manager interface, the router configuration is automatically downloaded.
- **UNMANAGED—**Assign this role to a device that you want to manage manually. If this role is assigned, then the device configuration is not assigned automatically when a new device is created and the device must be configured manually. An unmanaged CE cannot be provisioned directly by the provider. If Unmanaged is selected, the provider can use Prime Fulfillment to generate a configuration, and then send the configuration to the customer for placement on the CE.
- **MANAGED-MGMT-LAN—**Specifies that the device management is linked to the PE configuration. The configuration is downloaded automatically when a new device is created. A managed Management LAN or Management CE (MCE) is configured like a managed CE router, but it resides in the provider space. Normally, an MCE acts as the network operations center (NOC) gateway router.
- **UNMANAGED-MGMT-LAN—**Specifies that the device management is linked to the PE configuration, but the configuration is not downloaded automatically when a new device is created. An unmanaged Management LAN or MCE is configured like an unmanaged CE router, but it resides in the provider space. Normally, an MCE acts as the network operations center (NOC) gateway router.

- **DIRECT-CONNECTED-REGULAR—**In most cases, the CE is connected to a PE router. In this case, the CE is connected to a workstation or other device.
- **DIRECT-CONNECTED-MGMT-HOST—**In most cases, the CE is connected to a PE router. In this case, the CE is connected to a workstation or other device on which Prime Fulfillment resides.
- **MULTI-VRF—**Specifies that there is a device between the PE and the CE that is a VPN routing/forwarding instance (VRF). A multi-VRF CE (MVRFCE) is owned by the customer, but resides in the provider space. It is used to off-load traffic from the PE.
- **UNMANAGED-MULTI-VRF—**An unmanaged multi-VRF CE is provisioned like an unmanaged CE (configurations are not uploaded or downloaded to the device by the provider). It is owned by the customer and resides in the provider space.

d. Click **OK**.

The Role Assignment - CEs window appears with the specified values shown.

Note The CE Site value is unassigned at this point. To assign this value, you must edit the settings. See Editing the CE Role, page G-39 for instructions on this task.

Editing the CE Role

After you have assigned one or more devices as CE devices and they appear in the Role Assignment - CEs window, you can edit the CE role. You can edit the CE role even if no values have been assigned in the Assign as CE window.

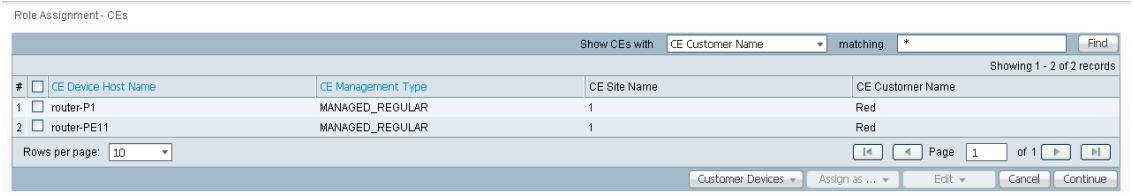
Follow these steps to edit the Role Assignment values for a CE device:

Step 1 While the Role Assignment phase of Device Discovery is active, choose the Role Assignment - CEs window.

If the Role Assignment - Un-assigned Devices or the Role Assignment - PE window is active, select **Role-Assignment - CEs** from the pull-down list at the bottom of the window.

The Role Assignment - CEs window appears.

Figure G-9 Role Assignment - CEs Window



In the sample Role Assignment - CEs window shown in Figure G-9, two of the CEs have role assignment information assigned, and two have no information assigned. You can edit any of the information for the CEs, whether information has been entered or not.

Note that on this window, sorting is disabled on the following columns:

- CE Device Host Name

- CE Site Name
- CE Customer Name

Step 2 Select one or more CEs to edit.

- To select a specific CE, check the check box next to the device name.
- To select all the CEs shown in the window, check the check box in the heading row.

Step 3 To edit the Customer name, follow these steps:

a. Click the **Edit** button at the bottom of the window and choose **Customer** from the pull-down list.

You are prompted to select a customer name.

b. To select a customer name, click the radio button next to one of the customer names that is displayed, and then click the **Select** button.

The Role Assignment - CEs window appears with the specified customer name displayed.

Step 4 To edit the CE management type, follow these steps:

a. Select one or more CEs to edit.

b. Click the **Edit** button at the bottom of the window and choose C**E Management Type** from the pull-down window.

The CE Management type specifies the architectural role that a CE router performs. Assign the CE management type based on the network layer to which the device belongs.

You can select the following CE management types:

- **MANAGED-REGULAR—**This is the default CE role assignment. Assign this role to CEs that you want the Provider to manage. The CE must be reachable from an Prime Fulfillment server. When you assign this role, then when you create a router in the Inventory Manager interface, the router configuration is automatically downloaded.
- **UNMANAGED—**Assign this role to a device that you want to manage manually. If this role is assigned, then the device configuration is not assigned automatically when a new device is created and the device must be configured manually. An unmanaged CE cannot be provisioned directly by the provider. If Unmanaged is selected, the provider can use Prime Fulfillment to generate a configuration, and then send the configuration to the customer for placement on the CE.
- **MANAGED-MGMT-LAN—**Specifies that the device management is linked to the PE configuration. The configuration is downloaded automatically when a new device is created. A managed Management LAN or Management CE (MCE) is configured like a managed CE router, but it resides in the provider space. Normally, an MCE acts as the network operations center (NOC) gateway router.
- **UNMANAGED-MGMT-LAN—**Specifies that the device management is linked to the PE configuration, but the configuration is not downloaded automatically when a new device is created. An unmanaged Management LAN or MCE is configured like an unmanaged CE router, but it resides in the provider space. Normally, an MCE acts as the network operations center (NOC) gateway router.
- **DIRECT-CONNECTED-REGULAR—**In most cases, the CE is connected to a PE router. In this case, the CE is connected to a workstation or other device.
- **DIRECT-CONNECTED-MGMT-HOST—**In most cases, the CE is connected to a PE router. In this case, the CE is connected to a workstation or other device on which Prime Fulfillment resides.

- **MULTI-VRF—**Specifies that there is a device between the PE and the CE that is a VPN routing/forwarding instance (VRF). A multi-VRF CE (MVRFCE) is owned by the customer, but resides in the provider space. It is used to off-load traffic from the PE.
- **UNMANAGED-MULTI-VRF—**An unmanaged multi-VRF CE is provisioned like an unmanaged CE (configurations are not uploaded or downloaded to the device by the provider). It is owned by the customer and resides in the provider space.

c. Click **Select**.

The Role Assignment - CEs window appears with the specified CE management type displayed.

Step 5 To specify a site name or edit an existing site name, follow these steps:

a. Select one or more CEs to edit.

b. Click the **Edit** button at the bottom of the window and choose **Site** from the pull-down window.

The Site Name window appears.

c. In the Site Name window, click the radio button next to the site name that you want to assign and then click the **Select** button.

The Role Assignment - CEs window appears with the specified site names displayed.

Saving the Role Assignment Information

After you finish assigning roles to the devices, click the **Continue** button.

The Role Assignment Discovery indicator turns green and indicates that Role Assignment is **Complete**.

You are now ready to start the NPC Discovery phase of Device Discovery.

Step 5: Perform NPC Discovery

After the Role Assignment phase of Device Discovery is complete, the Discovery Workflow window indicates that the NPC Discovery phase is **Pending Input**.

Follow these general steps to view a list of the NPCs that have been discovered and add or remove NPCs as required:

- If you are discovering *a Metro Ethernet topology with an Ethernet core*, perform the steps described in Preliminary Steps Before Completing NPC Discovery for Metro Ethernet Networks, page G-41.
- Complete the steps for starting NPC assignment as described in Starting NPC Assignment, page G-43
- If necessary, complete steps for adding or modifying NPCs as described in Adding a Device for an NPC, page G-44 and the sections that follow.

Preliminary Steps Before Completing NPC Discovery for Metro Ethernet Networks

Follow these steps if you are discovering a Metro Ethernet topology with an Ethernet core.

- Create one or more Access Domains and assign the devices that were discovered in the Device Discovery phase to the Access Domain(s).
- Create at least one Resource Pool.
- Edit the “inter N-PE interface” for each device.

These steps are performed using the Inventory and Connection Manager in the Service Inventory interface (**Service Inventory > Inventory and Connection Manager**).

Creating Access Domains

Follow these steps to create access domains and add discovered devices to the domains:

Step 1 In the Prime Fulfillment start page, select **Service Inventory**.

Step 2 In the Service Inventory window, select **Inventory and Connection Manager**.

The Inventory and Service manager window appears.

Step 3 In the left area of the window, select **Access Domains**.

The Access Domains window appears.

Step 4 Create one or more Access Domains and assign the devices in the L2VPN Metro Ethernet topology to these Access Domains.

For detailed instructions on creating Access Domains, see the Creating Access Domains, page 2-41 section of Chapter 2, “Before Setting Up Prime Fulfillment”.

Creating Resource Pools

Follow these steps to create a resource pool:

Step 1 In the Prime Fulfillment start page, select **Service Inventory**.

Step 2 In the Service Inventory window, select **Inventory and Connection Manager**.

The Inventory and Service manager window appears.

Step 3 In the left area of the window, select **Resource Pools**.

The Resource Pools window appears.

Step 4 Create a Resource Pools.

Step 5 For the **Pool Type**, make sure that you select **VLAN**.

Step 6 For the **Start** value, enter 2.

Step 7 For the **Pool Size** value, enter a value large enough to accommodate the number of devices in the resource pool, for example, 500.

For detailed instructions on creating Resource Pools, see the Resource Pools, page 2-44 section of Chapter 2, “Before Setting Up Prime Fulfillment”.

Editing Inter-N-PE Interfaces

Follow these steps to edit the "Inter N-PE" interfaces for the devices in your Metro Ethernet topology:

Note These steps are only required if the PE devices already exist in the repository.

Step 1 In the Prime Fulfillment start page, select **Service Inventory**.

Step 2 In the Service Inventory window, select **Inventory and Connection Manager**.

The Inventory and Service manager window appears.

Step 3 In the left area of the window, select **PE Devices**.

The PE Devices window appears.

Step 4 Select each PE device in your topology and do the following:

a. Click the **Edit** button

The Edit PE window appears.

b. Locate the interface that connects to each device that the device is connected to.

c. For each interface, in the Metro Ethernet column, change **Any** to **None**.

d. Save your changes

Go the following section, Starting NPC Assignment, page G-43 and follow the steps for starting NPC assignment.

Starting NPC Assignment

Follow these steps to initiate NPC assignment:

Step 1 In the Discovery Workflow window, click **Continue**.

The Named Physical Circuits window appears.

The Named Physical Circuits window initially displays any discovered circuits.

At this point, you can create, add, or remove NPCs as required.

The State column has the following categories:

- **New**—No corresponding NPC exists in Prime Fulfillment. Only the New NPCs are committed to Prime Fulfillment.
- **Existing**—The discovered NPC is the same as the NPC in Prime Fulfillment.
- **Existing Modified** —The NPC in Prime Fulfillment has the same source and endpoint but one or more of the intermediate links might not be the same.
- **Conflicting**—The discovered NPC conflicts with the NPC in Prime Fulfillment.

Named physical circuits (NPCs) are named circuits that describe a physical connection between a CPE or U-PE and a N-PE. The intermediate nodes of the NPCs can either be U-PE or PE-AGG. They can be connected in a circular fashion forming a ring of devices, which is represented by an entity known as

NPC Rings. NPC Rings represent the circular topology between devices to the Named Physical Circuits. To create an NPC, you must specify how the source CPE/U-PE and the destination N-PE are connected and specify the intermediate nodes.

Step 2 If you need to define an NPC, follow these steps:

a. In the Named Physical Circuits window, click **Create**.

The Create a Physical Circuit window appears.

Initially, the list of NPCs is empty.

b. Click the **Add Device** button

The Select a Device window appears.

Step 3 In this window, click the radio button for a device and then click the **Select** button.

The Create a Named Physical Circuit window appears with an initial device added.

The buttons on the window are now active.

c. Click a device that appears in the screen and then select one of the following actions:

- To insert a device, click the **Insert Device** button.
- To insert a ring, click the **Insert Ring** button.
- To add a device, click the **Add Device** button.
- To add a ring, click the **Add Ring** button.
- To delete an existing device or ring, select a device and then click the **Delete** button.

Step 4 Refer to the following sections for additional information.

Adding a Device for an NPC

Step 1 To select an incoming interface on the Create a Named Physical Circuit window click on **Select Incoming Interface**.

The Select Device Interface window appears. This window shows the interfaces on the selected device.

Step 2 Click the radio button next to an interface in the list and then click the **Select** button.

The selected interface now appears in the Create a Named Physical Circuit window.

Step 3 To select an outgoing interface, click on **Select Outgoing Interface**.

A list of interfaces configured on the device appears

Step 4 Click the radio button next to an interface in the list and then click the **Select** button.

The outgoing interface now appears in the Create a Named Physical Circuit window.

Step 5 Select additional devices as required and specify incoming and/or outgoing interfaces.

Step 6 After you are finished, click the **Save** button in the Create a Named Physical Circuit window.

Adding a Ring

Follow these steps to add a ring before the currently selected device:

Note Incremental service discovery of rings is not supported.

Step 1 In the Create a Named Physical Circuit window, click **Add Ring**.

The Select NPC Rings window appears. This window shows any rings that exist in the network topology.

Step 2 Click the radio button next to a ring listed in the window and then click the **Select** button.

The selected ring now appears in the Create a Named Physical Circuit window.

Inserting a Device

To insert a device after the last device in the topology, follow these steps:

Step 1 In the Create a Named Physical Circuit window, click the **Insert Device** button.

The Select a Device window appears.

Step 2 Check the check box next to a device that you want to insert and then click the **Select** button.

The device now appears on the Create a Named Physical Circuit window.

Step 3 Click **select incoming interface**.

A list of interfaces on the selected device appears.

Step 4 Check the check box next to the interface that you want to choose and then click **Select**.

The selected interface now appears on the list of interfaces.

Inserting a Ring

To insert a ring after the last device in the topology, follow these steps:

Step 1 In the Create a Named Physical Circuit window, click the **Insert Ring** button.

A list of the currently existing rings appears.

Step 2 In the list of rings, check the check box next to the ring that you want to insert and then click **Select**.

The selected ring now appears on the Create a Named Physical Circuit window.

Deleting a Device or a Ring

To delete a device or a ring, in the Create a Named Physical Circuit window, select a device or ring and then click the **Delete** button.

The create NPC window appears with the device deleted.

Saving the NPC Configuration

After you have selected two devices and have configured the connection between them, follow these steps to save the NPC configuration:

Step 1 In the Create a Named Physical Circuit window, click **Save**.

The NPC process validates the NPC configuration.

Step 2 Click **Continue** to continue.

The workflow window appears with NPC discovery marked as completed.

Step 6: Perform MPLS VPN Service Discovery (Optional)

After you have completed the NPC Discovery phase of Device discovery, if you selected **MPLS VPN Discovery** when you initiated the Discovery process, the NPC Discovery phase is marked as complete, and the MPLS VPN Discovery step is marked as **Pending Input**.

You are now ready to initiate configuration of the discovered MPLS VPN using the MPLS VPN Discovery user interface. Follow these steps to configure MPLS VPN services:

Note MPLS service discovery does not support devices running IOS XR.

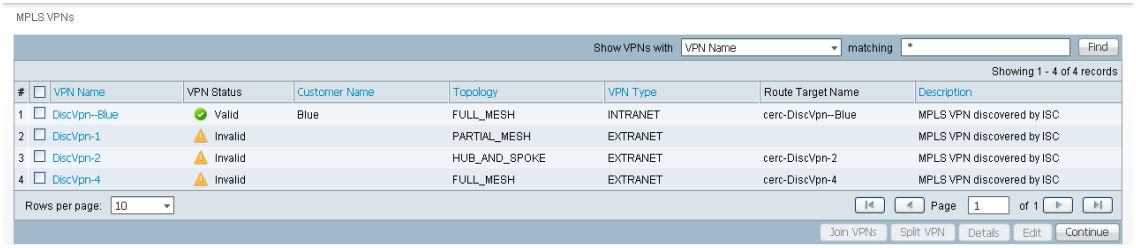
Step 1 In the Discovery Workflow window, click **Continue**.

The MPLS VPNs window appears and lists the MPLS VPNs that were discovered. The status of the discovered MPLS VPNs is indicated as follows:

- If the MPLS VPN topology for a discovered MPLS is valid and ready to save in the Prime Fulfillment Repository, then the VPN Status indicates a **Valid** VPN and the status indicator is green.
- If the MPLS VPN topology for a discovered MPLS is invalid (the topology is Partial Mesh), is missing a Customer assignment, or includes an invalid Route Target, then the VPN Status indicates an **Invalid** VPN and the status indicator is yellow. Partial Mesh topology VPNs are not supported by Prime Fulfillment, and must be broken into Full Mesh and/or Hub and Spoke components.

The MPLS VPN window shown in Figure G-10 shows an invalid MPLS VPN (the topology is Partial Mesh and the Customer Name is blank).

Figure G-10 MPLS VPNs Window with Invalid MPLS VPN



Note If the MPLS VPN Discovery process discovers an MPLS VPN with a Partial Mesh topology, you must split the VPN into two or more separate VPNs that have a supported topology (Hub and Spoke or Full Mesh).

Step 2 Do one of the following:

- If you want to change the view in the MPLS VPNs window, select another view option.

  For a description of the MPLS VPN view options, see Filtering the MPLS VPN View, page G-47.
- If the MPLS VPNs are valid and you do not need to make any changes to the MPLS VPN topology at this time, click **Continue** to create MPLS VPN services based on the discovered topology.
- If one or more of the discovered MPLS VPNs are invalid, you must complete the following steps:
  - **Split the VPN**—Select an invalid VPN and then click the **Split VPN** button.

    See Splitting a VPN, page G-48 for instructions.
  - **Create New VPNs and add Route Targets**—You must create new VPNs containing the devices in the VPN that you have split, and add Route Targets to each new VPN.

    See Creating a VPN, page G-49 for instructions.

Filtering the MPLS VPN View

Follow these steps to change the view in the MPLS VPNs window:

Step 1 Pull down the menu next to the **Show VPNs with** field.

You can filter the list of VPNs by VPN Name, Customer Name, Topology, VPN Type, or Description.

Step 2 To limit which VPNs are displayed within the selected category, enter a value in the **Matching** field.

The value in the **matching** field specifies a search mask that controls which sites are displayed. An asterisk (*) specifies display of all sites by the selected search criteria. A string followed by an asterisk specifies display of all sites starting with part of the element specified in the **Show VPNs with** field.

You can specify more than one wildcard (asterisk) value in a search string. For example, to display all VPNs that have “cisco” as part of the Customer Name, enter *cisco* in the matching field.

The display changes to display the VPNs with the selected criteria.

Splitting a VPN

In some situations, you might need to split an existing MPLS VPN before you complete the MPLS VPN Discovery process and actually create the MPLS VPN services.

For example:

- If the MPLS Service Discovery process discovers an invalid MPLS VPN (an MPLS VPN with a Partial Mesh topology), you must split the VPN into two or more Route Targets that have a supported topology (Hub and Spoke or Full Mesh).
- You might also choose to split MPLS VPNs to change your topology, depending on your processing needs. Only one VPN can be split at a time.

Follow these steps to split a VPN:

Step 1 In the MPLS VPNs window, check the check box next to a VPN that you want to split.

Step 2 Click the **Split VPN** button.

The Split VPN window appears, as shown in Figure G-11.

Figure G-11 Split VPN Window

Step 3 In the Split VPN window, select several of the links.

Select the links that would comprise either a Hub and Spoke or Full Mesh topology.

For example, in the Split VPN window shown in Figure G-11, the first three links all have Route Targets of **1:102** and together form a Full Mesh topology.

The remaining two links have Route Targets of **1:106** and **1:105**. These links together form a Hub and Spoke topology.

To split this VPN, the first three links need to be associated with one Route Target, and the two remaining links need to be associated with another Route Targets. Then we can split this VPN into two separate VPNs following the Prime Fulfillment best practice convention of one Route Targets per VPN.

Step 4 Click the **Create/Modify CERC** button.

You are prompted for a Route Targets name.

Step 5 Enter the new Route Targets name and then click the **Save** button.

Step 6 Repeat these steps for the rest of the devices that are included in invalid VPNs.

For example, in the topology shown Figure G-11, select the devices that have the route target **1:106 to 1:105**.

Step 7 Click the **Create/Modify CERC** button.

Step 8 When you are prompted for a Route Target name, enter the new Route Target name and then click the **Save** button.

The Split VPNs window appears again, and the window shows the new Route Targets that have been created.

Notice that in the example, the two new Route Targets that have been created (**valid_cerc_one** and **valid_cerc_two**), have valid topologies. The first Route Target, v**alid_cerc_one,** has a Full Mesh topology and the second Route Target, **valid_cerc_two**, has a Hub and Spoke topology.

Step 9 Click the **Save** button.

You are now ready to continue to the next step, creating VPNs and adding Route Targets to the VPNs.

Creating a VPN

After you have created a Route Target, you must create a VPN and then add the Route Target to it.

Follow these steps to create a VPN:

Step 1 In the Split VPN window, select **Create/Modify VPN.**

The Create VPN window appears.

Step 2 Select the Route Targets that you want to assign to the VPN.

Step 3 In the VPN Name field, enter a name for the VPN.

For this example, enter **vpn_one**.

Step 4 Click the **Assign VPN Name** button.

Step 5 Click **Save**.

The VPN is created and appears in the Split VPN window in the VPN Name field.

Step 6 Create any additional VPNs as needed.

Continuing with the Route Targets shown in the sample windows in Splitting a VPN, page G-48, a VPN must be created and have a Route Target assigned to it. To do this:

a. In the Split VPN window, click **Create/Modify VPN**.

b. In the Create VPN window, create a second VPN and assign a Route Target to it.

In the example screen, you could select the second Route Target (**valid_cerc_two**) to the newly created VPN to it.

Step 7 After you are finished creating VPNs, click the **Save** button in the Split VPN window.

The MPLS VPNs window appears.

Note In the example shown, one of the VPNs is marked as **Valid** and has a green status indicator. However, the other VPN shown in the window is marked as **Invalid** and has a yellow indicator.

This occurs because in some instances, the MPLS Discovery process cannot completely validate the data. In this situation, you can still continue with the Service Discovery process and create MPLS VPN services. However, the process will skip the invalid VPN, and you must configure the VPN service manually using the Prime Fulfillment provisioning commands.

Step 8 Follow these steps to assign a Customer to each VPN:

a. Select a VPN entry in the MPLS VPNs window and then click the **Edit** button.

The Edit VPN window appears.

b. Click the **Select** button next to the Customer Name field.

A list of customer names appears.

c. Click the radio button next to customer name and then **Select**.

d. If you want to rename the Route Target, click **Rename** and then rename it.

e. Click **Save**.

The Customer name now appears in the MPLS VPNs window.

Note In some cases, an apparently valid VPN will be marked as invalid. This VPN will be skipped in the processing. You will then have to configure it manually using the Prime Fulfillment provisioning commands.

Step 9 After you are finished editing VPNs, click the **Continue** button to initiate the MPLS VPN service creation process.

Viewing VPN Link Details

Follow these steps to view details of VPNs that were discovered:

Step 1 In the MPLS VPNs window, select a VPN that has details you want to view and then click the **Details** button.

The MPLS VPN Link window appears.

Step 2 To filter the MPLS VPN links that are displayed, select a value from the pull-down list in the **Show Sites with** field.

You can filter the list of VPNs by From Site, From CE, From CE Domain, Route Target, To Site, To CE, or to CE Domain.

The value in the **matching** field specifies a search mask that controls which sites are displayed. An asterisk (*) specifies display of all sites by the selected search criteria. A string followed by an asterisk specifies display of all sites starting with part of the element specified in the **Show Sites with** field.

You can specify more than one wildcard (asterisk) value in a search string. For example, to display all sites that have "realtime" in the From CE Name, select **From CE Name** in the **Show Sites with** field and then name, enter *realtime* in the matching field.

The display changes to show only the specified links.

Saving the MPLS VPNs and Initiating MPLS VPN Service Creation

After you are finished editing the data for the discovered MPLS VPNs in the MPLS VPNs window, click the **Continue** button.

The Discovery process creates VPN services. After the process is complete, the Discovery Workflow window indicates that the MPLS VPN Discovery process is **COMPLETE** and the status indicator is green.

If you also selected **L2VPN (Metro Ethernet) Discovery** in the Discovery window before starting the Discovery process, you can now proceed to Carrier Ethernet service discovery.

Step 7: Perform L2VPN (Metro Ethernet) Service Discovery (Optional)

If you selected **L2VPN (Carrier Ethernet) Discovery** in the Discovery window before starting the Discovery process, then after the previous steps are complete, the Discovery Workflow window shows the L2VPN (Metro Ethernet) Discovery as **Pending Input**.

Follow these steps to initiate Metro Ethernet Service Discovery:

Note L2VPN service discovery does not support devices running IOS XR and does not discover services defined using the EVC CLI framework.

Step 1 Before you initiate Metro Ethernet Service Discovery, follow these steps:

a. Choose **Service Inventory > Inventory and Connection Manager.**

b. In the task pane at the left of the Inventory and Connection Manager window, select **Access Domains**.

c. Create access domains for any N-PE devices in the Metro Ethernet topology.

   For detailed instructions, see the Creating Access Domains, page 2-41 section of Chapter 2, "Before Setting Up Prime Fulfillment".

d. Choose **Service Inventory > Inventory and Connection Manager.**

e. In the task pane at the left of the Inventory and Connection Manager window, select **Resource Pools**.

f. Create resource pools for each of the access domains that you created.

   For detailed instructions, see the Resource Pools, page 2-44 section of Chapter 2, "Before Setting Up Prime Fulfillment".

g. Choose **Service Inventory > Discovery.**

The Discovery Workflow window shows the L2VPN (Metro Ethernet) Discovery process as **Pending Input.**

Step 2 Click **Continue**.

The L2VPN Discovery (Ethernet Services) window appears.

Step 3 Select one of the following actions:

- **View/Edit Discovered Layer 2 Services grouped by VPN**—Allows you to view the discovered L2VPN services and edit them as required.
- **View/Edit Discovered Layer 2 End to End Wires**—Allows you to view the discovered Layer 2 End to End wires and edit them as required.
- **View/Edit Discovered Layer 2 VPLS Links**—Allows you to view the discovered Layer 2 Virtual Private LAN Service (VPLS) links and edit them as required.

The following sections of this chapter describe each of these actions.

Viewing Discovered Layer 2 Services Grouped by VPN

Follow these steps to view discovered Layer 2 services grouped by VPN:

Step 1 In the L2VPN Discovery (Ethernet Services) window, click the **VPNs** button.

The L2VPNs window appears.

The L2VPNs window allows you to perform the following tasks:

- View detailed information about a Layer 2 VPN.

  This task is explained in the following steps of this procedure.
- Display a window that allows you to edit the configuration information for an existing Layer 2 VPN.

  See Editing Discovered Layer 2 Services Grouped by VPN, page G-52 for detailed instructions.
- Delete an existing Layer 2 VPN.

  See Deleting Discovered Layer 2 Services Grouped by VPN, page G-53 for instructions on this task.

Step 2 To view detailed information about a Layer 2 service, check the check box next to a VPN that has details you want to view, and then click the **Details** button.

The L2VPN Details window appears.

The L2VPN Details window shows the details about the discovered VPN, such as the User-Network Interface (UNI), in a table format.

Step 3 When you are finished viewing the link details, click the **Close** button.

Editing Discovered Layer 2 Services Grouped by VPN

You can edit a discovered Layer 2 VPN service to change the policy that is applied to the service. Follow these steps to edit a Layer 2 VPN service:

Step 1 In the L2VPNs window, check the check box next to a VPN that you want to edit, and then click the **Edit** button.

The Edit VPN window appears.

Step 2 To edit the VPN name, enter a new VPN name in the VPN Name field.

Step 3 To edit the Customer Name, follow these steps:

a. Click the **Select** button next to the Customer Name.

A list of customers appears.

b. Click the radio button next to the new Customer Name that you want to configure.

c. Click the **Save** button.

The new VPN name and/or Customer Name appears in the Metro Ethernet End to End Wires window.

Deleting Discovered Layer 2 Services Grouped by VPN

Follow these steps to delete a Layer 2 service:

Step 1 In the L2VPNs window, check the check box next to a VPN that you want to delete, and then click the **Delete** button.

The following message appears:

```
Links/End to End wires associated with all selected VPNs will be deleted as a result of
this operation. Do you really want to Delete?
```

Step 2 If you are sure that you want to delete the VPN, click **OK**; otherwise, click **Cancel**.

If you click **OK**, the VPN and associated links and end-to-end wires are deleted.

Editing the Policy using Discovered Layer 2 VPN Services

You can edit a discovered Layer 2 VPN service to change the policy that is applied to the service. Follow these steps to edit a Layer 2 VPN service:

Step 1 In the L2VPNs Details window, check the check box next to a UNI associated to the VPN and then click the **Edit** button.

The Edit Link Policy window appears.

Step 2 To change the link policy for the service, follow these steps:

a. Click the **Policy** button next to the Policy Name field.

A list of policies appears.

You can change the list of policies by choosing a filter from the pull-down list in the **Show VPN policies with** field and/or entering a search mask in the **Matching** field.

You can filter the policy list by Policy Name, Customer Name, Provider Name, or Global policy name. And you can limit the lists of policies displayed in the selected category by entering a value in the Matching field.

Step 3 Click the radio button next to a policy that you want to apply to the service and then click **Select**.

Step 4 Do one of the following:

- Click **Save** to save your changes.
- Click **Cancel** to cancel the changes.

Viewing Discovered Layer 2 End to End Wires

Follow these steps to view discovered Layer 2 end-to-end wires:

Step 1 In the L2VPN Discovery (Ethernet Services) window, click the **End-End Wires** button.

The Metro Ethernet End to End Wires window appears.

The Metro Ethernet End to End Wires window allows you to perform the following tasks:

- View detailed information about a Metro Ethernet end-to-end wire.

  This task is explained in the following steps of this procedure.
- Edit the VPN associated with the end-to-end wire.

  See Editing the VPN Associated with an End to End Wire, page G-55 for a description of this task.
- Split an existing end-to-end wire into two end-to-end wires

  See Splitting Layer 2 Service End to End Wires, page G-55 for a description of this task.
- Join existing end-to-end wires into a single end-to-end wire

  See Joining Layer 2 Service End to End Wires, page G-55 for a description of this task.
- Delete an existing end-to-end wire.

  See Viewing Discovered Layer 2 End to End Wires, page G-54 for instructions on this task.

Step 2 To view detailed information about a Layer 2 service, check the check box next to a UNI that has details you want to view, and then click the **Details** button.

Step 3 When you are finished viewing the link details, click the **Close** button.

Step 4 If you want to view the details of the interfaces in the end-to-end wire, click the interface name in either the AC1 UNI or AC2 UNI field.

If you click on an interface name, the Interface Detail window appears. The Interface Detail window shows details about the selected interface, such as the hostname of the host where the interface is located, the type of encapsulation used on the interface, and the switch mode used on the interface.

Step 5 When you are finished viewing the interface details, click the **Close** button.

Editing the VPN Associated with an End to End Wire

From the Metro Ethernet End to End Wires window, you can also edit the VPN that is associated with the end-to-end wire.

Follow these steps to edit the VPN associated with an end-to-end wire:

Step 1 In the Metro Ethernet End to End Wires window, click a VPN name shown in the VPN name field.

The Edit VPN window appears.

Step 2 To edit the VPN name, enter a new VPN name in the VPN Name field.

Step 3 To edit the Customer Name, follow these steps:

a. Click the **Select** button next to the Customer Name.

A list of customers appears.

b. Click the radio button next to the new Customer Name that you want to configure.

c. Click the **Save** button.

The new VPN name and/or Customer Name appears in the Metro Ethernet End to End Wires window.

Splitting Layer 2 Service End to End Wires

You can split off an existing end-to-end wire from the VPN that it is associated with and associate it with a new VPN.

Follow these steps to split an end-to-end wire from an existing VPN:

Step 1 In the Metro Ethernet End to End Wires window, check the check box next to an end-to-end wire entry that you want to split from a VPN.

Note If there is only one ID for the VPN associated with the end-to-end wire, then you cannot perform a split action on the wire.

Step 2 Click the **Split** button.

A message appears asking if you want to proceed.

Step 3 If you want to continue with the process, click **OK**.

The end-to-end wires are split and are associated with two new VPNs. These names of the VPNs are created by the system by adding a new number to the end of the existing VPN name.

Joining Layer 2 Service End to End Wires

You can join two existing end-to-end wires to a single VPN.

Follow these steps to join two existing end-to-end wires:

Step 1 In the Metro Ethernet End to End Wires window, check the check box next to several end-to-end wire entries that you want to join.

A message appears asking if you want to proceed.

Step 2 If you want to continue with the process, click **OK**.

The selected end-to-end wires are joined to a new VPN. The name for this VPN is created by the system by adding a new number to the end of the existing highest numbered VPN name.

Deleting Layer 2 Service End to End Wires

Follow these steps to delete an existing end-to-end wire:

Step 1 In the Metro Ethernet End to End Wires window, check the check box next to one or more end-to-end wires that you want to delete.

A message appears asking if you want to proceed.

Step 2 If you want to continue with the process, click **OK**.

The selected end-to-end wire (or wires) is deleted. Any Attachment Circuit(s) associated with the wire(s) are also deleted.

Step 3 Click **Close** to close the Metro Ethernet End to End Wires window.

Viewing Discovered Layer 2 VPLS Links

Follow these steps to view discovered Layer 2 VPLS links:

Step 1 In the L2VPN Discovery (Ethernet Services) window, click the **VPLS Links** button.

The VPLS Links window appears.

The VPLS Links window allows you to perform the following tasks:

- View detailed information about a VPLS link.

  This task is explained in the following steps of this procedure.
- Display a window that allows you to edit the configuration information for an existing VPLS link.

  See Editing Discovered Layer 2 VPLS Links, page G-57 for detailed instructions.
- Delete an existing Layer 2 VPN.

  See Deleting Discovered Layer 2 VPLS Links, page G-57 for instructions on this task.

Step 2 To view detailed information about a VPLS link, check the check box next to a VPLS link that has details you want to view, and then click the **Details** button.

The VPLS Link Detail window appears.

The VPLS Link Detail window shows the details about the discovered VPN and its link properties.

Step 3 When you are finished viewing the link details, click the **Close** button.

Editing Discovered Layer 2 VPLS Links

You can edit a discovered Layer 2 VPLS link to change the policy that is applied to the service. Follow these steps to edit a Layer 2 VPLS link:

Step 1 In the VPLS Links window, check the check box next to a VPLS link that you want to edit and then click the **Edit** button.

The Edit Link Policy window appears.

Step 2 To change the link policy for the link, follow these steps:

a. Click the **Policy** button next to the Policy Name field.

A list of policies appears.

You can change the list of policies by choosing a filter from the pull-down list in the **Show VPN policies with** field and/or entering a search mask in the **Matching** field.

You can filter the policy list by Policy Name, Customer Name, Provider Name, or Global policy name. And you can limit the lists of policies displayed in the selected category by entering a value in the Matching field.

Step 3 Click the radio button next to a policy that you want to apply to the service, and then click **Select**.

Step 4 Do one of the following:

- Click **Save** to save your changes.
- Click **Cancel** to cancel the changes.

Deleting Discovered Layer 2 VPLS Links

Follow these steps to delete a VPLS link:

Step 1 In the VPLS Links window, check the check box next to a VPLS link that you want to delete and then click the **Delete** button.

The following message appears:

```
The selected link(s) will be deleted. Do you really want to Delete?
```

Step 2 If you are sure that you want to delete the VPLS, click **OK**; otherwise, click **Cancel**.

If you click **OK**, the VPLS link(s) are deleted.

Step 3 Click **Close** to close the VPLS links window.

Saving the L2VPN Metro Ethernet Policy and Initiating Service Creation

After you are finished viewing or editing the discovered L2VPN Metro Ethernet topology, click the **Close** button to return to the L2VPN Discovery (Ethernet Services) window.

Click the **Continue** button to initiate the L2VPN Service Discovery process.

The Discovery Workflow window appears and indicates that the L2VPN Service Discovery process is **In Progress**. The status indicator is yellow.

After the L2VPN Service Discovery process is complete, the status indicator changes to green, and the Discovery Workflow window indicates that the L2VPN Service Discovery process is **Complete.**

Step 8: Commit Discovered Devices and Services to Prime Fulfillment Repository

Click the **Continue** button to commit the discovered devices and services to the Prime Fulfillment repository. Prior to this step, discovery workflow stores the discovered devices and services in a temporary repository, which gets committed to Prime Fulfillment only at the last step of discovery workflow.

Step 9: Create and Run a Collect Config Task for the Discovered Devices

Before you view and edit services, follow these steps to run a Create Config task for the devices:

Note For additional information on the Create Config task, see the Tasks, page 10-23 section of Chapter 10, “Monitoring”.

Step 1 On the Prime Fulfillment Start page, select **Monitoring**.

The Monitoring window appears.

Step 2 Select **Task Manager**.

The Tasks window appears.

Step 3 Click the **Create** button and choose **Collect Config** from the pull-down list.

The Create Task window appears.

Step 4 Click the **Next** button.

The Collect Config Task window appears.

Step 5 On the Collect Config task window, follow these steps to create and run a Collect Config task:

a. Click the **Select/Deselect** button.

A dialog window appears, listing the devices that were discovered by the Discovery process.

b. Select all of the devices shown on the list.

c. Click the **Select** button.

The Collect Config Task window appears again.

d. Specify the additional settings for the Collect Config task as required.

e. Click the **Submit** button.

You are now ready to view and edit services as described in the following section, Step 10: View and Edit Services, page G-59

Step 10: View and Edit Services

After you have successfully completed the MPLS VPN and/or L2VPN Metro Ethernet service creation process, you can view the services that were created and modify them using the service requests editors.

Follow these steps to view the L2VPN services:

Step 1 If the Service Inventory window is not currently active, click the **Operate > Service Request > Service Request Manager**.

The Service Request Manager window appears.

You can modify the service requests shown in the Service Requests window as required.

Note If you need to edit MPLS VPNs as part of this process, see the Splitting a VPN, page G-48, Creating a VPN, page G-49, Viewing VPN Link Details, page G-50, and Saving the MPLS VPNs and Initiating MPLS VPN Service Creation, page G-51.

Step 2 For detailed information on modifying Service Requests for L2VPN Metro Ethernet networks, see the *Cisco Prime Fulfillment User Guide 6.2*.

Step 3 For general information on the release, see the *Release Notes for Cisco Prime Fulfillment 6.2*, provided with the release.

APPENDIX H

Adding Additional Information to Services

This appendix describes how the additional information feature is supported in Prime Fulfillment. It contains the following sections:

Overview

The additional information feature allows a set of attributes (name/value pairs) to be defined in an XML file by the user. The file is subsequently associated with a policy. The additional information attributes define values to be associated with a service request. They define labels and appearance in the GUI. In the service request workflow, these values can be entered by the user. It is also possible to access these additional attribute values either from templates or from the xDE provisioning logic, to provide data values that will be configured as part of a service. Using additional attributes in combination with templates allows template attribute values to be prompted for in the policy and service request GUI, instead of having to create data files with these values. This appendix provides the information needed to understand and use the additional information feature in Prime Fulfillment.

Prerequisites and Limitations

Be aware of the following prerequisites and limitations of the additional information feature:

- The additional information feature is only supported for MPLS, L2VPN, VPLS, and EVC services.
- MPLS-TP and TEM policies and service requests do not support additional information.
- VRF services requests do not have policies and so do not support additional information.

- Before using this feature in a supported policy or service request type, you must create an additional information definition file. This is an XML file that defines the user-defined attribute/value pairs. You later load this definition file in a step within the policy workflow. For more information about this, see Creating the Additional Information Definition File, page H-7.

Summary of the Additional Information GUI Workflow

The following steps summarize the tasks you need to perform to implement additional information in Prime Fulfillment. The remaining sections in this appendix provide detailed information on these topics.

1. Create an additional information definition file (perhaps using the supplied XSD to validate). This file defines the additional information attributes.
2. Create a template that refers to the additional attribute values or, alternatively, extend the xDE provisioning logic.
3. Create a single default data file for the template.
4. Optionally add the negate template and negate data file.
5. Create a policy of the appropriate policy type.
6. Go to the Additional Information window in the policy creation workflow.
7. Load in the additional information definition file that was created. The file will be parsed and validated, and any errors displayed in the GUI.
8. Fill in the values in the provided fields, if needed. You can define these in the additional information definition file if these are standard values that do not need to be changed.
9. In the policy workflow, mark the additional information attributes as editable or not. This determines whether or not you can edit these values in the service request based on the policy.
10. In the policy workflow, enable templates and reference the templates that access the additional values.
11. Save the policy. The additional information will be parsed and validated, and any errors displayed in the GUI.
12. Create a service request based on the policy.
13. The Service Request Editor window in the service request workflow will display the additional information attributes and allow you to edit them (if they are editable).
14. Save the service request. The additional information will be parsed and validated, and any errors displayed in the GUI.

Setting Additional Information in the Policy Workflow

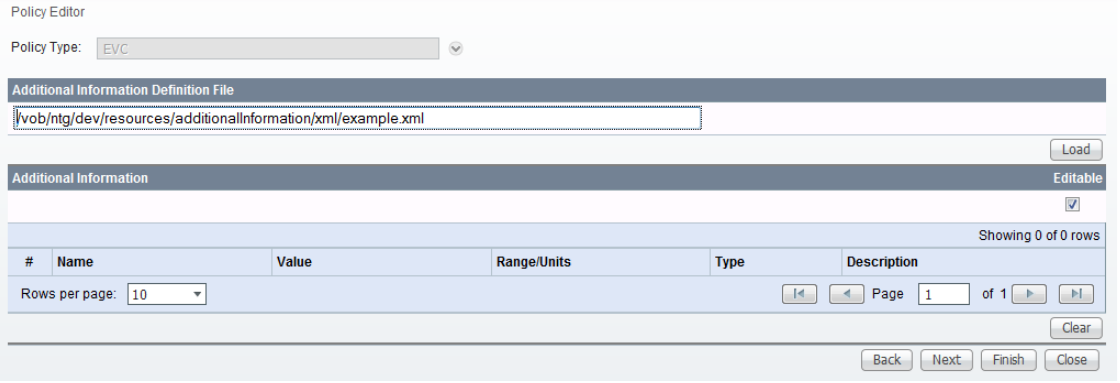
Perform the following steps to use the additional information feature within the supported policy types.

Step 1 Edit or create a supported policy type for which you want to add additional attributes.

Step 2 Navigate through the policy workflow windows and set attribute values as required for your configuration.

Several windows into the workflow, an Additional Information window appears. like the one shown in Figure H-1. This window looks and functions the same in all of the policy types that support the additional information feature.

Figure H-1 Additional Information Window



This window is the second to the last window of the policy workflow, and it appears before the Template Association window.

Use of the Additional Information window is optional.

Step 3 Click the **Load** button to load the XML definition file that defines the attribute/value pairs for the additional information to be added.

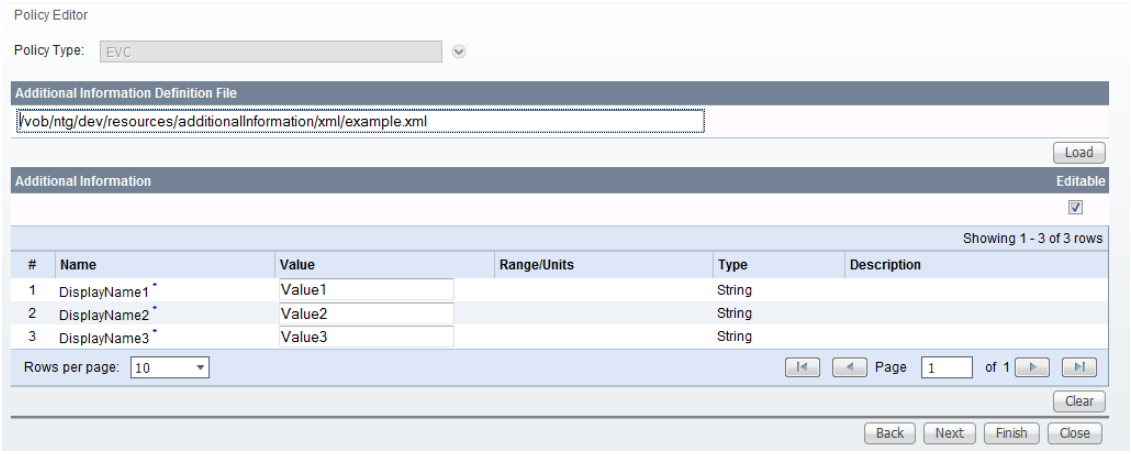
Note For information on how to create this file, see Creating the Additional Information Definition File, page H-7.

The default path and name of the definition file is:

$PRIMEF_HOME/resources/additionalInformation/xml/example.xml

The window refreshes and the attribute/value pairs from the definition file appear in the Additional Information section, as shown in Figure H-2.

Figure H-2 Attributes Loaded from an External XML Definition File



Step 4 If desired, you can click the Clear button to clear the attributes shown in the Display Section of the window.

Step 5 Check or uncheck the **Editable** check box to set all of the Additional Information attributes as editable or not.

You cannot make individual attributes editable or not.

Step 6 Set the values for the additional information attributes as desired for the policy.

See the discussion below for comments about the contents and behavior of this section of the window.

Step 7 Click **Next** to proceed to the next step of the policy workflow.

Step 8 Complete the policy workflow following the standard steps in Prime Fulfillment.

Be aware of the following points concerning the contents and behavior of the Additional Information section of the window:

- Additional Information attributes are grouped together in the GUI based on how they are defined in the additional information definition file.
- If groups are defined, then for each group the group name is displayed above a paging table containing the additional information attributes.
- If no groups are defined in the definition file, then only a paging table containing the additional information attributes is displayed.
- Each attribute is displayed in a row in the paging table.
- The Name column contains the DisplayName of the attribute, as defined in the definition file. If an attribute is marked as required in the definition file, a superscript asterisk is appended to the DisplayName. This does not indicate that the attribute must have a value in the policy, but that this is how it is defined in the definition file and that a value will be required for this attribute in a service request using this policy.
- The Value column contains the Value of the attribute, as defined in the definition file.
- The Range/Units column contains a combination of the range and units for the attribute.
- The Description column contains the Description of the attribute, as defined in the definition file.

Validation Checks Done to the Definition File in the Policy Workflow

In additional to the XSD validation, the parsing checks performed, and the validation performed for the additional information definition file, the following further validation checks are performed when a policy with additional information is saved to the Prime Fulfillment database. If the Additional Information section is marked as not editable (that is, the Editable check box is left unchecked), then any attributes marked as required need to have a value defined. A validation error is generated if this is not the case. This restriction is due to the fact that in a service request based on the policy, all required Additional Information attributes must have a value. So if you cannot edit the value (because the Additional Information is not editable) then you will never be able to create a service request based on the policy.

For more information about validation checks done on the additional information in the policy workflows, see How the XSD is Validated, page H-10.

Setting Additional Information in the Service Request Workflow

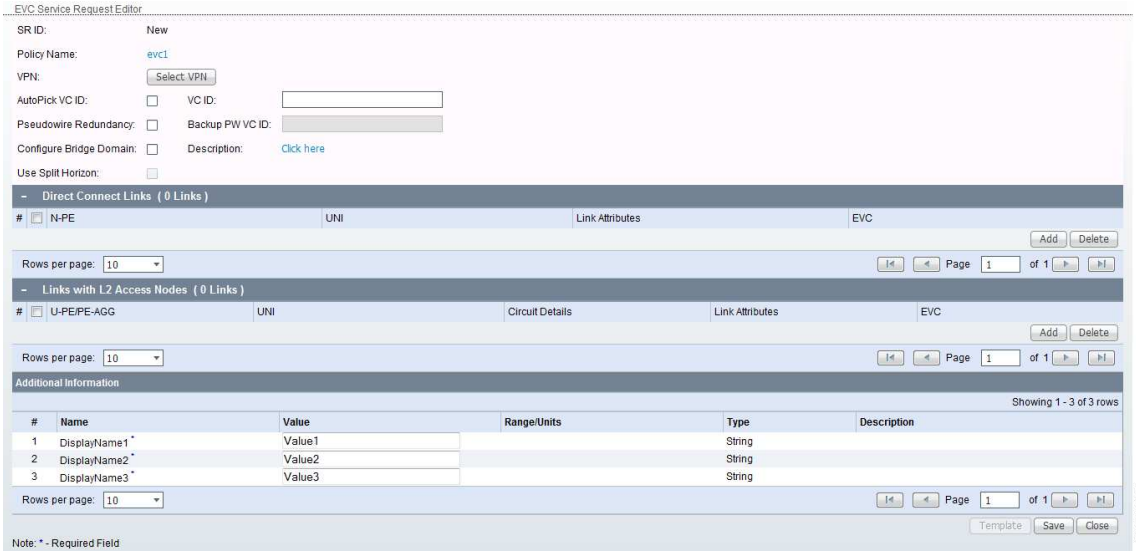
Perform the following steps to use the additional information feature in the service request workflow.

Step 1 Create or edit a service request based on a policy which was created using the additional information feature.

Step 2 Navigate to the Service Request Editor window within the service request workflow.

If the policy on which the service request is based had Additional Information attributes defined, these attributes are displayed, as shown in Figure H-3.

Figure H-3 Additional Information Attributes in the EVC Service Request Window



The attributes are displayed below any existing attributes of the Service Request Editor window. The format of the attribute in the Attribute Information is the same as the corresponding section in the policy.

Step 3 Set the attributes within the Service Request Editor based on the requirements for your configuration.

Be aware of the following points concerning the attributes Additional Information section:

- If the Additional Information attributes are editable, the values of the attributes can be changed.
- If the Additional Information attributes are not editable, the values are greyed-out and so cannot be changed.
- If there are no Additional Information attributes in the policy that the service is based upon, the Service Request Editor window will not show the Additional Information section.
- You must set values for attributes marked as required. If this is not done, a validation error is generated when you attempt to save the service request.

Step 4 At this stage, you may also add templates to the devices in order to map template variables to additional information attributes. For more information about this, see Using Additional Attributes with Templates and Data Files, page H-6.

Step 5 Click **Save** to save the service request.

Using Additional Attributes with Templates and Data Files

You can map template variables to Additional Information attributes in two places in Prime Fulfillment:

- When a template is created. To do this, perform the following steps:
  1. Edit the template variables that you want to map, and define them as type String.
  2. Enter the Additional Information attribute name as the default value for the template variable. You must use the exact name that is defined in the additional information definition file.

  Note The attribute used in the template must start with $ (for example, *$name*), as this indicates that this value will be substituted with another value at deployment time. When you create the default value or the data file for this attribute, then you give the exact name of the Additional Information attribute. The Additional Information attribute name must start with a $, as this indicates to the Template Manager that this attribute will be substituted with the actual value and is not just a hard-wired string.

- When a template data file is created. To do this, enter the Additional Information attribute name as the value for the template variable. You must use the exact name that is defined in the additional information definition file.

After you have performed either of these approaches, then when you associate a template and/or template data file with a policy or service request, the template variables are substituted with the values of the corresponding Additional Information attributes defined by the user in the policy or the service request.

Using Additional Attributes with xDE Provisioning

Additional Information attributes are added to the XML document that is passed to the xDE provisioning engine, and thus they can be accessed by any of the xDE procedures.

Append the following XML block to this XML document:

```
<additionalInformation>
<attribute>
```

```
<name>Name1</name>
<value>123</value>
</attribute>
</additionalInformation>
```

Note The attribute XML block must be repeated for each *additionalInformation* attribute.

In the current xDE procedures for provisioning, the request attribute is passed to every procedure that contains the input XML file. To use *additionalInformation* attribute values in the xDE procedure, you can extract the value of attribute Name1 from the MPLS SR XML request doc as follows:

```
xml.xpathreference($serviceRequest,
"/MplsSR/additionalInformation/attribute[name=\"Name1\"]/value/text()")
```

Alternatively, you can access the additional information attribute values via the $additionalInformation attribute that is passed to all xDE procedures. This attribute contains a map of all the additional information attribute name/value pairs. For example:

```
map.get($additionalInformation, "Name1")
```

returns the value associated with the Name1 attribute

Creating the Additional Information Definition File

This section provides reference information you can use to create an additional information definition file. This is an XML file containing a minimum set of mandatory XML elements, plus additional optional elements. This file is later loaded into a policy as described in the section Setting Additional Information in the Policy Workflow, page H-2.

Minimum Mandatory XML Elements

Example H-1 is an example additional information definition file that contains the minimum information needed to define an additional information attribute.

Example H-1 Additional Information Definition File with Minimum XML Elements

```
<additionalInformation>
<attribute>
       <name>Name1</name>
       <value>Value1</value>
   </attribute>
</additionalInformation>
```

Explanation of the mandatory XML elements:

- *additionalInformation*—The *additionalInformation* block starts and ends the definition file.
- *attribute*—The *attribute* block can be repeated for as many attributes as you would like to define. There must be only one *name* element and one *value* element in each *attribute* block.
- *name*—The *name* element must have non-null value, and this value must be unique with respect to the values of other *name* elements in the additional information definition file.

- *value*—The *value* element can have any value (including null), and this value does not need to be unique with respect to the values of other *value* elements in the additional information definition file.

Optional XML Elements

The additional information defintion file also may contains optional XML elements . This section describes the following optional elements:

- *group*
- *attribute/displayName*
- *attribute/description*
- *attribute/required*
- *attribute/type*
- *attribute/type/string*
- *attrribute/type/integer*
- *attribute/type/ipv4Address*
- *attribute/type/ipv6Address*
- a*ttribute/type/enumeration*

Information is provided on how each element is parsed and what conditions generate errors.

For an example additional information definition file that contains some of the optional elements that can be configured, see Example of the Additional Information Feature, page H-11.

group

There can be zero or more *group* elements.

Each *group* must have at least 1 attribute block. Zero attributes in a *group* will generate an error when the file is loaded.

If there is a *group* defined, then you cannot define attributes at the same level (that is, outside a *group*). *groups* and *attributes* at the same level will generate a parsing error when the file is loaded.

group elements must have a *name*, but *name* can be blank.

A *group name* must be unique, including blank names (that is, you can only have 1 blank *name*). A non-unique *group name* will generate a duplicate name error when the file is loaded.

attribute/displayName

The *displayName* element contains the text that is displayed in the Name column of the attribute in the the Additional Information table in the policy and service request workflow.

If *displayName* is not defined, it defaults to the text in the *name* element.

attribute/description

The *description* element contains the text that is displayed in the Description column of the attribute in the Additional Information table in the Policy and Service Request workflow. If *description* is not defined, it defaults to an empty string.

attribute/required

The *required* element contains a Boolean that indicates whether or not the attribute is required. If set to true, then a superscript asterisk is placed beside the *name* text that is displayed in the Name column of the attribute in the Additional Information table in the policy and service request workflow.

For policies, if an attribute is set as required, then it only needs to have a value if the Additional Information is set as not editable. Otherwise, the attribute does not need to have a value.

For service requests, if an attribute is set as required, then it needs to have a value set.

If *required* is not defined, it defaults to true.

attribute/type

The *type* element describes what type of attribute is being defined.

The available types are:

- *string*
- *integer*
- *ipv4Address*
- *ipv6Address*
- *enumeration*

If no *type* element is defined, then the default type is *string* (no ranges or regex is defined).

If the *type* element is defined but does not have one of the available types as a sub-element (either no type or a non-supported type), then this will generate a parsing error when the file is loaded.

If there are more than one *type* elements for an attribute, then a parsing error will be generated when the file is loaded.

attribute/type/string

The *string* type has a number of optional parameters that describe the range and units, as follows:

- *minLength*—Defines the minimum length of the string. The attribute string value length must be greater than or equal to this in order to pass validation. If *minLength* is not defined, then the default is 1.
- *maxLength*—Defines the maximum length of the string. The attribute string value length must be less than or equal to this in order to pass validation.
- *rangeUnits*—Defines the units to be displayed in the Range/Units column, in conjunction with the range parameters if defined. If *rangeUnits* is not defined then the default is “characters”.
- *regex*—Defines a regex that will be used to validate the attribute string value. The string value must satisfy the regex to pass validation. In addition, if *regex* is defined, then the *rangeDescription* will be appended with “Pattern: regex”.

attribute/type/integer

The *integer* type has a number of optional parameters that describe the range and units, as follows:

- *lower* —Defines the lower value of the range. The attribute *integer* value must be greater than or equal to this to pass validation.
- *upper* —Defines the upper value of the range. The attribute *integer* value must be less than or equal to this to pass validation.
- *rangeUnits*—Defines the units to be displayed in the Range/Units column, in conjunction with the range parameters if defined. If *rangeUnits* is not defined, then the default is an empty string.

attribute/type/ipv4Address

The *ipv4Address* type has a number of optional parameters that describe the range and units, as follows:

- *ipv4Lower*—Defines the *ipv4Lower* value of the range. The attribute *ipv4Address* value must be greater than or equal to this to pass validation.
- *ipv4Upper*—Defines the *ipv4Upper* value of the range. The attribute *ipv4Address* value must be less than or equal to this to pass validation.

attribute/type/ipv6Address

The *ipv6Address* type has a number of optional parameters that describe the range and units as follows:

- *ipv6Lower*—Defines the *ipv6Lower* value of the range. The attribute *ipv6Address* value must be greater than or equal to this to pass validation.
- *ipv6Upper*—Defines the *ipv6Upper* value of the range. The attribute *ipv6Address* value must be less than or equal to this to pass validation.
- *rangeUnits*—Defines the units to be displayed in the Range/Units column, in conjunction with the range parameters, if defined. If *rangeUnits* is not defined, then the default is an empty string.

attribute/type/enumeration

The *enumeration* type has a number of optional parameters that describe the range and units as follows:

- *enumOptions*—Defines the enumeration options for the attribute.
  - 1 or more *enumOptions* elements can be defined.
  - If there is not at least 1 *enumOption* element defined, then a parsing error will be generated when the file is loaded.
  - An empty string is not a valid *enumOption* value. If any of the *enumOption* elements have empty strings, a parsing error will be generated when the file is loaded.
- *rangeUnits*—Defines the units to be displayed in the Range/Units column, in conjunction with the range parameters, if defined. If *rangeUnits* is not defined, then the default is an empty string.

How the XSD is Validated

The additional information XML is validated using the XML schema definition (XSD). The XSD is defined in the main JAR file and so cannot be edited by the user. However, a copy of the file is available in the following location for users wanting to build additional information definition files:

$PRIMEf_HOME/resources/additionalInformation/extAttrs.xs

There is a DCPL property that allows the user to turn on/off the XSD validation. The DCPL property is **additionalInformation.XML.validateWithXSD**. It is on by default.

How the Additional Information Definition File is Validated

In addition to the XSD validation and the parsing checks performed, the following further validation checks are performed on the additional information definition file when it is loaded into a policy:

- *Enumeration* type—If an attribute value is defined but does not match one of the *enumeration* options, then a validation error is generated. If there are duplicate *enumeration* options, then a validation error is generated.
- *integer*, *ipv4Address* and *ipv6Address* types— If an attribute value is defined, then it is checked against the range (if no range defined then the defaults are used) and a validation error is generated if it is outside this range.
- *string* type—If an attribute value is defined, then in addition to the range checks (mentioned above), it must also match the *regex* (if it has been defined).

Example of the Additional Information Feature

This section provides an end-to-end example of the additional information feature. The example provides the following information:

- Template
- Template data file
- Additional information definition file
- List of attributes that display in the GUI
- Example GUI input and generated configlets

Template

Here is the example policy template body. The template is very generic. It shows an E-line service for an access port. It is for inbound traffic on a Cisco 3400 router.

```
policy-map qos-in-$Interface_Name
class class-default
#if($PIR_in_mbps==0)
     police cir $CIR_in_mbps m
#elseif($PIR_in_mbps!=0)
     police cir $CIR_in_mbps m pir $PIR_in_mbps m
#end

!
interface $Interface_Name
service-policy input qos-in-$Interface_Name
```

Template Data File

Here is the template date file to be attached to policy:

```
CIR_in_mbps:   $CIR_in_mbps
PIR_in_mbps:   $PIR_in_mbps
Interface_Name: $UNI_INTERFACE_NAME
```

Additional Attribute Definition File

Here is the additional information definition file:

```
<additionalInformation>
<group name="QoS">
        <attribute>
                <name>$CIR_in_mbps</name>
                <value></value>
                <displayName>Committed Bandwidth</displayName>
                <type>
                        <integer>
                                <lower>1</lower>
                                <upper>32000</upper>
                                <rangeUnits>Mbps</rangeUnits>
                        </integer>
                </type>
                <description>CIR value in Mbps</description>
                <required>true</required>
        </attribute>
        <attribute>
                <name>$PIR_in_mbps</name>
                <value></value>
                <displayName>Peak Bandwidth</displayName>
                <type>
                        <integer>
                                <lower>1</lower>
                                <upper>32000</upper>
                                <rangeUnits>Mbps</rangeUnits>
                        </integer>
                </type>
                <description>PIR value in Mbps</description>
                <required>false</required>
        </attribute>
</group>
</additionalInformation>
```

Additional Attributes Displayed in the Service Request Workflow

Based on this example, two new attributes are displayed in the service request workflow:

- Committed Bandwidth
- Peak Bandwidth

Committed Bandwidth is a required field, and Peak Bandwidth is an optional field.

User Input and Sample Configlets

The following examples show user input for the new attributes and the resulting configlets that are generated.

Example 1

User input:

- Committed Bandwidth: 25

Configlet generated:

```
policy-map qos-in-<uni interface>
class class-default
    police cir 25m
!
interface <uni interface>
service-policy input qos-in-<uni interface>
```

Example 2

User input:

- Committed Bandwidth: 25
- Peak Bandwidth: 50

Configlet generated:

```
policy-map qos-in-<uni interface>
class class-default
    police cir 25 m pir 50 m
!
interface <uni interface>
service-policy input qos-in-<uni interface>
```