



Cisco Prime Provisioning 7.2 User Guide

August 14, 2019

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:

THE SPECIFICATIONS AND INFORMATION REGARDING THE PRODUCTS IN THIS MANUAL ARE SUBJECT TO CHANGE WITHOUT NOTICE. ALL STATEMENTS, INFORMATION, AND RECOMMENDATIONS IN THIS MANUAL ARE BELIEVED TO BE ACCURATE BUT ARE PRESENTED WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED. USERS MUST TAKE FULL RESPONSIBILITY FOR THEIR APPLICATION OF ANY PRODUCTS.

THE SOFTWARE LICENSE AND LIMITED WARRANTY FOR THE ACCOMPANYING PRODUCT ARE SET FORTH IN THE INFORMATION PACKET THAT SHIPPED WITH THE PRODUCT AND ARE INCORPORATED HEREIN BY THIS REFERENCE. IF YOU ARE UNABLE TO LOCATE THE SOFTWARE LICENSE OR LIMITED WARRANTY, CONTACT YOUR CISCO REPRESENTATIVE FOR A COPY.

The Cisco implementation of TCP header compression is an adaptation of a program developed by the University of California, Berkeley (UCB) as part of UCB's public domain version of the UNIX operating system. All rights reserved. Copyright © 1981, Regents of the University of California.

NOTWITHSTANDING ANY OTHER WARRANTY HEREIN, ALL DOCUMENT FILES AND SOFTWARE OF THESE SUPPLIERS ARE PROVIDED "AS IS" WITH ALL FAULTS. CISCO AND THE ABOVE-NAMED SUPPLIERS DISCLAIM ALL WARRANTIES, EXPRESSED OR IMPLIED, INCLUDING, WITHOUT LIMITATION, THOSE OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT OR ARISING FROM A COURSE OF DEALING, USAGE, OR TRADE PRACTICE.

IN NO EVENT SHALL CISCO OR ITS SUPPLIERS BE LIABLE FOR ANY INDIRECT, SPECIAL, CONSEQUENTIAL, OR INCIDENTAL DAMAGES, INCLUDING, WITHOUT LIMITATION, LOST PROFITS OR LOSS OR DAMAGE TO DATA ARISING OUT OF THE USE OR INABILITY TO USE THIS MANUAL, EVEN IF CISCO OR ITS SUPPLIERS HAVE BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

Cisco and the Cisco logo are trademarks or registered trademarks of Cisco and/or its affiliates in the U.S. and other countries. To view a list of Cisco trademarks, go to this URL: www.cisco.com/go/trademarks. Third-party trademarks mentioned are the property of their respective owners. The use of the word partner does not imply a partnership relationship between Cisco and any other company. (1721R)

Any Internet Protocol (IP) addresses and phone numbers used in this document are not intended to be actual addresses and phone numbers. Any examples, command display output, network topology diagrams, and other figures included in the document are shown for illustrative purposes only. Any use of actual IP addresses or phone numbers in illustrative content is unintentional and coincidental.

Cisco Prime Provisioning 7.2 User Guide

Copyright © 1998-2019 Cisco Systems, Inc. All rights reserved.



Preface	xxxi
Objective	xxxi
Audience	xxxi
Organization	xxxii
Related Documentation	xxxii
Obtaining Documentation and Submitting a Service Request	xxxiii

CHAPTER 1

Prime Provisioning GUI Overview	1-1
System Recommendations	1-1
Introduction	1-1
Structural Overview	1-2
Links	1-3
User	1-3
Customer	1-4
TE Provider	1-4
Logout	1-4
About	1-4
Help	1-4
Common GUI Components	1-5
Filters	1-5
Header Row Check Box	1-6
Rows per Page	1-6
Go To Page	1-6
Auto Refresh	1-7
Color Coding	1-7
Icons	1-8
Operate	1-8
Inventory	1-9
Service Design	1-10
Traffic Engineering	1-10
Administration	1-11

Before Setting Up Prime Provisioning	2-1
Setting Up Devices and Device Groups	2-1
Devices	2-1
Configuring SSHv1 or SSHv2	2-2
Creating a Device	2-5
Copying a Device	2-11
Editing a Device	2-12
Deleting Devices	2-12
Editing a Device Configuration	2-12
E-mailing a Device's Owner	2-13
Device Configuration Collection	2-13
Synchronizing the Prime Provisioning Repository with Device Configuration	2-14
Providers	2-14
Creating a Provider	2-14
Editing a Provider	2-15
Deleting Providers	2-15
Provider Regions	2-16
Creating a Provider Region	2-16
Editing a Provider Regions	2-16
Deleting Provider Regions	2-17
Provider Devices	2-17
Creating a Provider Devices	2-17
Editing a Provider Devices	2-18
Deleting Provider Devices	2-18
Using the Inventory Manager Window	2-19
Importing Devices	2-19
Opening and Editing Devices	2-19
Opening and Editing PEs	2-20
Opening and Editing CEs	2-21
Assigning Devices	2-26
Device Groups	2-27
Creating a Device Group	2-27
Editing a Device Group	2-27
Deleting Device Groups	2-28
E-mailing a Device Group	2-28
Ethernet Access Topology Information	2-29
Physical Rings	2-29
Named Physical Circuits	2-31
Managing Customer Premise Devices	2-34

Customers	2-34
Customer Sites	2-35
Customer Devices	2-37
Setting Up Resources	2-39
Access Domains	2-40
Creating Access Domains	2-40
Editing Access Domains	2-40
Deleting Access Domains	2-41
Interface Access Domains	2-41
Creating Interface Access Domains	2-41
Editing Interface Access Domains	2-42
Deleting Interface Access Domains	2-42
Resource Pools	2-43
Creating an IP Address Pool	2-44
Creating an IPv6 Address Pool	2-45
Creating a Multicast Pool	2-45
Creating a Route Distinguisher and Route Target Pool	2-46
Creating a Site of Origin Pool	2-48
Creating a VC ID Pool	2-49
Creating a VLAN Pool	2-49
Creating an EVC Outer VLAN Pool	2-50
Creating an EVC Inner VLAN Pool	2-50
Creating a BVI Pool	2-51
Deleting Resource Pools	2-51
Supporting Manual Allocation of IPv6 Address from IPv6 Address Pool	2-52
Route Targets	2-52
Creating Route Targets	2-53
Deleting Route Targets	2-54
Setting Up Logical Inventory	2-55
VPNs	2-55
Creating a VPN	2-55
Deleting VPNs	2-57

Managing Ethernet Virtual Circuit (EVC) Services 3-1

Getting Started	3-2
Overview	3-2
Prepopulating a Service by Selecting Endpoints in Prime Network	3-2
Installing Prime Provisioning and Configuring the Network	3-3
Configuring the Network to Support Layer 2 Services	3-3

- Setting Up Basic Prime Provisioning Services 3-3
 - Setting Up Providers, Customers, and Devices 3-3
 - Setting Up the N-PE Loopback Address 3-4
 - Setting Up Prime Provisioning Resources for EVC Services 3-4
 - Setting Up NPCs 3-5
 - Setting Up VPNs 3-5
- Working with EVC Policies and Service Requests 3-5
- A Note on Terminology Conventions 3-6
- Setting Up the Prime Provisioning Services 3-7
 - Creating Target Devices and Assigning Roles (N-PE or U-PE) 3-7
 - Configuring Device Settings to Support Prime Provisioning 3-7
 - Configuring Switches in VTP Transparent Mode 3-7
 - Setting the Loopback Addresses on N-PE Devices 3-8
 - Setting Up Devices for IOS XR Support 3-8
 - Defining a Service Provider and Its Regions 3-9
 - Defining Customers and Their Sites 3-9
 - Defining VPNs 3-9
 - Creating Access Domains 3-10
 - Creating VLAN Pools 3-10
 - Creating Inner and Outer VLAN Pools 3-11
 - Creating a VC ID Pool 3-11
 - Creating Named Physical Circuits 3-12
 - Creating NPCs Through the NPC GUI Editor 3-13
 - Creating a Ring-Only NPC 3-14
 - Terminating an Access Ring on Two N-PEs 3-15
 - Creating NPC Links Through the Autodiscovery Process 3-15
 - Creating and Modifying Pseudowire Classes 3-15
 - Creating a Pseudowire Class 3-15
 - Modifying a Pseudowire Class 3-17
 - Deleting a Pseudowire Class 3-18
 - Configuring the Transport Mode When Pseudowire Classes are Not Supported 3-18
 - Defining L2VPN Group Names for IOS XR Devices 3-19
- Creating an EVC Ethernet Policy 3-19
 - Overview 3-20
 - Defining the EVC Ethernet Policy 3-20
- Managing an EVC Ethernet Service Request 3-22
 - Overview 3-22
 - Creating an EVC Service Request 3-23
 - Setting up Links to the N-PE 3-24

Using Templates and Data Files with an EVC Ethernet Service Request	3-31
Saving the EVC Ethernet Service Request	3-31
Modifying the EVC Ethernet Service Request	3-31
Deploying the EVC Ethernet Service Request	3-32
Creating an EVC ATM-Ethernet Interworking Policy	3-32
Overview	3-32
Defining the EVC ATM-Ethernet Interworking Policy	3-34
Customizing EVC and MPLS Policies	3-35
Managing an EVC ATM-Ethernet Interworking Service Request	3-35
Overview	3-36
Creating an EVC ATM-Ethernet Interworking Service Request	3-36
Setting up Links to the N-PE	3-37
Using Templates and Data Files with an EVC ATM-Interworking Service Request	3-41
Saving the EVC ATM-Interworking Service Request	3-42
Modifying the EVC ATM-Interworking Service Request	3-42
Deploying the EVC ATM-Ethernet Service Request	3-43
Defining Frame Relay Policies	3-43
Defining ATM Policies	3-44
Managing a VPLS Service Request	3-45
Overview	3-45
Creating a VPLS Service Request	3-46
Creating a VPLS Service Request with a CE	3-46
Creating a VPLS Service Request without a CE	3-48
Using Templates and Data Files with a VPLS Service Request	3-50
Saving the VPLS Service Request	3-51
Modifying the VPLS Service Request	3-51
Deploying, Monitoring, and Auditing Service Requests	3-51
Pre-Deployment Changes	3-52
Provisioning VPLS Autodiscovery on Devices using EVC Service Requests	3-52
Overview	3-53
Limitations and Restrictions for VPLS Autodiscovery	3-53
Preconfiguring PE Devices to Support VPLS Autodiscovery	3-54
Enabling VPLS Autodiscovery in the EVC Workflow	3-54
Sample Configlets	3-55
Policy and Service Request Attributes Reference Tables	3-56
EVC Ethernet Service Attributes	3-56
EVC Ethernet Policy Attributes	3-56
EVC Ethernet Service Request Attributes	3-74
EVC ATM-Ethernet Interworking Service Attributes	3-92

EVC ATM-Ethernet Interworking Policy Attributes	3-92
EVC ATM-Ethernet Interworking Service Request Attributes	3-106
Sample Configlets	3-122
Overview	3-124
ERS (EVPL) (Point-to-Point)	3-125
ERS (EVPL) (Point-to-Point, UNI Port Security)	3-126
ERS (EVPL) (1:1 VLAN Translation)	3-127
ERS (EVPL) (2:1 VLAN Translation)	3-128
ERS (Pseudowire Class, E-Line, L2VPN Group Name, IOS XR Device)	3-129
ERS (EVPL) (NBI Enhancements for L2VPN, IOS Device)	3-130
ERS (EVPL) and EWS (EPL) (Local Connect on E-Line)	3-131
ERS (EVPL), EWS (EPL), ATM, or Frame Relay (Additional Template Variables for L2VPN, IOS and IOS XR Device)	3-132
EWS (EPL) (Point-to-Point)	3-133
EWS (EPL) (Point-to-Point, UNI Port Security, BPDU Tunneling)	3-134
EWS (EPL) (Hybrid)	3-136
EWS (EPL) (Pseudowire Class, E-Line, L2VPN Group Name, IOS XR Device)	3-139
EWS (EPL) (NBI Enhancements for L2VPN, IOS Device)	3-140
ATM over MPLS (VC Mode)	3-141
ATM over MPLS (VP Mode)	3-142
ATM (Port Mode, Pseudowire Class, E-Line, L2VPN Group Name, IOS XR Device)	3-143
Frame Relay over MPLS	3-144
Frame Relay (DLCI Mode)	3-145
VPLS (Multipoint, ERMS/EVP-LAN)	3-146
VPLS (Multipoint, EMS/EP-LAN), BPDU Tunneling)	3-147
EVC (Pseudowire Core Connectivity, UNI Port Security)	3-148
EVC (Pseudowire Core Connectivity, UNI, without Port Security, with Bridge Domain)	3-149
EVC (Pseudowire Core Connectivity, UNI, and Pseudowire Tunneling)	3-150
EVC (Pseudowire Core Connectivity, With Pseudowire Headend Support)	3-151
EVC (Pseudowire Core Connectivity, Without Pseudowire Headend Support)	3-152
EVC (VPLS Core Connectivity, UNI Port Security)	3-153
EVC (VPLS Core Connectivity, no UNI Port Security)	3-154
EVC DOT1Q Encapsulation	3-155
EVC (VPLS Core Connectivity, With E-Tree Role, Communication between the Spokes of Different Hubs)	3-159
EVC (VPLS Core Connectivity, With E-Tree Role, Communication between the Spokes of Same HUB)	3-160
EVC (VPLS Core Connectivity, EFPs in same UNI, Switchport, CPT)	3-161
EVC (VPLS Core Connectivity, EFPs in Different UNI, Service Instance, CPT)	3-162
EVC (VPLS Core Connectivity, With MTU Attribute Support for VPLS Services)	3-163

EVC (Local Connect Core Connectivity, UNI Port Security)	3-164
EVC (Local Connect Core Connectivity, UNI, no Port Security, Bridge Domain)	3-165
EVC (Pseudowire Core Connectivity, Bridge Domain, Pseudowire on SVI)	3-166
EVC (Pseudowire Core Connectivity, no Bridge Domain, no Pseudowire on SVI)	3-167
EVC (AutoPick Service Instance Name)	3-168
EVC (No AutoPick Service Instance Name, No Service Instance Name)	3-169
EVC (Pseudowire Core Connectivity, User-Provided Service Instance Name)	3-170
EVC (Pseudowire Core Connectivity, Pseudowire Redundancy, "A" – "Z")	3-171
EVC (Pseudowire Core Connectivity, Pseudowire Redundancy, "A", "Z", and "Z'")	3-172
EVC (Pseudowire Core Connectivity, Pseudowire Redundancy, "A", "Z", and "Z'", where "Z" = "Z'")	3-173
EVC (Pseudowire Core Connectivity, Service Instance Syntax on L2 Access Nodes)	3-174
EVC (Pseudowire Core Connectivity, Mixture of Switchport and Service Instance Syntax on L2 Access Nodes, Push Outer Enabled)	3-175
EVC (Pseudowire Core Connectivity, Service Instance Syntax on L2 Access Nodes, Push Both Enabled)	3-177
EVC (Pseudowire Core Connectivity, Static Pseudowire, IOS Device)	3-178
EVC (Pseudowire Core Connectivity, Static Pseudowire, IOS Device, Pseudowire Redundancy)	3-179
EVC (Pseudowire Core Connectivity, Static Pseudowire, IOS Device, Bridge Domain Disabled)	3-180
EVC (Pseudowire Core Connectivity, Pseudowire Service with BVI)	3-181
EVC (Pseudowire Core Connectivity, Static Pseudowire, OAM Class Set in DCPL Property)	3-182
EVC (Local Core Connectivity, User-Provided Service Instance Name)	3-183
EVC (VPLS Core Connectivity, User-Provided Service Instance Name)	3-184
EVC (ATM-Ethernet Interworking, Pseudowire Core Connectivity, Point-to-Point Circuit)	3-185
EVC (ATM-Ethernet Interworking, Pseudowire Core Connectivity, Multipoint Circuit)	3-186
EVC (ATM-Ethernet Interworking, Pseudowire Core Connectivity, Pseudowire Redundancy)	3-187
EVC (ATM-Ethernet Interworking, Local Core Connectivity, Point-to-Point Circuit)	3-188
EVC (ATM-Ethernet Interworking, Local Core Connectivity, Multipoint Circuit)	3-189
EVC (ATM-Ethernet Interworking, Local Core Connectivity, Multipoint Circuit)	3-190
EVC (ATM-Ethernet Interworking, Local Core Connectivity, Point-to-Point Circuit)	3-191
EVC (ATM-Ethernet Interworking, Pseudowire Core Connectivity, End-to-End Circuit)	3-192
EVC (ATM-Ethernet Interworking, Pseudowire Core Connectivity, Multipoint Circuit)	3-193
EVC (ATM-Ethernet Interworking, Local Core Connectivity, Point-to-Point Circuit)	3-194
EVC (ATM-Ethernet Interworking, Pseudowire Core Connectivity, End-to-End Circuit, with Bridge Domain)	3-195
EVC (ATM-Ethernet Interworking, Pseudowire Core Connectivity, End-to-End Circuit, with Bridge Domain)	3-196
EVC (ATM-Ethernet Interworking, Pseudowire Core Connectivity, End-to-End Circuit, no Bridge Domain)	3-197
EVPL(Priority Tagged to Tagged, "A" – "Z")	3-198
EVPL(Priority Tagged to Untagged, "A" – "Z")	3-199

Managing TDM-CEM Services (RAN Backhaul)	4-1
Overview of RAN Backhaul Services	4-1
Overview of the CEM TDM Service	4-3
Prerequisites	4-4
Managing CEM Classes	4-4
Creating a CEM Class Object	4-4
Editing a CEM Class Object	4-5
Deleting a CEM Class Object	4-6
Sample Configlets for CEM Classes	4-6
Creating a TDM-CEM Policy	4-7
Setting the Service Options	4-8
Setting the Service Attributes	4-9
Using Pseudowire and CEM Classes	4-9
Adding User-Defined Fields into the TDM-CEM Policy Workflow	4-10
Enabling Template Association	4-10
Using Template Variables in TDM-CEM Services	4-11
Managing TDM-CEM Service Requests	4-11
Creating a TDM-CEM Service Request	4-12
Setting the Service Request Details	4-12
Selecting Devices	4-14
Modifying the TDM-CEM Service Request	4-18
Using Templates and Data Files with a TDM-CEM Service Request	4-19
Saving the TDM-CEM Service Request	4-19
Creating an E1-E1, T1-T1 and SERIAL- SERIAL circuit using Prime Provisioning	4-20
Sample Configlets for TDM-CEM Services	4-21
Overview	4-21
TDM-CEM using SAToP PW3	4-22
TDM-CEM using framing type SDH (IOS-XR device)	4-23
TDM-CEM using framing type SONET (IOS device)	4-24
TDM-CEM using framing type SONET (IOS-XR device)	4-25
TDM-CEM using CESoPSN	4-27
TDM-CEM between E1 controllers (IOS device)	4-28
TDM-CEM between E1 controllers (IOS-XR device)	4-29
TDM-CEM between T1 controllers (IOS device)	4-30
TDM-CEM between T1 controllers (IOS-XR device)	4-31
TDM-CEM between SERIAL controllers (IOS device)	4-32

CHAPTER 5

Managing ATM Services (RAN Backhaul)	5-1
Overview of RAN Backhaul Services	5-1
Overview of the ATM Service	5-3
Prerequisites	5-3
Managing Pseudowire Classes	5-4
Creating an ATM Policy	5-4
Setting the ATM Interface Attributes	5-5
Setting the Service Attributes	5-5
Using Pseudowire Classes	5-6
Adding User-Defined Fields into the ATM Policy Workflow	5-6
Enabling Template Association	5-6
Using Template Variables in ATM Services	5-7
Creating an ATM/IMA Interface Using Templates	5-7
Creating Template and Data File and Downloading it to a Device.	5-8
Adding ATM/IMA Interfaces to the Inventory	5-9
Managing an ATM Service Request	5-10
Creating an ATM Service Request	5-10
Setting the Service Request Details	5-11
Selecting Devices	5-13
Modifying the ATM Service Request	5-15
Using Templates and Data Files with an ATM Service Request	5-15
Saving the ATM Service Request	5-16
Sample Configlets for ATM Services	5-17
Overview	5-17
ATM/IMA PVP Service	5-18
ATM/IMA VCC Service	5-20
ATM PVC Service (IOS-XR device)	5-21
ATM PVP Service (IOS-XR device)	5-22
ATM/PVP Service (ASR platform, IOS device)	5-23
ATM/PVP Service (ASR platform, IOS-XR device)	5-24
ATM/PVC Service (ASR platform, IOS device)	5-25
ATM/PVC Service (ASR platform, IOS-XR device)	5-26

CHAPTER 6

Managing MPLS VPN Services	6-1
Getting Started with MPLS VPN	6-2
Before You Begin	6-2
Prime Provisioning Service Activation	6-2
Working with MPLS Policies and Service Requests	6-3

- Setting Up the Prime Provisioning Services 6-4
 - Overview 6-4
 - Setting Up Devices for IOS XR Support 6-6
 - Migrating PE Devices from IOS to IOS XR 6-6
 - Defining VPNs 6-6
 - Creating an MPLS VPN 6-7
 - Creating an IP Multicast VPN 6-9
 - Enabling a Unique Route Distinguisher for a VPN 6-11
 - Provisioning MPLS Service Requests Using Unique Route Distinguisher 6-12
- Independent VRF Management 6-14
 - Multicast Support for IPv6 on IOS XR Devices 6-15
 - Working with VRF Objects 6-16
 - Creating a New VRF Object 6-16
 - Copying a VRF Object 6-19
 - Searching for VRF Objects in the Prime Provisioning Repository 6-20
 - Modifying Non-Deployed VRF Objects 6-20
 - Modifying Deployed VRF Objects 6-21
 - Deleting VRF Objects 6-22
 - Working with VRF Service Requests 6-22
 - Overview of VRF Service Requests 6-23
 - Defining VRF Service Requests 6-23
 - Deploying VRF Service Requests 6-25
 - Modifying VRF Service Requests 6-25
 - Decommissioning and Deleting VRF Service Requests 6-26
 - Searching for VRF Service Requests by VRF Object Name 6-26
 - Viewing the Configlet Generated by a Deployed VRF Service Request 6-26
 - Using VRFs with MPLS VPN Service Requests and Policies 6-27
 - Relationship of VRF Object and Service Requests and PE Device 6-27
 - Specifying VRF Objects within MPLS VPN Service Requests 6-27
 - Notes On Using a VRF Object in an MPLS Service Request 6-29
 - Searching for MPLS VPN Service Requests by VRF Object Name 6-29
 - Specifying VRF Objects within MPLS VPN Service Policies 6-30
 - Migrating Existing MPLS VPN Service Requests to the VRF Object Model 6-30
- IPv6 and 6VPE Support in MPLS VPN 6-31
 - Overview of IPv6 and 6VPE 6-31
 - Internet Protocol Version 6 (IPv6) 6-31
 - IPv6 VPN Provider Edge Router (6VPE) 6-32
 - MPLS VPN Support for IPv6 and 6VPE 6-33
 - IOS and IOS XR Support for IPv6 6-33
 - Inventory and Device Management 6-33

Searching for MPLS VPN Service Requests by VRF Object Name	6-29
Specifying VRF Objects within MPLS VPN Service Policies	6-30
Migrating Existing MPLS VPN Service Requests to the VRF Object Model	6-30
IPv6 and 6VPE Support in MPLS VPN	6-31
Overview of IPv6 and 6VPE	6-31
Internet Protocol Version 6 (IPv6)	6-31
IPv6 VPN Provider Edge Router (6VPE)	6-32
MPLS VPN Support for IPv6 and 6VPE	6-33
IOS and IOS XR Support for IPv6	6-33
Inventory and Device Management	6-33
MPLS VPN Service Provisioning	6-34
Multicast Routing on IOS and IOS XR Devices	6-36
Multicast Support for IPv6 (IOS XR Only)	6-37
DCPL Properties Updated for IOS 6VPE Support	6-38
MPLS Reports	6-38
Upgrading an Existing IPV4 VRF to Be a Dual-Stack (IPV4+IPV6) VRF	6-38
Unsupported IPv6 and 6VPE Features	6-40
MPLS VPN Service Policies	6-40
Service Policy Overview	6-40
Service Policy Editor	6-41
About IP Addresses in Cisco Prime Provisioning	6-41
Defining an MPLS VPN Service Policy	6-42
Specifying PE and CE Interface Parameters	6-43
Specifying the IP Address Scheme	6-46
Using Existing Loopback Interface Number	6-48
Specifying the Routing Protocol for a Service	6-49
Redistribution of IP Routes	6-50
CSC Support	6-50
Giving Only Default Routes to CE	6-50
Static Protocol Chosen	6-51
RIP Protocol Chosen	6-52
BGP Protocol Chosen	6-55
OSPF Protocol Chosen	6-61
EIGRP Protocol Chosen	6-69
None Chosen: Cable Services	6-72
Defining VRF and VPN Information	6-73
BGP Multipath Load Sharing and Maximum Path Configuration	6-77
BGP Multipath Support for IOS XR Devices	6-80
BGP Multipath Support for ASR 920 Devices	6-80
Removing a Multipath Configuration	6-81

Supporting EVPN-VPWS Attributes through a Customization XML File	6-95
Supporting EVPN-VPLS Attributes through a Customization XML File	6-96
Creating a Multi-VRF Service Request	6-101
Creating a PE-Only Service Request	6-103
Adding a CLE to a Service Request	6-106
Migrating PE Devices from IOS to IOS XR	6-106
Pseudowire access into an L3VPN	6-106
Pseudowire Headend Interface	6-108
Provisioning Regular PE-CE Links	6-109
MPLS VPN PE-CE Link Overview	6-109
Network Topology	6-110
Prerequisite Tasks	6-110
Defining a VPN for the PE-CE Link	6-110
Creating MPLS VPN PE-CE Service Policies	6-111
PE-CE Service Policy Overview	6-111
Creating a PE-CE Service Policy	6-112
Creating a PE-NoCE Service Policy	6-113
Creating MPLS VPN PE-CE Service Requests	6-115
Creating PE-CE Service Requests	6-115
Creating PE-NoCE Service Requests	6-118
Provisioning Multi-VRFCE PE-CE Links	6-121
MPLS VPN MVRFCE PE-CE Link Overview	6-121
Network Topology	6-122
Prerequisite Tasks	6-122
Creating MPLS VPN MVRFCE PE-CE Service Policies	6-123
Creating MVRFCE PE-CE Service Policies	6-123
Creating PE-NoCE Service Policies	6-125
Creating MPLS VPN MVRFCE PE-CE Service Requests	6-126
Creating MVRFCE PE-CE Service Requests	6-126
Creating MVRFCE PE-NoCE Service Requests	6-129
Creating an Unmanaged MVRFCE	6-131
Provisioning Management VPN	6-132
Unmanaged Customer Edge Routers	6-132
Managed Customer Edge Routers	6-133
Network Management Subnets	6-134
Issues Regarding Access to VPNs	6-134
Implementation Techniques	6-135
Management CE (MCE)	6-135
Management PE (MPE)	6-135

Management VPN	6-136
Out-of-Band Technique	6-137
Provisioning a Management CE in Prime Provisioning	6-138
Defining CE as MCE	6-138
Creating MCE Service Requests	6-138
Adding PE-CE Links to Management VPNs	6-140
Provisioning Cable Services	6-141
Benefits of Cable MPLS VPNs	6-141
The Cable MPLS VPN Network	6-141
Management VPN in the Cable Network	6-143
Cable VPN Configuration Overview	6-143
Cable VPN Interfaces and Subinterfaces	6-144
Provisioning Cable Services in Prime Provisioning	6-145
Creating the Service Requests	6-145
Creating a Cable Subinterface Service Request	6-146
Creating Cable Link Service Requests	6-148
Provisioning Carrier Supporting Carrier	6-150
Carrier Supporting Carrier Overview	6-151
Backbone Network with ISP Customer Carrier	6-151
Backbone Network with BGP/MPLS VPN Service Provider Customer Carrier	6-153
Prime Provisioning Configuration Options	6-154
Defining CSC Service Policies	6-155
Provisioning CSC Service Requests	6-155
Provisioning Multiple Devices	6-155
NPC Ring Topology	6-155
Ring Topology Overview	6-155
Creating Ring of Three PE-CLEs	6-156
Configuring NPC Ring Topology	6-157
Ethernet-To-The-Home (ETTH)	6-159
Access Domain Management	6-161
Prime Provisioning ETTH Implementation	6-161
Creating an ETTH Policy	6-161
Creating a Service Request for ETTH	6-162
Residential Service	6-163
Creating a Policy for Residential Services Over Shared VLAN	6-163
Creating a Service Request for Residential Services Over Shared VLAN	6-164
Spanning Multiple Autonomous Systems	6-165
Overview	6-166
Benefits	6-166

Routing Between Autonomous Systems	6-167
Exchanging VPN Routing Information	6-168
Routing Between Subautonomous Systems in a Confederation	6-172
Using Prime Provisioning to Span Multiple Autonomous Systems	6-173
Using Templates to Support Inter-Autonomous System Solutions	6-175
Inter-AS 10B Hybrid Model	6-175
Inter-AS RT-Rewrite	6-176
Creating the Inter-AS Templates	6-176
Sample Configlets	6-177
Overview	6-177
L2 Access into L3 MPLS VPN	6-179
CE-PE L3 MPLS VPN (BGP with full-mesh)	6-181
CE-PE L3 MPLS VPN (BGP with S00)	6-182
CE-PE L3 MPLS VPN	6-184
PE L3 MPLS VPN (Dual-stack, Static [IPv4], BGP [IPv6], IOS)	6-185
CE-PE L3 MPLS VPN (Q-in-Q/Second VLAN ID, IOS)	6-187
CE-PE L3 MPLS VPN (Q-in-Q/Second VLAN ID, IOS XR)	6-189
PE L3 MPLS VPN (with Multicast, IPv4 and IPv6 Enabled VPN, IOS XR)	6-191
PE L3 MPLS VPN (Static, IOS, IPv6)	6-193
CE L3 MPLS VPN (Static, IOS, IPv6)	6-194
PE L3 MPLS VPN (BGP, IOS)	6-195
PE L3 MPLS VPN (BGP, IOS, IPv6)	6-196
PE L3 MPLS VPN (BGP, IOS XR)	6-197
PE L3 MPLS VPN (BGP, RD Format, IOS XR)	6-198
PE L3 MPLS VPN (BGP, Maximum Prefix/Restart, IOS XR)	6-199
PE L3 MPLS VPN (BGP, Default Information Originate, IOS XR)	6-201
PE L3 MPLS VPN (OSPF, IOS)	6-203
PE L3 MPLS VPN (OSPF, IOS XR)	6-204
L3 MPLS VPN (OSPF, Default Information Originate, IOS XR)	6-205
PE L3 MPLS VPN (EIGRP, Authentication Keychain Name, IOS XR)	6-207
PE L3 MPLS VPN (Independent VRF, IOS XR)	6-209
PE L3 MPLS VPN (Independent RTs for IPv4 and IPv6, IOS XR)	6-211
PE L3 MPLS VPN (Bundle-Ether Interface, IOS XR)	6-213
PE L3 MPLS VPN (Outgoing Interface + Next Hop IP Address, Static Route Configuration, IOS XR and IOS)	6-214
Troubleshooting MPLS VPNs	6-214
General Troubleshooting Guidelines	6-214

Gathering Logs for Development Engineering	6-215
Frequently Asked Questions	6-216
What is the MPLS provisioning workflow?	6-216
What do I do if my task does not execute even if I schedule it for immediate deployment?	6-216
What do I do when a service request is in the Wait Deployed state?	6-217
What do I do if the service request is in the same state as it was before a deployment?	6-218
What do I do if I receive the following out-of-memory error: OutOfMemoryError?	6-218
What do I do if Prime Provisioning will not remove a route target import/export for a VPN?	6-218
Why does my service request go to Invalid when I choose provisioning of an extra CE Loopback interface?	6-218
When saving a service request, why does it say "CERC not initialized"?	6-218
Why does creation of a VLAN ID pool require an Access Domain?	6-219
In a Paging table, why are the Edit and Delete options disabled, even though only one check box is checked?	6-219
Why can I not edit an MPLS VPN or L2VPN policy?	6-219
I am unable to create a CERC—can you explain why?	6-219
How can I modify the configlet download order between the PE, CE, and PE-CLE devices?	6-219
What does the property Provisioning.Service.mpls.reapplyIpAddress do?	6-219
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?	6-219
During service request provisioning, in the Interface selection list box, why don't I see the entire list of interfaces on the device?	6-219
Why does my service request go to Invalid with the message "loopback address missing"?	6-220
What is the intent of the Allocate New Route Distinguisher check box in the MPLS policy?	6-220
How can an MPLS service request using standard UNI ports allow CDP packets?	6-221
Is it possible to use 2 or 3 address pools when creating an L3 VPN?	6-222
When will an IP address from the MPLS IP address pool be returned to the available pool after the service request is decommissioned?	6-222
Why doesn't Prime Provisioning remove some of the router BGP/EIGRP commands when a service request is decommissioned?	6-222
VRFs	6-223
Creating a VRF	6-224
Editing VRFs	6-226
Deleting VRFs	6-226

Introduction	7-1
Prerequisites and Limitations	7-2

- Preconfiguration Process 7-2
 - MPLS-TP Setup and Installation 7-4
 - MPLS-TP User Roles 7-4
 - Other MPLS-TP Preconfiguration Requirements 7-4
- Running MPLS-TP Discovery 7-5
 - Creating an MPLS-TP Discovery Task 7-6
 - Verifying the MPLS-TP Discovery Results 7-7
 - Viewing Logs 7-7
 - Verifying Links, Pools, and MPLS-TP Global and Router IDs 7-7
 - MPLS Label Sync 7-7
- Creating an MPLS-TP Policy 7-7
 - Global ID and Router ID 7-9
 - Global ID 7-9
 - Router ID 7-9
- Creating an MPLS-TP Service Request 7-9
 - Working with Path Constraints 7-12
 - Running Config Audit 7-12
 - Running MPLS-TP Functional Audit 7-13
 - Managing MPLS-TP Topology Changes 7-13
- Deploying an MPLS-TP Tunnel 7-14
 - Decommissioning 7-15
- Sample Configlets 7-15
 - MPLS-TP Working Tunnel Configlet (IOS) 7-16
 - MPLS-TP Working Tunnel Configlet (IOS-XR) 7-17

CHAPTER 8

- Customizing EVC, MPLS and MPLS-TP Policies 8-1**
 - Customizing EVC and MPLS Policies 8-1
 - User Interface Customizations 8-1
 - Adding User Interface Groups to Pages 8-1
 - Adding User Interface Attributes to Groups 8-2
 - Command Line Interface Customizations 8-2
 - Creating Templates 8-3
 - Variable Completions for Specifying CLIs 8-3
 - Creating Rules for CLI Templates 8-4
 - Importing and Exporting Customizations 8-4
 - Changing Customizations When a Policy is in Use 8-5
 - Customizing MPLS-TP Policies 8-6

Managing MPLS Traffic Engineering Services	9-1
Getting Started	9-1
Process Overview	9-2
Prerequisites and Limitations	9-3
General Limitations	9-3
Feature-Specific Prerequisites and Limitations	9-3
Non-Cisco Devices and TEM	9-4
Supported Platforms	9-4
Error Messages	9-4
Preconfiguration Process Overview	9-4
TEM Setup and Installation	9-7
Editing DCPL Properties (Optional)	9-7
Creating a TE Provider	9-8
TE Network Discovery	9-11
TE Discovery Prerequisites and Limitations	9-13
Accessing TE Routers for TE Discovery	9-13
Memory Shortage on Large Networks	9-13
IOS XR and Enable Passwords	9-14
Limitations	9-14
Creating a TE Discovery Task	9-14
TE Incremental Discovery	9-14
TE Full Discovery	9-15
Managing Per Area Discovery	9-16
Performing a Per Area TE Discovery	9-16
Running a Per Area TE Discovery Through an ABR	9-17
Verifying a TE Discovery Task	9-17
Task Logs	9-17
View Network Element Types	9-20
Setting Up Management Interfaces	9-20
MPLS-TE Management Process	9-20
Configuring Ethernet Links	9-20
TE Resource Management	9-21
Modifying Network Resources	9-22
Changing Link Status	9-24
Deleting TE Links	9-25
Restrictions	9-25
Use Case	9-25
Note on Associated TE Objects	9-26
Deleting TE Tunnels	9-26

- Deleting TE Nodes 9-27
 - Restrictions 9-27
 - Use Case 9-27
- Basic Tunnel Management 9-28
 - Create TE Policy 9-29
 - Create Explicit Path 9-30
 - Delete Explicit Path 9-32
 - Primary Tunnel Operations 9-32
 - Create Primary Tunnel 9-33
 - Edit Primary Tunnel 9-38
 - Delete Primary Tunnel 9-39
 - Backup Tunnel Operations 9-40
 - Create Backup Tunnel 9-40
 - Edit Backup Tunnel 9-43
 - Delete Backup Tunnel 9-44
 - Deleting a Service Request 9-45
- Advanced Primary Tunnel Management 9-45
 - Tunnel Operations 9-46
 - Create Primary Tunnel 9-47
 - Edit Primary Tunnel 9-50
 - Delete Primary Tunnel 9-50
 - Admit Primary Tunnel 9-50
 - Import Primary Tunnel 9-50
 - Planning Strategy 9-52
 - Placement Tools 9-53
 - Tunnel Audit 9-53
 - Tunnel Placement 9-56
 - Tunnel Repair 9-57
 - Grooming 9-58
- Protection Planning 9-59
 - SRLG Operations 9-62
 - Create SRLG 9-62
 - Edit SRLG 9-62
 - Delete SRLG 9-63
 - Configure Element Protection 9-63
 - Protection Tools 9-63
 - Compute Backup 9-64
 - Recompute Backup 9-65
 - Audit Protection 9-66

Audit SR	9-67
TE Traffic Admission	9-68
Creating a TE Traffic Admission SR	9-69
Deploying a TE Traffic Admission SR	9-70
Other Traffic Admission SR Operations	9-71
Viewing the SR State	9-71
Administration	9-71
TE User Roles	9-72
TE Policies	9-72
Create Policy	9-72
Edit Policy	9-73
Delete Policy	9-74
TE Tasks	9-74
Creating a TE Task	9-74
SR History and Configlets	9-78
Managing the Locking Mechanism	9-78
Unlocking the TE Provider Lock	9-79
Unlocking the TE Router Lock	9-79
Locking Operation Errors	9-79
TE Topology	9-81
Using the TE Topology Interface Applet	9-82
Displaying and Saving Layouts	9-84
Using Maps	9-85
Using Highlighting and Attributes	9-87
Using Algorithms	9-88
Sample Configlets	9-89
Primary Tunnel Configlet (IOS)	9-90
Bandwidth Protection Backup Tunnel Configlet (IOS)	9-91
Connectivity Protection Backup Tunnel Configlet (IOS)	9-92
TE Traffic Admission Configlet Using CBTS (IOS)	9-93
TE Traffic Admission Configlet (IOS)	9-94
Primary Tunnel Configlet (IOS XR)	9-95
Bandwidth Protection Backup Tunnel Configlet (IOS XR)	9-96
Connectivity Protection Backup Tunnel Configlet (IOS XR)	9-97
TE Traffic Admission Configlet Using PBTS (IOS XR)	9-98
TE Traffic Admission Configlet (IOS XR)	9-99
Warnings and Violations	9-99
Warnings	9-100
Protection Computation Warnings	9-100

- Violations 9-101
 - Primary Placement Computation Violations 9-101
 - Protection Computation Violations 9-107
- Document Type Definition (DTD) File 9-109
 - DTD File 9-109
 - Example 9-112
- Traffic Engineering Management Concepts 9-112
 - Prime Provisioning TEM Overview 9-113
 - Features in Prime Provisioning 9-113
 - Prime Provisioning TEM Basics 9-113
 - Managed/Unmanaged Primary Tunnels 9-113
 - Conformant/Non-Conformant Tunnels 9-114
 - Multiple Concurrent Users 9-115
 - Multiple OSPF Areas 9-116
 - Bandwidth Pools 9-117
 - Planning Tools 9-118
 - Connectivity Protection (CSPF) Backup Tunnels 9-119
 - Class-Based Tunnel Selection 9-119
 - Policy-Based Tunnel Selection 9-120

CHAPTER 10

- Managing Service Requests 10-1**
 - Accessing the Service Request Manager Window 10-1
 - Viewing Service Request Details 10-2
 - Viewing Service Request History Information 10-3
 - Viewing Audit Reports Service Requests 10-3
 - Viewing Configuration Audit Reports 10-3
 - Viewing a Functional Audit Report 10-4
 - Viewing Service Request Configlets 10-5
 - Viewing Configlets on IOS XR Devices 10-5
 - Editing Configuration Files 10-6
 - Viewing the Status of Service Requests 10-7
 - Viewing Links 10-7
 - Viewing Logs 10-7
 - Previewing Configlets for Deploy and Decommission 10-8
 - Editing Service Requests 10-8
 - Deploying Service Requests 10-9
 - Service Deployment 10-9
 - Monitoring Service Requests 10-10
 - Simulated Deployment of Service Requests 10-11

Echo Mode	10-11
What is Echo Mode?	10-12
Who Should Use Echo Mode and When Should It Be Used?	10-12
How Should You Use Echo Mode?	10-12
Decommissioning Service Requests	10-12
Deleting Service Requests	10-13
Service Request States	10-14

CHAPTER 11

Managing Templates and Data Files 11-1

Overview	11-1
Summary of Template Manager Features	11-2
Template and Data File Workflow	11-4
Basic Template and Data File Tasks	11-5
Viewing the Templates Tree and Data Pane	11-5
Creating Folders and Subfolders	11-6
Copying Folders or Subfolders	11-6
Creating Templates	11-7
Negate Template	11-9
User Section	11-9
Optional Attributes	11-10
Sub-Template	11-12
Variables	11-13
Validate	11-16
Creating Data Files	11-16
Editing Templates and Data Files	11-18
Deleting Templates and Data Files	11-19
Listing Service Requests Associated with a Data File	11-20
Listing Policies Associated with a Data File	11-20
Using Templates with Policies	11-20
Overview	11-21
Associating Templates and Data Files to a Policy	11-21
Selectively Determining Templates for U-PE and PE-AGG Device Roles	11-23
Using Templates with Service Requests	11-24
Overview	11-24
Associating Templates to a Service Request	11-24
Associating Subtemplates During Service Provisioning	11-25
Creating Data Files During Service Request Creation	11-26
Using Negate Templates to Decommission Template Configurations	11-27
Using Templates and Data Files in the Service Request Workflow	11-28

- Choosing a Template in the Service Request Workflow 11-28
- Creating a Data File in the Service Request Workflow 11-29
- Decommissioning Service Requests with Added Templates 11-30
- Viewing Templates from the Service Requests Window 11-30
- Template Examples 11-31
- Summary of Repository Variables 11-33
- Importing and Exporting Templates 11-55
 - Known Issue with Importing Template Data Using the importExportTemplateDB.sh Script 11-56
- Frequently Asked Questions 11-56
 - How do I split a string? 11-57
 - How do I obtain address information from the given IP address? 11-57
 - How do I obtain the octets from the given IP address? 11-57
 - How do I call a subtemplate in a template? 11-58
 - How do I concatenate two strings? 11-58
 - How can I convert a string to an integer and how can I increase the last octet of the IP address by one? 11-58
 - Can I use nested if statements? 11-59
 - How can I perform basic arithmetic operations? 11-59
 - How can I retrieve data from a two-dimensional array and what is the use of \$velocityCount? 11-59
 - How can I print \$a instead of its value? 11-60
 - What is the difference between #include() and #parse()? 11-60
 - What is a macro and how is it used? 11-61
 - What is a range operator and how can I use it? 11-62
 - How can I split strings containing special characters? 11-62
 - How can I use repository variables? 11-62
 - How can I use a variable as a dynamic URL? 11-62
 - Can I see more examples? 11-63
 - Usage of Strings 11-63
 - Usage of a Macro 11-64
 - Usage of Subtemplates 11-65

CHAPTER 12

Monitoring: Task Manager 12-1

- Tasks 12-1
 - Starting Task Manager 12-1
 - Create 12-2
 - Audit 12-3
 - Details 12-3
 - Schedules 12-4
 - Logs 12-4

Delete	12-4
Collect Config from Files	12-4
Task Logs	12-5

CHAPTER 13

Using Inventory Manager	13-1
Inventory - Device Console	13-1
Download Commands	13-2
Download Template	13-3
Device Configuration Manager	13-6
EXEC Commands	13-8
Reload	13-10
Prime Network Device Import	13-11
Single Device Import during Device Creation	13-12
Importing Device(s) from Prime Network in Multi Instance Environment	13-13
Changing a Node to Unmanaged State	13-14

APPENDIX A

Prime Provisioning XML Reference	A-1
---	------------

APPENDIX B

Terminating an Access Ring on Two N-PEs	B-1
Overview	B-1
Setting Up an NPC Access Ring with Two N-PEs	B-3
Using N-PE Redundancy in FlexUNI/EVC Service Requests	B-3
Using N-PE Redundancy in MPLS Service Requests	B-4
Additional Network Configurations and Sample Configlets	B-5
Example 1: Pseudowire Connectivity (A)	B-5
Example 2: Pseudowire Connectivity (B)	B-6
Example 3: Pseudowire Connectivity (C)	B-8
Example 4: VPLS Connectivity	B-9

APPENDIX C

Repository Views	C-1
Creating Repository Views	C-1
Creating Views Sybase Repository	C-1
New and Upgrade Installation	C-1
Creating Views in Oracle Repository	C-2
New and Upgrade Installation	C-2
Using Views in Prime Provisioning	C-2
Summary View	C-2
Site View	C-4

Customer View C-5

Region View C-5

APPENDIX D

Adding Additional Information to Services D-1

Overview D-1

Prerequisites and Limitations D-1

Summary of the Additional Information GUI Workflow D-2

Setting Additional Information in the Policy Workflow D-2

Validation Checks Done to the Definition File in the Policy Workflow D-4

Setting Additional Information in the Service Request Workflow D-4

Using Additional Attributes with Templates and Data Files D-5

Using Additional Attributes with xDE Provisioning D-6

Creating the Additional Information Definition File D-7

Minimum Mandatory XML Elements D-7

Optional XML Elements D-7

group D-8

attribute/displayName D-8

attribute/description D-8

attribute/required D-8

attribute/type D-8

attribute/type/string D-9

attribute/type/integer D-9

attribute/type/ipv4Address D-9

attribute/type/ipv6Address D-9

attribute/type/enumeration D-10

How the XSD is Validated D-10

How the Additional Information Definition File is Validated D-10

Example of the Additional Information Feature D-10

Template D-11

Template Data File D-11

Additional Attribute Definition File D-11

Additional Attributes Displayed in the Service Request Workflow D-12

User Input and Sample Configlets D-12

Example 1 D-12

Example 2 D-12

APPENDIX E

Deprecated Features: Layer 2 Legacy Services and Other Services E-1

Getting Started with L2VPN Services E-2

Overview	E-2
Prepopulating a Service by Selecting Endpoints in Prime Network	E-2
Installing Prime Provisioning and Configuring the Network	E-3
Configuring the Network to Support Layer 2 Services	E-3
Setting Up Basic Prime Provisioning Services	E-3
Setting Up Providers, Customers, and Devices	E-3
Setting Up the N-PE Loopback Address	E-4
Setting Up Prime Provisioning Resources for L2VPN and VPLS Services	E-4
Setting Up NPCs	E-4
Setting Up VPNs	E-5
Working with L2VPN and VPLS Policies and Service Requests	E-5
A Note on Terminology Conventions	E-5
Setting Up the Prime Provisioning Services	E-6
Creating Target Devices and Assigning Roles (N-PE or U-PE)	E-7
Configuring Device Settings to Support Prime Provisioning	E-7
Configuring Switches in VTP Transparent Mode	E-7
Setting the Loopback Addresses on N-PE Devices	E-7
Setting Up Devices for IOS XR Support	E-8
Defining a Service Provider and Its Regions	E-9
Defining Customers and Their Sites	E-9
Defining VPNs	E-9
Creating Access Domains	E-9
Creating VLAN Pools	E-10
Creating a VC ID Pool	E-11
Creating Named Physical Circuits	E-12
Creating NPCs Through the NPC GUI Editor	E-13
Creating a Ring-Only NPC	E-14
Terminating an Access Ring on Two N-PEs	E-14
Creating NPC Links Through the Autodiscovery Process	E-15
Creating and Modifying Pseudowire Classes	E-15
Creating a Pseudowire Class	E-15
Modifying a Pseudowire Class	E-16
Deleting a Pseudowire Class	E-17
Configuring the Transport Mode When Pseudowire Classes are Not Supported	E-17
Defining L2VPN Group Names for IOS XR Devices	E-18
Creating an L2VPN Policy	E-19
Overview	E-19
Defining L2VPN Ethernet ERS and EWS Policies	E-20
Defining Frame Relay Policies	E-21
Defining ATM Policies	E-22

Managing an L2VPN Service Request	E-24
Overview	E-24
Creating an L2VPN Service Request	E-25
Creating an ERS, ATM, or Frame Relay L2VPN Service Request with a CE	E-25
Creating an ERS, ATM, or Frame Relay L2VPN Service Request without a CE	E-27
Creating an EWS L2VPN Service Request with a CE	E-30
Creating an EWS L2VPN Service Request without a CE	E-31
Using Templates and Data Files with an L2VPN Service Request	E-33
Saving an L2VPN Service Request	E-33
Modifying an L2VPN Service Request	E-33
Creating a VPLS Policy	E-35
Overview	E-35
Defining a VPLS Policy	E-36
Managing a VPLS Service Request	E-38
Overview	E-38
Creating a VPLS Service Request	E-39
Creating a VPLS Service Request with a CE	E-39
Creating a VPLS Service Request without a CE	E-41
Using Templates and Data Files with a VPLS Service Request	E-43
Saving the VPLS Service Request	E-43
Modifying the VPLS Service Request	E-44
Deploying, Monitoring, and Auditing Service Requests	E-44
Pre-Deployment Changes	E-44
Setting Up VLAN Translation for L2VPN ERS (EVPL) Services	E-45
VLAN Translation Overview	E-45
Setting Up VLAN Translation	E-45
Creating a Policy	E-46
Creating a Service Request	E-46
Modifying a Service Request	E-48
Deleting a Service Request	E-48
Platform-Specific Usage Notes	E-49
VLAN Translation on the 3750	E-49
VLAN Translation on the 7600	E-49
Failed Service Requests When Hardware Does Not Support VLAN Translation	E-49
Policy and Service Request Attributes Reference Tables	E-50
L2VPN Service Attributes	E-50
VPLS Service Attributes	E-58
Sample Configlets	E-63
Overview	E-63

ERS (EVPL) (Point-to-Point)	E-65
ERS (EVPL) (Point-to-Point, UNI Port Security)	E-66
ERS (EVPL) (1:1 VLAN Translation)	E-67
ERS (EVPL) (2:1 VLAN Translation)	E-68
ERS (Pseudowire Class, E-Line, L2VPN Group Name, IOS XR Device)	E-69
ERS (EVPL) (NBI Enhancements for L2VPN, IOS Device)	E-70
ERS (EVPL) and EWS (EPL) (Local Connect on E-Line)	E-71
ERS (EVPL), EWS (EPL), ATM, or Frame Relay (Additional Template Variables for L2VPN, IOS and IOS XR Device)	E-72
EWS (EPL) (Point-to-Point)	E-73
EWS (EPL) (Point-to-Point, UNI Port Security, BPDU Tunneling)	E-74
EWS (EPL) (Hybrid)	E-76
EWS (EPL) (Pseudowire Class, E-Line, L2VPN Group Name, IOS XR Device)	E-79
EWS (EPL) (NBI Enhancements for L2VPN, IOS Device)	E-80
ATM over MPLS (VC Mode)	E-81
ATM over MPLS (VP Mode)	E-82
ATM (Port Mode, Pseudowire Class, E-Line, L2VPN Group Name, IOS XR Device)	E-83
Frame Relay over MPLS	E-84
Frame Relay (DLCI Mode)	E-85
VPLS (Multipoint, ERMS/EVP-LAN)	E-86
VPLS (Multipoint, EMS/EP-LAN), BPDU Tunneling)	E-87
Reports	E-88
Introducing Reports	E-88
Accessing Reports	E-89
Using Reports GUI	E-89
Layout	E-89
Filters	E-89
Output Fields	E-90
Sorting	E-90
Running Reports	E-90
Exporting Reports	E-91
Printing Reports	E-91
E-mailing Reports	E-91
Creating Custom Reports	E-92
Generating L2 and VPLS Reports	E-93
Accessing L2 and VPLS Reports	E-93
L2 and VPLS Reports	E-93
Creating Custom L2 and VPLS Reports	E-100
Generating MPLS Reports	E-100
Accessing MPLS Reports	E-101

- Running Reports E-101
- MPLS PE Service Report E-102
- MPLS Service Request Report E-102
- MPLS Service Request Report - 6VPE E-103
- 6VPE Supported Devices Report E-104
- Creating Custom Reports E-105
- Generating TEM Reports and Logs E-105
 - TE Task Logs E-105
 - TE Performance Reports E-107
- EMAIL E-108

APPENDIX F

- Removed Features: Cisco Configuration Engine Server F-1**
 - Creating a Cisco CNS IE2100 Appliance F-1
 - Creating a Cisco IOS Device Using the Cisco CNS Device Access Protocol F-2
 - Using Plug-and-Play F-4

APPENDIX G

- Property Settings G-1**



Preface

This preface contains the following sections:

- [Objective, page xxxi](#)
- [Audience, page xxxi](#)
- [Organization, page xxxii](#)
- [Related Documentation, page xxxii](#)
- [Obtaining Documentation and Submitting a Service Request, page xxxiii](#)

Objective

The *Cisco Prime Provisioning 7.2 User Guide* contains detailed explanations of Prime Provisioning services and components across all applications.



Note

Prime Provisioning can be used as a standalone product or as part of the Prime for Carrier Management Suite. When installed as part of the suite, you can launch Prime Provisioning from the Prime Central portal. For more information about Prime Central, see the documentation for [Cisco Prime Central 2.1](#).

Audience

This guide is designed for service provider network managers and operators who are responsible for provisioning Prime Provisioning 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 Provisioning GUI Overview,”](#) describes how to get started with the Prime Provisioning graphical user interface (GUI).
- [Chapter 2, “Before Setting Up Prime Provisioning,”](#) describes how to set up the Cisco Prime Provisioning services
- [Chapter 3, “Managing Ethernet Virtual Circuit \(EVC\) Services,”](#) describes how to manage Ethernet Virtual Circuit (EVC) services.
- [Chapter 4, “Managing TDM-CEM Services \(RAN Backhaul\),”](#) describes how to manage CEM classes, work with TDM-CEM policies, and manage TDM-CEM service requests.
- [Chapter 5, “Managing ATM Services \(RAN Backhaul\),”](#) describes how to work with pseudewire classes, create ATM policies and ATM/IMA interfaces using templates, and manage ATM service requests.
- [Chapter 6, “Managing MPLS VPN Services,”](#) describes how to manage MPLS VPN services using policies and service requests.
- [Chapter 7, “Managing MPLS Transport Profile Services,”](#) describes how to manage MPLS Transport Profile services.
- [Chapter 8, “Customizing EVC, MPLS and MPLS-TP Policies,”](#) describes how to add custom data fields to the UI for a specific policy and how to embed customized CLI templates to EVC, MPLS and MPLS-TP policies.
- [Chapter 9, “Managing MPLS Traffic Engineering Services,”](#) describes how to manage MPLS Traffic Engineering services, including primary and backup tunnels.
- [Chapter 10, “Managing Service Requests,”](#) describes how to manage service requests using the Service Request Manager.
- [Chapter 11, “Managing Templates and Data Files,”](#) describes how to manage template and data files.
- [Chapter 12, “Monitoring: Task Manager,”](#) describes how to perform various tasks using Task Manager and produce reports using Prime Provisioning.
- [Chapter 13, “Using Inventory Manager,”](#) describes how to use the inventory manager in Prime Provisioning.
- Appendices provide supplementary information.

Related Documentation

See the [Cisco Prime Provisioning 7.2 Documentation Overview](#) for a list of all Prime Provisioning guides.

We sometimes update the documentation after original publication. Therefore, you should also review the documentation on Cisco.com for any updates.

Other Cisco Prime Product Documentation

If you are deploying Prime Provisioning as part of the Prime Carrier Management suite, then see also the documentation for the other suite components:

- [Cisco Prime Central 2.1](#)

- [Cisco Prime Network 5.1](#)
- [Cisco Prime Optical 10.7](#)
- [Cisco Prime Performance Manager 1.7](#)

Obtaining Documentation and Submitting a Service Request

For information on obtaining documentation, using the Cisco Bug Search Tool (BST), submitting a service request, and gathering additional information, see [What's New in Cisco Product Documentation](#).

To receive new and revised Cisco technical content directly to your desktop, you can subscribe to the [What's New in Cisco Product Documentation RSS feed](#). The RSS feeds are a free service.





Prime Provisioning GUI Overview

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

- [System Recommendations, page 1-1](#)
- [Introduction, page 1-1](#)
- [Structural Overview, page 1-2](#)
- [Operate, page 1-8](#)
- [Inventory, page 1-9](#)
- [Service Design, page 1-10](#)
- [Traffic Engineering, page 1-10](#)
- [Administration, page 1-11](#)

System Recommendations

The system recommendations and requirements are listed in Chapter 1, “System Recommendations” of the *Cisco Prime Provisioning 7.2 Installation Guide* and the *Cisco Prime Provisioning 7.2 Release Notes*. 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

Prime Provisioning 7.2 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 Provisioning are listed in the *Cisco Prime Provisioning 7.2 Release Notes*.

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 Provisioning.

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
- “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.

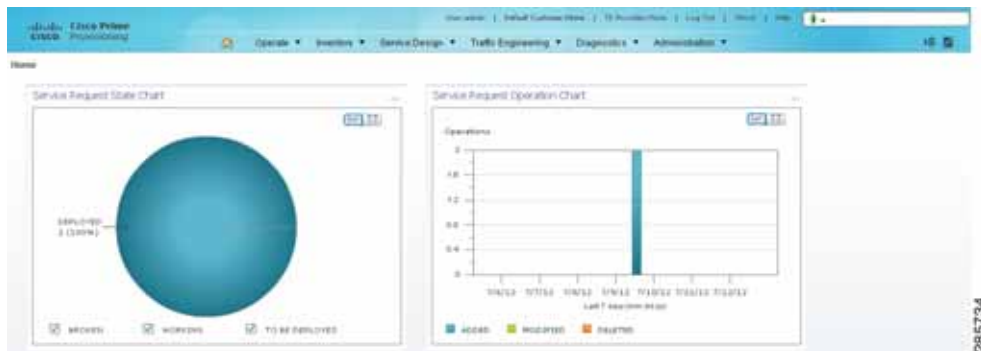

Note

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

Structural Overview

After you log into Prime Provisioning, 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, refer to the [Cisco Prime Provisioning 7.2 Administration Guide](#).

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 Provisioning 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 Provisioning. 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, page 1-3](#)

- [Common GUI Components, page 1-5](#)

Links

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

- [User, page 1-3](#)
- [Customer, page 1-4](#)
- [TE Provider, page 1-4](#)
- [Logout, page 1-4](#)
- [About, page 1-4](#)
- [Help, page 1-4](#)

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*

The screenshot shows a web interface for managing a user account. It is divided into several sections:

- User Account**: The main title of the window.
- Security**: A section with a dark header containing:
 - User ID: admin
 - Permissions for Others: View, Edit, Delete
 - Group Membership:
 - Assigned Roles: SysAdminRole
- Personal Information**: A section with a dark header containing:
 - Full Name: System Administrator
 - Work Phone:
 - Mobile Phone:
 - Pager:
 - Email:
 - Location: (represented by a large grey rectangular box)
 - Supervisor Information:
- User Preferences**: A section with a dark header containing:
 - Rows per page: 10
 - Logging Level: Warning

At the bottom right of the form, there is an **Edit** button and a vertical ID number 285735.

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.



Note

Policy or SR creator has to provide **Delete** permission along with **View** and **Edit** permissions to other users to edit the policy.

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. This link becomes active when a default customer is set. The default customer can edit or view customer context.

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 1-3 TE Provider Context

TE Provider

TE Provider Selection

TE Provider*:

Note: * - Required Field

285736

- 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 1-3**, 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.

About

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

Help

When you click **Help**, you receive a pointer to the Prime Provisioning 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 Provisioning document you want to see.

Common GUI Components

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

- [Filters](#), page 1-5
- [Header Row Check Box](#), page 1-6
- [Rows per Page](#), page 1-6
- [Go To Page](#), page 1-6
- [Auto Refresh](#), page 1-7
- [Color Coding](#), page 1-7
- [Icons](#), page 1-8

Filters

As shown in [Figure 1-4](#), you can filter information in the different windows of the software using the below instructions.

Figure 1-4 Example of Filtering, Header Row Check Box, Rows per Page, and Changing Pages



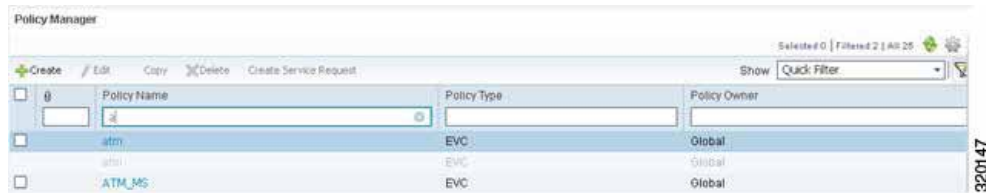
Note

Not all fields mentioned below are present in every window and the order of the fields might vary.

- Step 1** Select an option from the **Show Category with** drop-down list.
- Step 2** Enter the search criteria in the **matching** field, using * in any of the following ways:
 - a. Enter only * for searching.
 - b. Place * at the beginning, in between or at the end of characters.
- Step 3** In some screens, an extra field is present where you can select or enter more specifics.
- Step 4** Click **Find**.

As shown in [Figure 1-5](#), in some windows of the software such as Policy Manager, Customer and Toggle Picker of the Service Request Editor window, quick filtering option is present.

Figure 1-5 Example of Quick Filter



When you select **Quick Filter** from the **Show** drop-down list and start typing in any of the text fields, the list is automatically filtered. The count of the filtered records is shown in the top right corner. From the available records, if you want to keep certain rows visible as you scroll to others, you need to fix the rows.

To fix a row, do the following:

- Step 1 Select a row that you want to keep visible when you scroll.
- Step 2 Click the **Settings** icon in the top right corner.
- Step 3 Choose **Fix Row**.
- Step 4 Choose **Fix to Top** or **Fix to Bottom**.

The selected row remains fixed at the top or bottom as selected while also appearing greyed out in the scroll list. You can detach a row by selecting it and choosing **Detach Row** from the **Settings** icon.

Header Row Check Box

Many windows have a check box in the header row, where the column names exist, as shown in [Figure 1-4](#). 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 1-4](#), 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 1-4](#), 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.

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

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

#	SR ID	Data File	Status	Type	Op Type	Creator	Customer Name	Task Name	Last Modified	Description
1	4		DEPLOYED	EVC	ADD	admin	customer	sw-gw	11/01/12 8:07 AM	
2	5		DEPLOYED	EVC	ADD	admin	customer	sw-gw	11/01/12 8:16 AM	

Icons

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

Figure 1-7 Devices—Icons

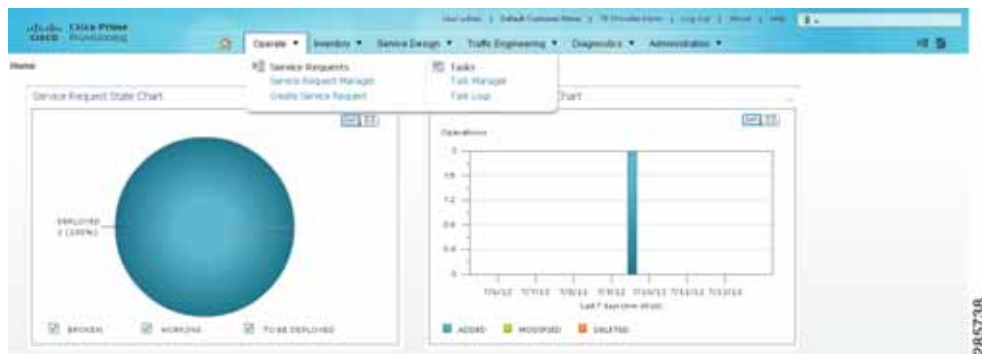
#	Device Name	Management IP Address	Type	Parent Device Name
1	sw-17075-1		Class XGS Device	
2	sw-17075-2		Class XGS Device	
3	sw-17075-1-1		Class XGS-IR Device	

Operate

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

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 10, “Managing Service Requests”.

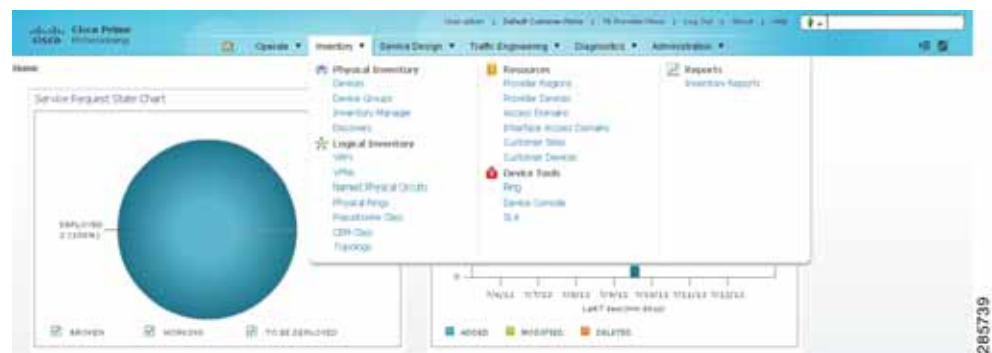
- **Tasks**—Create and manage the tasks associated with Prime Provisioning. This is explained in detail in [This chapter contains the following sections:](#), page 12-1 section of Chapter 12, “Monitoring: Task Manager”.

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:

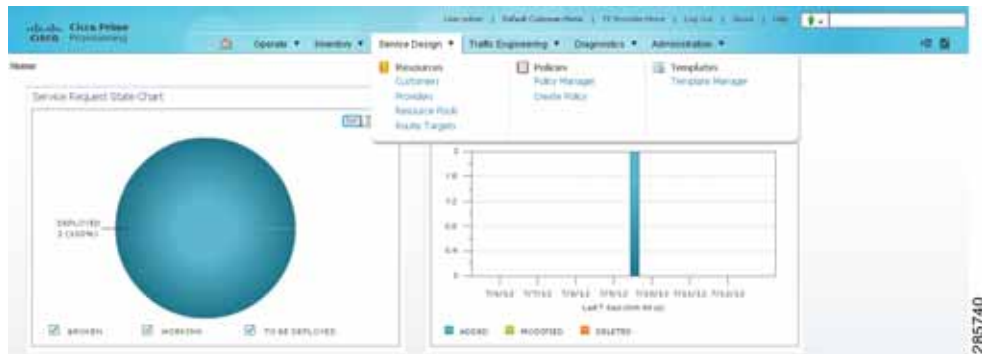
- **Physical Inventory**—Create and manage Devices, Device Groups, Inventory Manager, and Discovery.
 - **Devices**—Create and manage devices (explained in detail in [Devices](#), page 2-1 section of Chapter 2, “Before Setting Up Prime Provisioning”).
 - **Device Groups**—Create and manage device groups (explained in detail in [Device Groups](#), page 2-27 section of Chapter 2, “Before Setting Up Prime Provisioning”).
 - **Inventory Manager**—Bulk-manage inventory elements (explained in detail in [Chapter 13](#), “Using Inventory Manager”).
- **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 2-55 section of Chapter 2, “Before Setting Up Prime Provisioning”.
- **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 2-39 section of Chapter 2, “Before Setting Up Prime Provisioning”:
- **Device Tools**—Contains the following choice:
 - **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 Provisioning. This is explained on page 88 in the section [Appendix E](#), “Deprecated Features: Layer 2 Legacy Services and Other Services.”

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:

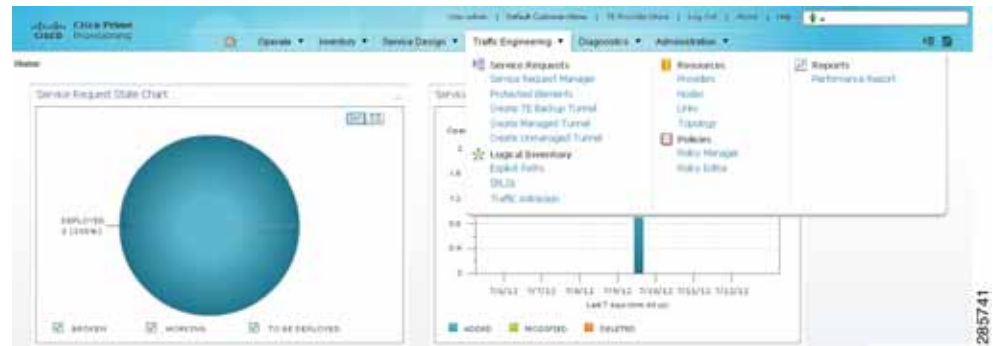
- **Resources**—Create and manage Customers, Providers, Resource Pools, and Route Targets. The following choices are explained in detail in [Setting Up Resources, page 2-39](#) section of [Chapter 2, “Before Setting Up Prime Provisioning”](#):
 - **Customers**—Create and manage customers.
 - **Providers**—Create 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 11, “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 9, “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

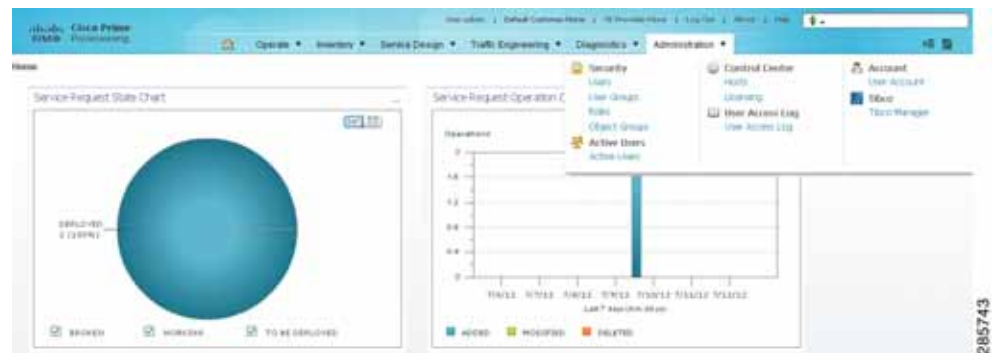


Administration

Administration contains tools to manage users, Prime Provisioning 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 1-12.

Figure 1-12 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 *Cisco Prime Provisioning 7.2 Administration Guide*.
 - **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 Provisioning configuration, servers, and licensing. The following choices are explained in detail in *Cisco Prime Provisioning 7.2 Administration Guide*.
 - **Hosts**

**Note**

If you want to do a **custom** install, this is only available through the Installation procedure explained in the *Cisco Prime Provisioning 7.2 Installation Guide*.

- **Collection Zones**
- **Licensing**
- **Active Users**—View users currently connected to Prime Provisioning. Disconnect users (explained in detail in *Cisco Prime Provisioning 7.2 Administration Guide*).
- **User Access Log**—View the user access log (explained in detail in *Cisco Prime Provisioning 7.2 Administration Guide*).



Before Setting Up Prime Provisioning

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

- [Setting Up Devices and Device Groups, page 2-1](#)
- [Setting Up Resources, page 2-39](#)
- [Setting Up Logical Inventory, page 2-55](#)

Setting Up Devices and Device Groups

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

- [Devices, page 2-1](#)
- [Device Configuration Collection, page 2-13](#)
- [Providers, page 2-14](#)
- [Provider Regions, page 2-16](#)
- [Provider Devices, page 2-17](#)
- [Using the Inventory Manager Window, page 2-19](#)
- [Device Groups, page 2-27](#)
- [Ethernet Access Topology Information, page 2-29](#)
- [Managing Customer Premise Devices, page 2-34](#)

Devices

Every network element that Cisco Prime Provisioning (Prime Provisioning) manages must be defined as a device in the system. An element is any device from which Prime Provisioning can collect information. Devices can be Cisco IOS XR and IOS XE routers that function as Provider Edge Routers (PEs) or Customer Edge Routers (CEs) in the MPLS VPN or they can be a variety of non-Cisco devices.



Note

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

This section describes how to configure SSHv1 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 SSHv1 or SSHv2, page 2-2](#)
- [Creating a Device, page 2-5](#)
- [Copying a Device, page 2-11](#)
- [Editing a Device, page 2-12](#)
- [Deleting Devices, page 2-12](#)
- [Editing a Device Configuration, page 2-12](#)
- [E-mailing a Device's Owner, page 2-13](#)

**Note**

To add non-Cisco devices in Prime Provisioning, please contact your Cisco representative and request information about activating and licensing this feature.

Configuring SSHv1 or SSHv2

Prime Provisioning 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, SSH version 1 (SSHv1) or SSH version 2 (SSHv2) must be enabled.

**Note**

SSHv1 is only supported for Cisco IOS devices.

The following sections describe:

- [Configuring SSHv1 on Cisco IOS Routers Using a Domain Name, page 2-2](#)
- [Configuring SSHv1 or SSHv2 on Cisco IOS Routers Using RSA Key Pairs, page 2-3](#)
- [Configuring SSHv1 or SSHv2 on Cisco IOS XR Routers, page 2-3](#)

Configuring SSHv1 on Cisco IOS Routers Using a Domain Name

The procedure for configuring SSHv1 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 <i><domain_name></i>	Specifies the IP domain name.
Step 3	Router(config)# username <i><username></i> password <i><password></i>	Configures the user ID and password. Enter your Prime Provisioning 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.

	Command	Description
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.
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 SSHv1 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.

	Command	Description
Step 5	RP/0/RP0/CPU0:router(config)# crypto key generate rsa [usage keys general-keys] [<keypair-label>]	Generates an RSA key pair.
Step 6	RP/0/RP0/CPU0:router# crypto key generate dsa	<p>Enables the SSH server for local and remote authentication on the router.</p> <p>The recommended minimum modulus size is 1024 bits.</p> <p>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.</p>
Step 7	RP/0/RP0/CPU0:router# configure	Enters global configuration mode.
Step 8	RP/0/RP0/CPU0:router# ssh timeout <seconds>	<p>(Optional) Configures the timeout value for user authentication to authentication, authorization, and accounting (AAA).</p> <p>If the user fails to authenticate itself to AAA within the configured time, the connection is aborted.</p> <p>If no value is configured, the default value of 30 is used for 30 seconds. The range is from 5 to 120.</p>
Step 9	RP/0/RP0/CPU0:router(config)# ssh server or RP/0/RP0/CPU0:router(config)# ssh server v2	<p>Brings up an SSH server.</p> <p>To bring down an SSH server, use the no ssh server command.</p> <p>(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.</p>
Step 10	RP/0/RP0/CPU0:router(config)# end or RP/0/RP0/CPU0:router(config)# commit	<p>Saves configuration changes.</p> <p>When you issue the end command, the system prompts you to commit changes: Uncommitted changes found, commit them before exiting (yes/no/cancel)? [cancel]</p> <p>Entering yes saves configuration changes to the running configuration file, exits the configuration session, and returns the router to EXEC mode.</p> <p>Entering no exits the configuration session and returns the router to EXEC mode without committing the configuration changes.</p> <p>Entering cancel leaves the router in the current configuration session without exiting or committing the configuration changes.</p> <p>Use the commit command to save the configuration changes to the running configuration file and remain within the configuration session.</p>

	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 device that runs the Cisco IOS. This includes Catalyst devices running Cisco IOS.
 - **Non Cisco Device**—Any device that does not run Cisco IOS.
 - **Terminal Server**—A device that represents the workstation that can be used to provision edge routers.
- Step 3** The following sections include examples with instructions for creating each type of device.
- [Creating a Catalyst Switch, page 2-5](#)
 - [Creating a Cisco or Non-Cisco Device, page 2-6](#)
 - [Creating a Terminal Server, page 2-7](#)
-

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 2-11](#)
 - [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 or Non-Cisco Device

To create a Cisco or Non-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 a device.
The Create 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** Perform the following steps, if you are creating a Cisco IOS device:
- a. Enter the desired information for the Cisco IOS device you are creating.
 - b. 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 2-11](#)
 - [Device Platform Information Attributes, page 2-11](#)
 - c. Enter any desired Additional Properties information for the Cisco IOS device you are creating.

Step 5 Click **Save**.

The Devices window reappears with the new 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 2-11](#)
- [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 Provisioning. Choices include: None and all collection zones within the Prime Provisioning. Default: None.
- **Management IP Address** —Valid IP address of the device that Prime Provisioning uses to configure the target router device.
- **Element Management Key** —Valid IP address of the device that Prime Provisioning.
- **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 Provisioning. However, collection and upload/download will not function without the Login User and Login Password as Prime Provisioning 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 Provisioning. However, collection and upload/download will not function without the Login User and Login Password, because Prime Provisioning 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 Provisioning. 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 Provisioning. 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 Provisioning and the device. Choices include: Telnet, SSH version 1 (SSHv1), RSH, and SSH version 2 (SSHv2). In previous versions of Prime Provisioning, 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.
- **SOCKS5 Proxy IP** (optional)—The device with SOCKS5 Proxy IP acts as an End-point Network Element (ENE) device and is used for both Telnet and SSH type of Terminal Session Protocol. This attribute configures the IP address of Gateway Network Element (GNE) as the proxy IP for the ENE device. ENE device cannot directly connect to any network or to any another ENE device. The connection is possible only using SOCKS5 proxy IP.



Note You can also configure this attribute for other Cisco devices which needs proxy connection.

- **PORT** (optional)—Connects to the Cisco devices via the specified port number while connecting using the Terminal Session Protocol either as Telnet or SSH.
- **OS** (optional)—The choices are: IOS, IOS_XR, and ME1200. 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_DCPL>*), Authentication/No Encryption, Authentication/Encryption, and No Authentication/No Encryption. Default: Default (*<default_set_in_DCPL>*). Note: When you change the DCPL property, the *<default_set_in_DCPL>* 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 Provisioning, 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 Provisioning, this field was called Privacy Protocol. Should be provisioned if the SNMP Security Level is Authentication/Encryption. Choices include: None, DES 56 and AES 128. 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 Provisioning performs additional management actions only for fully managed devices. These actions include e-mail notifications upon receipt of device configuration changes originated outside Prime Provisioning and the scheduling of enforcement audit tasks upon detection of possible intrusion. Default: Not selected and therefore not selected.
- **IE2100** (optional)—Disabled unless the Device State field is INACTIVE. Choices include: None and the list of existing IE2100 names. Default: None.

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.



Note

Email feature has been deprecated and will be removed in a subsequent release.

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 Provisioning 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 Provisioning Repository with this information.

For details of how to add Device interface configuration using a Task Manager Collect Configuration task, see [This chapter contains the following sections:](#), page 12-1.

Synchronizing the Prime Provisioning Repository with Device Configuration



Note

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 Provisioning 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 [This chapter contains the following sections; page 12-1](#). 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, page 2-14](#)
- [Editing a Provider, page 2-15](#)
- [Deleting Providers, page 2-15](#)

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 (1 to 65535) or a 32-bit number (1 to 4294967295) by the same central authority that assigns IP network numbers. This has an impact on the RD and RT values associated with the BGP AS number.

ASNumber:VPN ID/index (hex or decimal format)

When BGP AS is a 16 bit number, you need to enter a 32 bit value as the VPN ID/index and vice versa. If these values are not entered correctly, the service request moves to failed deploy state.

- **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, page 2-16](#)
- [Editing a Provider Regions, page 2-16](#)
- [Deleting Provider Regions, page 2-17](#)

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 2-17](#)
- [Editing a Provider Devices, page 2-18](#)
- [Deleting Provider Devices, page 2-18](#)

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, page 2-19](#)
- [Opening and Editing Devices, page 2-19](#)
- [Opening and Editing PEs, page 2-20](#)
- [Opening and Editing CEs, page 2-21](#)
- [Assigning Devices, page 2-26](#)

Importing Devices

To import a device, it must be in an existing directory on the same server that is running Prime Provisioning. After a device is imported into the Prime Provisioning 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 Provisioning 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 Provisioning 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 Device Select Picker 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 2-22](#)
- [Password Attributes, page 2-23](#)
- [SNMP Attributes, page 2-23](#)
- [Platform Attributes, page 2-24](#)
- [Interfaces, page 2-24](#)

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

- Check the one or more boxes to the left of the Device Name.
- Check the check box above the attribute name column.
- 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 Provisioning 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 2-22](#)
 - [Password Attributes, page 2-23](#)
 - [SNMP Attributes, page 2-23](#)
 - [Platform Attributes, page 2-24](#)
 - [Platform Attributes, page 2-24](#)
 - [PE Attributes, page 2-25](#)
 - [Interfaces, page 2-24](#)
- 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 Provisioning 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 2-22](#)
- [Password Attributes, page 2-23](#)
- [SNMP Attributes, page 2-23](#)
- [Platform Attributes, page 2-24](#)
- [CPE Attributes, page 2-25](#)
- [Interfaces, page 2-24](#)

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 Provisioning uses to configure the target router device. This IP address must be reachable from the Prime Provisioning 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 Provisioning and the device. Choices include: Telnet, SSH version 1 (SSHv1), SSH version 2 (SSHv2), 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 Provisioning. However, collection and upload/download will not function without the Login User and Login Password, as Prime Provisioning 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 Provisioning. However, collection and upload/download will not function without the Login User and Login Password, as Prime Provisioning 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 Provisioning. 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 Provisioning. 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 G, “Property Settings”](#) for more details. (DCPL properties can be set in the software UI, see [Cisco Prime Provisioning 7.2 Administration Guide](#).)

- **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, DES 56 and AES 128. Default: None.

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 Provisioning. 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 Device Select Picker 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 Provisioning 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 Provisioning repository.

Device Groups

Every network element that Cisco Prime Provisioning 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, page 2-27](#)
- [Editing a Device Group, page 2-27](#)
- [Deleting Device Groups, page 2-28](#)
- [E-mailing a Device Group, page 2-28](#)

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, page 2-29](#)
- [Named Physical Circuits, page 2-31](#)

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, page 2-29](#)
- [Editing Physical Rings, page 2-31](#)
- [Deleting Physical Rings, page 2-31](#)

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 Ring 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 Ring 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 2-29](#) 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 Provisioning software, the device you select can be any node in the link. The Prime Provisioning 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 2-32](#)
- [Deleting Named Physical Circuits, page 2-33](#)

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 2-29](#).



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 Device Select Picker 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 2-29](#), 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 2-29](#).
- 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, page 2-34](#)
- [Customer Sites, page 2-35](#)
- [Customer Devices, page 2-37](#)

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, page 2-34](#)
- [Editing a Customer, page 2-35](#)
- [Deleting Customers, page 2-35](#)

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 **Enable Site Of Origin** check box if you want this enabled.

**Note**

If **Enable Site Of Origin** check box is checked, the BGP extended community Site-of-Origin (SOO) provisioning in fulfillment MPLS service will be 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, page 2-36](#)
- [Editing a Customer Site, page 2-36](#)
- [Deleting Customer Sites, page 2-37](#)

Creating a Customer Site

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

To create 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:
- Click the **Select** button next to the Customer field.
A list of customer names appears.
 - Click the radio button next to customer name and then **Select**.
 - In the **Site Of Origin Pool** field, either use the **Select** button to select the existing site of origin pool name or enter a specific site of origin value in the form of X:Y. For example 2:214, where X is the Provider's BGP AS number in the range of 1-65535 and Y is the site of origin value in the range of (0-2147483646). The **Site Of Origin Pool** field is available only if the **Enable Site of Origin** check box is enabled in the Customer Details tab at the time of creating a customer.
- 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** Modify the Site Of Origin Pool with available pool values or with other than pool value, if required.
- Step 6** 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 2-38](#)
- [Edit CPE Device, page 2-39](#)
- [Delete CPE Device, page 2-39](#)

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 Provisioning. The CE must be reachable from an Prime Provisioning server.
 - **Unmanaged**—An unmanaged CE cannot be provisioned directly by the provider. If Unmanaged is selected, the provider can use Prime Provisioning 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 Provisioning 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.



Note Unmanaged devices are subject to read-only based operations, but will not be interacted with for ready-write based operations.

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, page 2-40](#)
- [Interface Access Domains, page 2-41](#)
- [Resource Pools, page 2-43](#)
- [Route Targets, page 2-52](#)

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, page 2-40](#)
- [Editing Access Domains, page 2-40](#)
- [Deleting Access Domains, page 2-41](#)

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:

- [Creating Interface Access Domains, page 2-41](#)
- [Editing Interface Access Domains, page 2-42](#)
- [Deleting Interface Access Domains, page 2-42](#)



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

Prime Provisioning 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 Region, VPN, and Customer pools. 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 Prime Provisioning 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 of 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.
- **BVI pool:** BVI pools are defined with a starting value and a size of the BVI pool. A given BVI ID pool can be attached to an interface access domain. During the deployment of an Ethernet Service (EWS, ERS for example), BVI ID can be auto-allocated from the interface access domain's BVI pools. This gives the Service Provider a tighter control of BVI 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 2-44](#)
- [Creating an IPv6 Address Pool, page 2-45](#)
- [Creating a Multicast Pool, page 2-45](#)
- [Creating a Route Distinguisher and Route Target Pool, page 2-46](#)
- [Creating a Site of Origin Pool, page 2-48](#)
- [Creating a VC ID Pool, page 2-49](#)
- [Creating a VLAN Pool, page 2-49](#)
- [Creating an EVC Outer VLAN Pool, page 2-50](#)
- [Creating an EVC Inner VLAN Pool, page 2-50](#)
- [Deleting Resource Pools, page 2-51](#)

Creating an IP Address Pool

Prime Provisioning uses IP address pools to automatically assign IP addresses to PEs and CEs. Each Region, VPN, and Customer has an IP address pool to use for IP numbered addresses (/30 and /31 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**, **31** and **32**
 where:
30 and **31** are used for IP numbered address pools (/30 and /31)
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 an IPv6 Address Pool

Prime Provisioning uses IPv6 address pool to automatically assign IPv6 addresses to PEs and CEs. Each Region has an IPv6 address pool to use for IPv6 numbered addresses (/64, /126 and /127 pools).

From the Create New IPv6 Address Resource Pool window, you can create IPv6 address pool.

To create an IPv6 address pool, follow these steps:

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

The Resource Pools window appears.

Step 2 Select **IPv6 Address** from the **Pool Type** in the upper left of the Resource Pools window.

Step 3 Click the **Create** button.

The Create New IPv6 Address Resource Pool window appears.

The Create New IPv6 Address Resource Pool window contains the following fields:

- **IPv6 Address Pool** (required)—IPv6 Address/Mask, text field in the format a:b:c:d:e:f:g:h/i, for example 2003:adbc:99ff:5623:9023::/120.
- **Pool Mask (bits)** (required)—Choices include: 64, 126, and 127.
- **Pool Association** (required)—Choice: only **Region** from the drop-down list.

Provides option to select a Region and associate IPv6 Pool to a particular Region. Select **Region** from the drop-down list, then click the **Select** button. From this new window, select the particular Region, and then click **Select**.

Step 4 Enter the required information for the IPv6 address pool you are creating.

Step 5 Click **Save**.

The Resource Pools window reappears with the new IPv6 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)



Note

When a new VRF object is created using Autopick RD option or when a service request is created using VPN, Prime Provisioning picks the RD start value of the provider and reduces RD pool size value by one. The current limitation is that this value never returns back to the pool even if the corresponding VRF or service request gets deleted.

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 Provisioning 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 Select 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.
-

Creating an EVC Inner VLAN Pool

From the Create EVC Inner VLAN Pool window, you can create EVC INNER VLAN pools.
To create an INNER VLAN pool, follow these steps:

-
- Step 1** Choose **Service Design > Resources > Resource Pools**.
The Resource Pools window appears.
- Step 2** Select **EVC INNER VLAN** from the **Pool Type** in the upper left of the Resource Pools window.
- Step 3** Click the **Create** button.
The Create New INNER VLAN Resource Pool window appears.

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

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

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

Step 5 Click the **Select** button.

The Select Interface Access Domain for new **INNER VLAN** Pool window appears.

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

Step 7 Click **Save**.

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

Creating a BVI Pool

From the Create New BVI Resource Pool window, you can create BVI resource pools.

To create a BVI pool, follow these steps:

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

The Resource Pools window appears.

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

Step 3 Click the **Create** button.

The Create New BVI Resource Pool window appears.

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

- **BVI Pool Start** (required)— Range: 1 to 65535.
- **BVI Pool Size** (required)—Range: 1 to 65535.
- **Interface Access Domain** (required)

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

Step 5 Click the **Select** button.

The Select Interface Access Domain for new **BVI** Pool window appears.

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

Step 7 Click **Save**.

The **BVI** Pools window reappears with the new **BVI** 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.

Supporting Manual Allocation of IPv6 Address from IPv6 Address Pool

From this release, you can enter an IPv6 address manually in the IP Address Scheme window of Cisco Prime Provisioning (CPP) while creating an MPLS service request. The IPv6 address entered must be available in the IPv6 address pools. If the IPv6 address belongs to a valid IPv6 block, then the IPv6 address is blocked as allocated and is updated in the respective IPv6 pool. The IPv6 address pool is then split as valid IPv6 address blocks or pools.



Note

The IPv6 value is validated against the allocated pool and an error is displayed if the specified address belongs to allocated block.

The IPv6 value is validated against the available pool for a matching IPv6 subnet mask and an error is displayed must be reported if not matched.

The IPv6 value is validated against the available pool and if it belongs to a valid IPv6 block, the block is updated with specified IPv6 address block as allocated

While entering the IPv6 address, ensure that you remember the following conditions:

1. The subnet mask that you enter should be same as that of the IPv6 pool mask.
2. The IPv6 address mask of the Provider Edge (PE) interface and the Customer Edge (CE) interface must be under the same subnet mask as that of the IPv6 address pool.

Once the allocation is successful, you can view the valid IPv6 address pools with Allocated status in the Resource Pools window of CPP.

To remove the MPLS service request, you need to release the IPv6 address back to the available pool. The pools that are split as blocks are combined again into a single pool. For this, you can either decommission the service or perform force delete of the IPv6 allocated block or pool.

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. Prime Provisioning 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 Provisioning 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 Provisioning 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 Provisioning 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 2-53](#)
- [Deleting Route Targets, page 2-54](#)

Creating Route Targets

When you create a VPN, the Prime Provisioning 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, Prime Provisioning 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 Prime Provisioning 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 Provisioning 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 Prime Provisioning 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 2-55](#)
- [Deleting VPNs, page 2-57](#)

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:
- Name** (required)—Enter the name of the VPN, any name of your choice.
 - Customer** (required)—To select the customer associated with this VPN, choose **Select**.
 - From the list of customers, select the appropriate customer, then click **Select**.
 - If you want MPLS attributes, complete the fields in the MPLS Attributes section of the window. For VPLS, skip to step **aa**.
 - Create Default Route Targets** (optional)—To create a default Route Targets, check the **Create Default Route Targets** check box and select a provider.
 - 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.
 - 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 Provisioning. 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. **Import RT List**— Enter one or more Route Targets (RTs) to be imported in the VPN.
- x. For multiple RTs, use a comma (,) separated list. An example RT list is 100:120,100:130,100:140.
- y. **Export RT List**— Enter one or more Route Targets (RTs) to be exported from the VPN.
- z. For multiple RTs, use a comma (,) separated list.
- aa. If you want VPLS attributes, the optional fields for that are in **ab.** to **ae.**
- ab. **Enable VPLS** (optional)—Check this check box to enable VPLS.
- ac. **VPLS VPN ID** (optional)—Enter an integer in the range of 1 to 2147483646.
- ad. **Service Type** (optional)—Click the drop-down list and choose from **ERS** (Ethernet Relay Service) or **EWS** (Ethernet Wire Service).
- ae. **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.



Managing Ethernet Virtual Circuit (EVC) Services

This chapter describes how to use Prime Provisioning policies and service requests to manage various Ethernet Virtual Circuit services. It contains the following sections:

- [Setting Up the Prime Provisioning Services, page 3-7](#)
- [Creating an EVC Ethernet Policy, page 3-19](#)
- [Managing an EVC Ethernet Service Request, page 3-22](#)
- [Creating an EVC ATM-Ethernet Interworking Policy, page 3-32](#)
- [Customizing EVC and MPLS Policies, page 3-35](#)
- [Managing an EVC ATM-Ethernet Interworking Service Request, page 3-35](#)
- [Deploying, Monitoring, and Auditing Service Requests, page 3-52](#)
- [Provisioning VPLS Autodiscovery on Devices using EVC Service Requests, page 3-53](#)
- [Policy and Service Request Attributes Reference Tables, page 3-57](#)
- [Sample Configlets, page 3-122](#)

Getting Started

This section provides a road map to help you get started using the EVC component in Cisco Prime Provisioning 7.2. It contains the following sections:

- [Overview, page 3-2](#)
- [Prepopulating a Service by Selecting Endpoints in Prime Network, page 3-2](#)
- [Installing Prime Provisioning and Configuring the Network, page 3-3](#)
- [Configuring the Network to Support Layer 2 Services, page 3-3](#)
- [Setting Up Basic Prime Provisioning Services, page 3-3](#)
- [Working with EVC Policies and Service Requests, page 3-5](#)
- [A Note on Terminology Conventions, page 3-6](#)

Overview

Before you can use the EVC 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 Provisioning. The following subsections provide a summary of the key tasks you must accomplish to be able to provision EVC services using Prime Provisioning. 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 Provisioning documentation set are provided. See the referenced documentation for more detailed information. After the basic installation and configuration steps are completed for Prime Provisioning, see the subsequent sections to create and provision EVC services.

Prepopulating a Service by Selecting Endpoints in Prime Network

It is possible to create service by picking endpoints on a map in Prime Network Vision, when Prime Provisioning and Prime Network are integrated with Prime Central.

-
- | | |
|---------------|---|
| Step 1 | On any map, select one or more endpoint devices by using CTRL click. |
| Step 2 | In the right click menu, select Fulfill/Create Service . |
| Step 3 | You will be taken to the same first screen as you see when creating a service in Prime Provisioning. |
| Step 4 | Pick a policy.
Depending on the number of endpoints selected, not all policies will work. |
| Step 5 | Once you have selected the policy, the service request main page will appear as usual, prepopulated with links and with the selected devices. |
-

Installing Prime Provisioning and Configuring the Network

Before you can use the EVC module in Prime Provisioning to provision EVC services, you must first install Prime Provisioning and do the basic network configuration required to support Prime Provisioning. Details on these steps are provided in [Chapter 2, “Before Setting Up Prime Provisioning.”](#) See that chapter for information about Prime Provisioning installation and general network configuration requirements.

Configuring the Network to Support Layer 2 Services

In addition to basic network configuration required for Prime Provisioning, you must perform the following network configuration steps to support Layer 2 services. Information on doing these steps is not provided in the Prime Provisioning 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 Provisioning Services

After the basic network configuration tasks are completed to support Prime Provisioning and L2 services, you use Prime Provisioning to define elements in the Prime Provisioning 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 Provisioning tasks are covered in [Chapter 2, “Before Setting Up Prime Provisioning.”](#) You can also find a summary of some important Prime Provisioning set up tasks in [Setting Up the Prime Provisioning Services, page 3-7](#). The information below is a checklist of basic Prime Provisioning 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 Provisioning repository. These are global resources that can be used by all Prime Provisioning 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 2-39](#). See also [Defining a Service Provider and Its Regions, page 3-9](#).
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 2-39](#). See also [Defining Customers and Their Sites, page 3-9](#).

3. **Import or add raw devices.** Every network element that Prime Provisioning manages must be defined as a device in the Prime Provisioning repository. An element is any device from which Prime Provisioning can collect information. In most cases, devices are Cisco IOS routers and switches. It is recommended that you discover and import devices via Prime Network. However, you can also set up devices in Prime Provisioning manually or by importing device configuration files.
4. **Assign devices roles as PE or CE.** After devices are created in Prime Provisioning, 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 Provisioning 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 Provisioning, 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 3-4](#).

Setting Up Prime Provisioning Resources for EVC Services

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

1. **Create access domain(s).** For EVC services, you create an access domain if you provision an Ethernet-based service and want Prime Provisioning 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 Provisioning. 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 2-39](#). See also [Creating Access Domains, page 3-10](#).



Note

Creating the access domain is not mandatory. The Access Domain needs to be created in Prime Provisioning only when VLAN resources are managed through Prime Provisioning.

2. **Create VLAN pool(s).** A VLAN pool is created for each access domain. For EVC services, you create a VLAN pool so that Prime Provisioning 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 2-39](#). See also [Creating VLAN Pools, page 3-10](#).
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 2-39](#). See also [Creating a VC ID Pool, page 3-11](#).
4. **Create interface access domain:** For EVC services, you create an Interface Access Domain if you provision an Ethernet-based service and want Prime Provisioning to automatically assign an EVC Outer VLAN for the link from the Outer VLAN resource pool. For each Layer 2 Interface Access Domain, you need a corresponding Interface Access Domain object in Prime Provisioning. During creation, select all the interfaces of a N-PE device that are associated with this interface access domain. At a later time, one EVC Outer VLAN pool can be created for this domain.

Setting Up NPCs

Before creating an EVC service request, you must predefine the physical links 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 EVC service requests. For detailed steps to create NPCs, see [Setting Up Logical Inventory, page 2-55](#). See also [Creating Named Physical Circuits, page 3-12](#).

Setting Up VPNs

You must define VPNs before provisioning EVC services. Normally for EVC services, one VPN can be shared by different service types but for EVC-VPLS, one VPN is required for each VPLS instance. To define VPNs, see [Setting Up Logical Inventory, page 2-55](#). See also [Defining VPNs, page 3-9](#).

Working with EVC Policies and Service Requests

After you have set up providers, customers, devices, and resources in Prime Provisioning, you are ready to create EVC 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.



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.

1. **Review overview information about L2 services concepts.** See the chapter “Prime Provisioning Layer 2 VPN Concepts” in the [Cisco Prime Provisioning 7.2 Administration Guide](#).
 2. **Set up an EVC policy.** See the appropriate section, depending on the type of policy you want to create:
 - [Creating an EVC Ethernet Policy, page 3-19](#)
 - [Creating an EVC ATM-Ethernet Interworking Policy, page 3-32](#)
 3. **Provision the EVC service request.** See the appropriate section, depending on the type service request you want to provision:
 - [Managing an EVC Ethernet Service Request, page 3-22](#)
 - [Managing an EVC ATM-Ethernet Interworking Service Request, page 3-35](#)
 4. **Deploy the service request.** See [Deploying, Monitoring, and Auditing Service Requests, page 3-52](#).
 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 3-52](#).
 - Audit service requests. See [Deploying, Monitoring, and Auditing Service Requests, page 3-52](#).
-

A Note on Terminology Conventions

The Prime Provisioning 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 Provisioning Layer 2 VPN Concepts,” in the [Cisco Prime Provisioning 7.2 Administration Guide](#), 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 Provisioning Services

To create 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 Provisioning 7.2 resources. It contains the following sections:

- [Creating Target Devices and Assigning Roles \(N-PE or U-PE\), page 3-7](#)
- [Configuring Device Settings to Support Prime Provisioning, page 3-7](#)
- [Defining a Service Provider and Its Regions, page 3-9](#)
- [Defining Customers and Their Sites, page 3-9](#)
- [Defining VPNs, page 3-9](#)
- [Creating Access Domains, page 3-10](#)
- [Creating VLAN Pools, page 3-10](#)
- [Creating Inner and Outer VLAN Pools, page 3-11](#)
- [Creating a VC ID Pool, page 3-11](#)
- [Creating Named Physical Circuits, page 3-12](#)
- [Creating and Modifying Pseudowire Classes, page 3-15](#)

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

Every network element that Prime Provisioning manages must be defined as a device in the system. An element is any device from which Prime Provisioning 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 Provisioning

Two device settings must be configured to support the use of Prime Provisioning 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 Provisioning 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 Provisioning 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 > 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 **LDP Termination 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 Provisioning 7.2, supports devices running Cisco's IOS XR software. 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 Provisioning as an IOS XR device, as follows:
- Create the Cisco device by choosing **Inventory > Devices > Create Cisco Device**.
 - Choose **Cisco Device** in the drop-down list.
The Create Cisco Router window appears.
 - 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 G, “Property Settings”](#) or see instructions in the *Cisco Prime Provisioning 7.2 Administration Guide*.
-
- 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 3-122](#).

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 2-39](#).

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 2-39](#).

Defining 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. For detailed steps to create VPNs, see [Setting Up Logical Inventory, page 2-55](#).

**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 Provisioning 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 Provisioning. 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 2-39](#).

Creating VLAN Pools

For L2VPN and VPLS, you create a VLAN pool so that Prime Provisioning 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 Provisioning 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 Provisioning service, Prime Provisioning warns you if the VLAN ID is outside the valid range of the defined VLAN pool. If so, Prime Provisioning 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 Provisioning automatically assign a VLAN to the links, perform the following steps.

-
- Step 1** Choose **Service Design > 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.
- Choose an Access Domain Name by clicking the button in the Select column to the left of that Access Domain.
 - 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 3-10](#).) 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 Inner and Outer VLAN Pools

An inner and outer VLAN pools are used in conjunction with the AutoPick Inner VLAN and AutoPick Outer VLAN attributes in EVC Ethernet and EVC ATM-Ethernet policies and services. For instructions on how to set up inner and outer VLAN pools, see the section [Resource Pools, page 2-43](#).

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 EVC 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 Provisioning service, Prime Provisioning warns you if the VC ID is outside the valid range of the defined VC ID pool. If so, Prime Provisioning 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 EVC services, perform the following steps.

Step 1 Choose **Service Design > 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 EVC 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 EVC 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 3-13](#).

- Through the autodiscovery process. For details on how to do this, see [Creating NPC Links Through the Autodiscovery Process, page 3-15](#).

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 > 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 2-55](#).

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 > 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 Provisioning 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 Provisioning, see [Appendix B, “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 Provisioning database. NPCs are further abstracted from the discovered connectivity.

For detailed steps to create NPCs using autodiscovery, see [Setting Up Logical Inventory, page 2-55](#).

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.



Note



The pseudowire class feature is supported on both IOS and IOS XR devices. For IOS XR devices, the pseudowire class feature is supported on IOS XR version 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 Provisioning 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. The pseudowire class is a separately defined object in the Prime Provisioning repository that can be attached to an L2VPN service policy or service request.

This section describes how to create and modify pseudowire classes. For information on how the pseudowire class is used in policies and service requests, see later sections of this guide on setting attributes for specific services.

Creating a Pseudowire Class

To create a pseudowire class, perform the following steps.

-
- Step 1** Choose **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 or IOS XR device.
 - Follow the below naming conventions while entering the PseudoWireClass name:
 - The name should not exceed 32 characters.
 - It should not contain spaces.
 - It should not contain any special characters except underscore.
- Step 4** In the **Description** field, enter a meaningful description of less than 128 characters.
This field is optional.
- Step 5** Check the **Control Word** check box to enable dynamic pseudowire connection.
- Step 6** Choose the **MPLS** encapsulation type from the **Encapsulation** drop-down list.
-  **Note** Currently, the only encapsulation type supported is MPLS.
-
- Step 7** 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 or IOS XR being used. If pseudowire class is not supported in a particular version of IOS or 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 3-18](#).
-
- Step 8** 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 9** To configure 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 10** To determine the MPLS-TP or MPLS-TE tunnel path used by the pseudowire, click the **Tunnel Type** radio button.

- **NONE** (default)—User need not provide the Tunnel ID value.
- **TE**—User need to provide the Tunnel ID value.
- **TP**—User need to provide the Tunnel ID value.

Step 11 Enter a **Tunnel ID** of a TE tunnel that has already been provisioned by Prime Provisioning 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 Provisioning, as covered in the next step.

Step 12 Click **Select TE Tunnel** if you want to select a TE tunnel that has been previously provisioned by Prime Provisioning.

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 13 Check the **Disable Fallback** check box to disable the fallback option for the pseudowire tunnel.

Choose this option based on your version of IOS or IOS XR. It is required for IOS XR 3.6.1 and optional for IOS XR 3.7 and above.

Modifying a Pseudowire Class

To modify (edit) a pseudowire class, perform the following steps.

Step 1 Choose **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 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, perform the following steps.



Note A Pseudowire Class that is in use with a service request or policy cannot be deleted.

- Step 1** Choose **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 or 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 Provisioning, navigate to **Administration > 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 Provisioning 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 or 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 override 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 Provisioning, navigate to **Administration > 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 Provisioning 7.2, as well as the basic steps to create an EVC Ethernet policy. It contains the following subsections:

- [Overview, page 3-20](#)
- [Defining the EVC Ethernet Policy, page 3-20](#)

**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.

Overview

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 Provisioning templates and data files with a policy. See [Chapter 11, “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 D, “Adding Additional Information to Services.”](#)

For information on creating EVC Ethernet service requests, see [Managing an EVC Ethernet Service Request, page 3-22](#).

**Note**

For a general overview of EVC support in Prime Provisioning, see the chapter “Layer 2 Concepts” in the [Cisco Prime Provisioning 7.2 Administration Guide](#).

Defining the EVC Ethernet Policy

To define an EVC Ethernet policy, perform the following steps.

-
- Step 1** Choose **Service Design > Create Policy**.
The Policy Editor window appears.
- Step 2** Choose **EVC** from the Policy Type drop-down list.
The Policy Editor window appears.
- Step 3** Enter a **Policy Name** for the EVC policy.
- Step 4** 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 Provisioning 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 5 Click **Select** to choose the owner of the EVC policy.

The policy owner was established when you created customers or providers during Prime Provisioning setup. If the ownership is global, the Select function does not appear.

Step 6 Choose the **Policy Type**.

The choices are:

- **ETHERNET**—This section.
- **ATM**—See [Creating an ATM Policy, page 5-4](#).
- **ATM Ethernet Interworking**—See [Creating an EVC ATM-Ethernet Interworking Policy, page 3-32](#).
- **TDM Circuit Emulation**—See [Creating a TDM-CEM Policy, page 4-7](#).
- **EVC**—

Step 7 Click **Next**.

The Service Options window appears.

Step 8 Set the attributes in the Service Options window as described in [Service Options Window, page 3-57](#).

Step 9 When you have set the attributes, click **Next**.

The EVC Attributes window appears.

Step 10 Set the attributes in the EVC Attributes window as described in [EVC Attributes Window, page 3-62](#).

Step 11 When you have set the attributes, click **Next**.

The Interface Attributes window appears.

Step 12 Set the attributes in the Interface Attributes window as described in [Interface Attributes Window, page 3-68](#).

Step 13 When you have set the attributes, click **Next** to proceed to the next window (or else click **Finish** to save the policy).

Step 14 If you would like to use user-defined attributes within this policy, click **Next** (before clicking **Finish**).

An additional window appears the policy workflow. 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 D, “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 15 If you would like to enable template association for this policy, click **Next** (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 [Chapter 11, “Managing Templates and Data Files”](#) for more information about using templates and data files. 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 16 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 3-22](#).

Managing an EVC Ethernet Service Request

This section provides information on how to provision an EVC Ethernet service request. It contains the following subsections:

- [Overview, page 3-22](#)
- [Creating an EVC Service Request, page 3-23](#)
- [Using Templates and Data Files with an EVC Ethernet Service Request, page 3-31](#)
- [Saving the EVC Ethernet Service Request, page 3-31](#)
- [Modifying the EVC Ethernet Service Request, page 3-31](#)
- [Deploying the EVC Ethernet Service Request, page 3-32](#)

Overview

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 3-19](#). 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:

1. Choose an existing EVC Ethernet policy.
2. 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.

3. Specify a bridge domain configuration (if applicable).
4. Specify a service request description.
5. Specify automatic or manual allocation of the VC ID or VPLS VPN ID.
6. Add direct connect links (if applicable).
7. Add links with L2 access nodes (if applicable).
8. Choose the N-PE and UNI interface for links.
9. 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.
10. Edit the link attributes.

11. Modify the service request.
12. Save the service request. After Save, select the SR id from the SR Manager page and perform Preview Deploy. Verify the configlets for its correctness, deploy it by choosing Deploy > Deploy now option. The SR state will be displayed as Deployed state

For sample configlets for EVC Ethernet scenarios, see [Sample Configlets, page 3-122](#).

Creating an EVC Service Request

To create an EVC Ethernet service request, perform the following steps.

-
- Step 1** Choose **Operate > Service Request Manager**.
- The Service Request Manager window appears.
- Step 2** Click **Create**.
- The Service Request Editor window appears.
- Step 3** From the policy picker, choose an EVC policy from the policies previously created (see [Creating an EVC Ethernet Policy, page 3-19](#)).
- The EVS Service Request editor window appears. This window enables you to specify options for the service request, as well as configure links. 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).
- Step 4** Set link attributes based on the MPLS Core Connectivity Type for the policy:
- [Table 3-7, “Pseudowire Core Connectivity Attributes,” on page 75](#)
 - [Table 3-8, “VPLS Core Connectivity Attributes,” on page 77](#)
 - [Table 3-9, “Local Core Connectivity Attributes,” on page 80](#)
- Step 5** Set up links to the N-PE as described in section [Setting up Links to the N-PE, page 3-24](#).
- The following link types are covered:
- [Setting Direct Connect Links, page 3-24](#)
 - [Setting Links with L2 Access Nodes, page 3-26](#)
 - [Using Templates and Data Files with an EVC Ethernet Service Request, page 3-31](#)
- After you have set up links, return to this section and perform the following steps to finishing creating the service request.
- Step 6** If you are using templates and data files with the service request, follow the guidelines in section [Using Templates and Data Files with an EVC Ethernet Service Request, page 3-31](#).
- Step 7** When you have completed setting the attributes in the EVC Service Request Editor window, click the **Save** button to save the settings and create the EVC service request.
- If any attributes are missing or incorrectly set, Prime Provisioning displays a warning in the lower left of the window. Make any corrections or updates needed (based on the information provided by Prime Provisioning), and click the **Save** button.
- Step 8** If you are ready to deploy the EVC Ethernet service request, see [Deploying Service Requests, page 10-9](#).
-

For additional information on working with EVC service requests, see the following sections:

- [Using Templates and Data Files with an EVC Ethernet Service Request, page 3-31.](#)
- [Saving the EVC Ethernet Service Request, page 3-31.](#)
- [Modifying the EVC Ethernet Service Request, page 3-31](#)
- [Deploying the EVC Ethernet Service Request, page 3-32.](#)

Setting up Links to the N-PE

The lower two sections of the EVC Service Request Editor window allow you to set up and configure links to the N-PE(s). See the appropriate section, depending on which type of link you are setting up:

- [Setting Direct Connect Links, page 3-24](#)
- [Setting Links with L2 Access Nodes, page 3-26](#)
- [Configuring Multi-segment Pseudowires, page 3-27](#)
- [Setting Up Pseudowire Redundancy and a Backup Peers, page 3-29](#)
- [Setting VPLS Neighbor Links \(VPLS only\), page 3-30](#)



Note

Many of the steps for setting up the 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 3-24](#). 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

For direct connect links, the CE is directly connected to the N-PE, with no intermediate L2 access nodes. No NPC are involved.

To set up the direct connect links, perform the following steps.

-
- Step 1** Click **Add** to add a link.
A new numbered row for the link attributes appears.
- Step 2** To select the PE device for the link, click the toggle button in the **Select Device** field in N-PE column.
The Device Selection window appears. This window displays the list of currently defined PEs, including Device Name, Provider Name, and PE Region Name for each device. The Quick Filter option allows you to type in strings in filter fields to narrow the list of devices.
- Step 3** Choose the PE device for the link by clicking the radio button next to the device name.
The EVC Service Request Editor window reappears displaying the name of the selected PE in the N-PE column.
- Step 4** To choose the UNI interface, click on the toggle button in the **Select One** field of the UNI column.
The Direct Link Interface Selection window appears. This window 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.
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 5** Choose the UNI interface by clicking the radio button next to the interface name.
- Step 6** Check the **EVC** check box to mark the link for configuring service instance for the links.

- 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.
- 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 Provisioning\ProvDrv\CheckFlexUniCheckBox.

Step 7 Click **Edit** in the Link Attributes column to specify the UNI 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: Service Instance Details and Standard UNI Details. If the EVC check box for the link is not checked, only the Standard UNI Details window is presented.

Step 8 If applicable, set the attributes in the Service Instance Details window as described in [Table 3-10](#).



Note

All of the fields in the Service Instance Details screen are enabled based on the policy settings.

Step 9 Click **Next** to save the settings in the Service Instance Details window.

The Standard UNI Details window appears.

Step 10 If applicable, set the standard UNI link attributes as described in [Table 3-11](#).

- 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.
- The attributes that appear in the Standard UNI Details window are dynamically configured by Prime Provisioning. Some of the attributes 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 [Table 3-11](#).

Step 11 Click **OK** to save the Standard UNI settings and return to the EVC SR window.

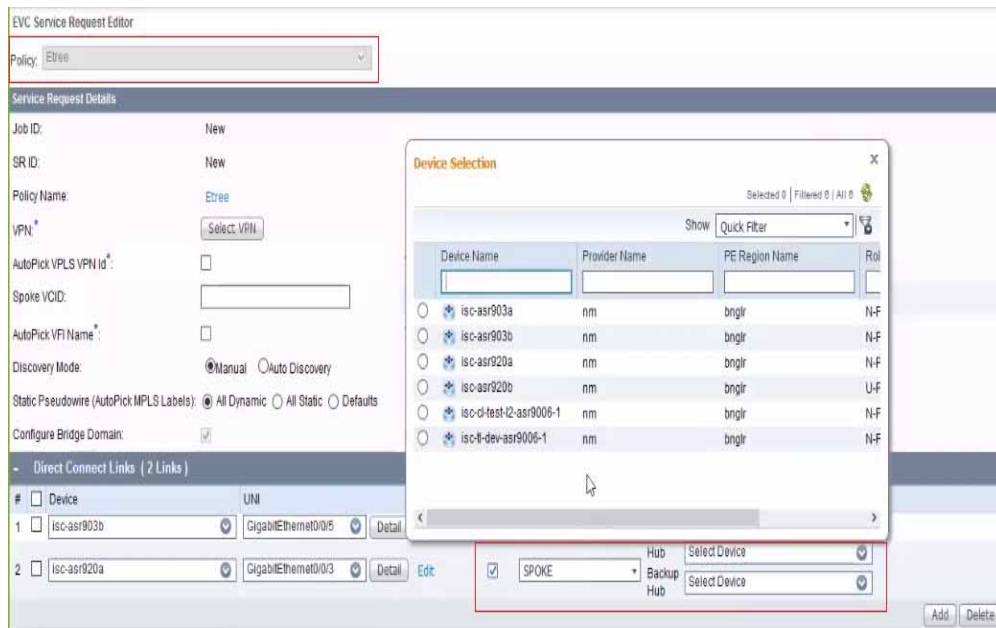
The value in the Link Attributes column now displays as “Changed,” signifying that the link settings have been updated. You can edit the link attributes now or at a future time by clicking on the Changed link and modifying the settings in the Standard UNI Details window.

Step 12 To add another link click the **Add** button and set the attributes for the new link as in the previous steps in this section.

If are creating an EVC service request for either HVPLS or Etree policies and if you want to select devices:

- In the Direct Connect Links area, Click **Add** to Select **Device** and **UNI**.
- Check the **Edit** check box, and then choose the SPOKE option. The **Hub** and **Backup Hub** drop-down list boxes appears. You can use toggle picker and choose the Hub device names using the quick filters in the **Device Selection** window as shown in the [Figure 3-1](#).

Figure 3-1 Device picker with filter option for Hub and Backup Hub



- Step 13** To delete a link, check the check box in the first column of the row for that link and click the **Delete** button.
- Step 14** To complete the EVC Ethernet service request, see steps presented in [Creating an EVC Service Request](#), page 3-23.

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 3-24. 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 than 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 3-24](#).

The following points cover the use of the EVC (UNI) check box:

- The EVC (UNI) attribute is equivalent to the All L2 Access Links Default to EVC UNI attribute in the policy. When you enable the attribute in the policy, it is enabled in the service request.
- The EVC (UNI) attribute only appears in the Links with L2 Access Nodes section of the EVC Service Request Editor window, in which you may have “n” number of U-PE and PE-AGG devices using the link. The Direct Connect Links section does not have this check box because EVC syntax is supported by default on N-PE devices of direct connect links.
- An NPC link must be available on a U-PE or PE-AGG device/interface in order to use this feature.
- This feature is only supported with IOS running on the U-PE or PE-AGG device. IOS XR is not supported.
- When the EVC (UNI) check box is enabled and you click the Edit link, the Service Instance Details window appears. The EVC syntax-related attributes appear for the U-PE device as well as the N-PE device. The optimum number of attributes appear within the U-PE section. Attributes set in the U-PE section are not repeated in the N-PE section. Note that any VLAN matching criteria for the U-PE side are matched on the N-PE side also.
- For descriptions of attributes that appear in GUI when the EVC (UNI) check box is enabled, see [Table 3-10](#).

Step 4 To complete the EVC Ethernet service request, see steps presented in [Creating an EVC Service Request, page 3-23](#).

Configuring Multi-segment Pseudowires

This section describes how pseudowire classes may be used to configure multi-segment pseudowires in Prime Provisioning. This enables you to create and independently assign pseudowire classes at the endpoints of a multi-segment pseudowires. You can perform all of the configuration steps in EVC Service Request Editor window, as described in the steps below. Alternatively, you can create pseudowire classes independently of the EVC service request, and then as they are deployed on the device you can reuse them. This feature is available with EVC Ethernet service requests using MPLS core connectivity types of PSEUDOWIRE and VPLS. It is also available for EVC ATM and EVC TDM Circuit Emulation service requests.



Note

To use this feature, the attribute Configure Pseudowire Segment(s) must be enabled in the policy for the service request. For more information about this attribute, see the appropriate tables in the section [EVC Ethernet Policy Attributes, page 3-57](#).

The following steps provide an example showing the basic steps of how to configure multi-segment pseudowires.

Step 1 Navigate to the EVC Service Request Editor window.

Step 2 In the Direct Connect Links section of the window, add the N-PE devices between which you want to configure the pseudowire.

In the EVC Pseudowires section of the window an EVC pseudowire appears in the Pseudowire column.

Step 3 To configure the pseudowire, click the **Configure Pseudowire** link in the Pseudowire Configuration column.

A dialog window appears in which you can view and configure the pseudowire. The window has four tabs:

- Calculated Path
- Path Summary
- Segment Configuration
- Create Pseudowire Class

It also provides a drop-down menu (in the lower left of the window) in which you can choose a (required) tunnel for the link.



Note

Internet Explorer 8 (IE8) will not show the calculated path graphically (as described in Creating an MPLS-TP Service Request) as IE8 offers no support for SVG display. However, a textual summary of the path can be used to review the path in IE8. IE9 (and other Prime Provisioning supported browsers) shows the calculated path graphically.

Step 4 In the drop-down menu, choose one tunnel to be the required tunnel for the link.

Note that you can 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 the path.
- **Excluded NE/Link**—Specify network elements or links that traffic must pass through for the path.

Step 5 Click the **Calculate** button to re-calculate the path.

To do this, enter the path constraints then click **Calculate** to re-calculate the path. Once the path is decided, you can use the other tabs to configure it.

Step 6 Click on the **Calculated Path** tab to view a diagram of the link.

This displays a path diagram using the shortest path between the previously selected N-PEs.

Step 7 Click on the **Path Summary** tab for a textual representation of the path.

This can be used if the browser does not support the technology required to show the graphical path.

Step 8 Click on the **Segment Configuration** tab to set configuration options on a per-segment basis.

Perform the following steps:

- a. Check the radio button of the segment you want to configure.
- b. Use the **Pseudowire Class** drop-down menus to associate pseudowire classes to each end of the segment. The pseudowire classes must already exist. In order for the pseudowire classes to be valid, they must match up with the same core type and same tunnel number. Otherwise, you will not be able to choose the pseudowire class. In that case, you can leave the Pseudowire Class drop-down menu blank and if needed Prime Provisioning will autogenerate a pseudowire class with an autogenerated name on the device.

If the pseudowire class does not exist and you would like to create one, you can create it in-line using the Create Pseudowire Class tab as covered below.

- c. The **Segment Type** field is not selectable, but is autogenerated. Depending on the type of segment, this field displays one of the following: TP Tunnel, TE Tunnel, or LDP.
- d. Use the **MPLS Labels** drop-down to configure static or dynamic labels. This setting overrides the global settings, that is the value of the Static Pseudowire (AutoPick MPLS Labels) attribute previously set in the policy or service request.

- e. Once the configuration is set up on a segment, click the **Save** button below the segment information to save the settings for the segment.



Note Note that you cannot delete a TE tunnel if it is in use by an EVC service. Therefore, if you configure a multi-segment pseudowire to use a TE tunnel anywhere in the path of the multi-segment pseudowire, it will prevent that TE tunnel from being removed by Prime Provisioning.

- Step 9** If needed, click on the **Create Pseudowire Class** tab to create a pseudowire class in line during the configuration.

A Create Pseudowire Class window appears. The options in the window are similar to the top-level pseudowire creation operation available at **Inventory > Logical Inventory > Pseudowire Class**.

- a. Set the options for the pseudoewire class per your requirements.
- b. Click the **Create** button to create the pseudowire class.

Then you will be able to see and choose the new pseudowire class in the Pseudowire Class drop-down menu in the Segment Configuration tab.

- Step 10** Click the **Revert** button to revert the calculated path.

For example, in the case of a single segment, clicking the Revert button reverts the calculated path to reflect the pseudowire classes that are defined on the individual links. If you never open the Configure Pseudowire dialog, then you can still define pseudowire classes using each of the link attribute editors.

- Step 11** Click the **Save** button to save the configuration settings.

- Step 12** Click the **Close** button to close the dialog.

The dialog closes and you return to the EVC Service Request Editor window.

- Step 13** To complete the EVC Ethernet service request, see steps presented in [Creating an EVC Service Request, page 3-23](#).

Setting Up Pseudowire Redundancy and a Backup Peers

This section describes how to configure pseudowire redundancy and backup peers for EVC Ethernet services with a PSEUDOWIRE core type. This is done by designating links as A, Z, and Z-backup [Z' (prime)] links.



Note You can add two direct connect links and one NPC circuit (derived from a Single Homed Ring) as the L2 access node. In this scenario, the L2 access node acts as the source node for the pseudowire and the direct links are the two distribution nodes (Z and Z' link). The Z' link associated with the direct connect link acts as the backup for the source device present in the L2 access node.

This feature is activated when the Pseudowire Redundancy check box is enabled in the EVC Service Request Editor window.

To configure pseudowire redundancy or set up a backup peer, perform the following steps.

- Step 1** In the EVC Service Request Editor window, check the **Pseudo Wire Redundancy** check box to enable pseudowire redundancy.

- Step 2** Add two N-PE devices in the Direct Connect Links section of the window.

Note that the first N-PE is designated as “A” in the Terminal column, while the second N-PE is marked as “Z”.

You may also configure a third link as a backup peer, as follows.

Step 3 Add a third N-PE into the list of devices in the Direct Connect Links section.

The third N-PE is designated as “Z - Backup” in the Terminal column.

In the EVC Pseudowires section of the window, two pseudowires are listed. One is designated as the backup. Note the pseudowires are now between:

- First and second N-PE (“A” and “Z”), and
- First and third N-PE (“A” and “Z - Backup”)

You can configure the pseudowires by clicking the **Configure Pseudowire** link to the right of the pseudowire name. The steps to do this are similar to those provided in the section [Configuring Multi-segment Pseudowires, page 3-27](#) (preceding section, above).



Note Note that you cannot delete a TE tunnel if it is in use by an EVC service. Therefore, if you configure a multi-segment pseudowire to use a TE tunnel anywhere in the path of the multi-segment pseudowire, this prevents that TE tunnel from being removed by Prime Provisioning.

Setting VPLS Neighbor Links (VPLS only)

If a VPLS policy has been selected, the bottom window will show VPLS Neighbor Links. If you select two or more N-PEs under Direct Connect Links, you will be able to discover any VPLS enabled neighbors.

To choose the desired path in a Multisegment Pseudowire topology, do the following:

Step 1 Configure the pseudowire by clicking the **Configure Pseudowire** link under VPLS Neighbor Links.



Note Pseudowires are configured not just for the links in this service request but for all links in the VPLS.

Step 2 In the pop-up window, click the **Calculate Path** button.

This displays a path diagram using the shortest path between the previously selected N-PEs. Any existing MPLS-TP tunnels between them will be given priority.

Step 3 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 the path.
- **Excluded NE/Link**—Specify network elements or links that traffic must *not* pass through for the path.

Step 4 For additional details on using features in the pop-up window, see the previous section [Configuring Multi-segment Pseudowires, page 3-27](#).

- Step 5** To complete the EVC Ethernet service request, see steps presented in [Creating an EVC Service Request, page 3-23](#).
-

Using Templates and Data Files with an EVC Ethernet Service Request

The template mechanism in Prime Provisioning 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 [Chapter 11, “Managing Templates and Data Files”](#) for more information about using templates and data files.

Saving the EVC Ethernet 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.
-

Modifying the EVC Ethernet 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 Request Manager**.
- The Service Request Manager window appears, showing service requests available in Prime Provisioning.
- 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 starting with “[Creating an EVC Service Request](#)” section on page 3-23 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 [Saving the EVC Ethernet Service Request](#), page 3-31.
- 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 Ethernet Service Request](#), page 3-31.
-

Deploying the EVC Ethernet Service Request

You can deploy an EVC Ethernet service in two different ways:

- If a service request has been saved, you may deploy it through the Service Request Manager window (choose **Operate > Service Request Manager**). For steps on how to do this, see [Chapter 10, “Managing Service Requests.”](#)
- Alternatively, you can deploy an EVC Ethernet service request from within the Service Request Editor window (while creating the service request). The Deploy button at the bottom of the window allows you to save and deploy the service request in one step.

Creating an EVC ATM-Ethernet Interworking Policy

This section contains an overview of EVC ATM-Ethernet Interworking support in Prime Provisioning, as well as the basic steps to create an EVC ATM-Ethernet Interworking policy. It contains the following subsections:

- [Overview](#), page 3-20
- [Defining the EVC Ethernet Policy](#), page 3-20



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.

Overview

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.

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 Provisioning templates and data files with a policy. See [Chapter 11, “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 D, “Adding Additional Information to Services.”](#)

For information on creating EVC ATM-Ethernet service requests, see [Managing an EVC ATM-Ethernet Interworking Service Request, page 3-35.](#)

Defining the EVC ATM-Ethernet Interworking Policy

To define an EVC ATM-Ethernet Interworking policy, perform the following steps.

Step 1 Choose **Service Design > Create Policy**.

The Policy Editor window appears.

Step 2 Choose **EVC** from the Policy Type drop-down list.

The Policy Editor window appears.

Step 3 Enter a **Policy Name** for the EVC ATM-Ethernet Interworking policy.

Step 4 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 Provisioning 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 5 Click **Select** to choose the owner of the EVC policy.

The policy owner was established when you created customers or providers during Prime Provisioning setup. If the ownership is global, the Select function does not appear.

Step 6 Choose the **Policy Type**.

The choices are:

- **ETHERNET**—See [Creating an EVC Ethernet Policy, page 3-19](#).
- **ATM**—See [Creating an ATM Policy, page 5-4](#).
- **ATM Ethernet Interworking**—See [Creating an EVC ATM-Ethernet Interworking Policy, page 3-32](#).
- **TDM Circuit Emulation**—See [Creating a TDM-CEM Policy, page 4-7](#).



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 3-19](#).

Step 7 Click **Next**.

The Service Options window appears.

Step 8 Set the attributes in the Service Options window as described in [Service Options Window, page 3-93](#).

Step 9 When you have set the attributes, click **Next**.

The ATM Interface Attribute window appears.

Step 10 Set the attributes in the ATM Interface Attribute window as described in [ATM Interface Attributes Window, page 3-96](#).

Step 11 When you have set the attributes, click **Next**.

The EVC Attributes window appears.

- Step 12** Set the attributes in the EVC Attributes window as described in [EVC Attributes Window, page 3-96](#).
- Step 13** When you have set the attributes, click **Next**.
The Interface Attribute window appears.
- Step 14** Set the attributes in the Interface Attributes window as described in [Interface Attributes Window, page 3-101](#).
- Step 15** When you have set the attributes, click **Next** to proceed to the next window (or else click **Finish** to save the policy).
- Step 16** If you would like to use user-defined attributes within this policy, click **Next** (before clicking **Finish**).
An additional window appears the policy workflow. 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 D, “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 17** If you would like to enable template association for this policy, click **Next** (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 [Chapter 11, “Managing Templates and Data Files”](#) for more information about using templates and data files. 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 18** To save the EVC ATM-Ethernet Interworking policy, click **Finish**.

To create a service request based on an EVC policy, see [Managing an EVC Ethernet Service Request, page 3-22](#).

Customizing EVC and MPLS Policies

For instructions in how to use this feature, see [Chapter 8, “Customizing EVC, MPLS and MPLS-TP Policies”](#).

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, page 3-36](#)
- [Creating an EVC ATM-Ethernet Interworking Service Request, page 3-36](#)
- [Using Templates and Data Files with an EVC ATM-Interworking Service Request, page 3-41](#)
- [Saving the EVC ATM-Interworking Service Request, page 3-42](#)
- [Modifying the EVC ATM-Interworking Service Request, page 3-42](#)
- [Deploying the EVC ATM-Ethernet Service Request, page 3-43](#)

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 3-32](#). To create an EVC ATM-Ethernet Interworking service request, an EVC ATM-Ethernet Interworking service policy must already be defined. 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:

1. Choose an existing EVC ATM-Ethernet Interworking policy.
2. 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.

3. Specify a bridge domain configuration (if applicable).
4. Specify a service request description.
5. Specify automatic or manual allocation of the VC ID or VPLS VPN ID.
6. Add direct connect links (if applicable).
7. Add links with L2 access nodes (if applicable).
8. Choose the N-PE and UNI interface for links.
9. 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.
10. Edit the link attributes.
11. Modify the service request.
12. Save the service request.

For sample configlets for EVC ATM-Ethernet Interworking scenarios, see [Sample Configlets, page 3-122](#).

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 Request Manager**.
The Service Request Manager window appears.
- Step 2** Click **Create**.
The Service Request Editor window appears.
- Step 3** From the policy picker, choose an EVC ATM-Ethernet Interworking policy from the policies previously created (see [Creating an EVC ATM-Ethernet Interworking Policy, page 3-32](#)).
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** Set link attributes based on the MPLS Core Connectivity Type for the policy as described in the following tables:
- [Table 3-18, “Pseudowire Core Connectivity Attributes,” on page 107.](#)
 - [Table 3-19, “Local Core Connectivity Attributes,” on page 109.](#)
- Step 5** Set up links to the N-PE as described in section [Setting up Links to the N-PE, page 3-37](#).
The following link types are covered:
- [Setting Direct Connect Links, page 3-38](#)
 - [Setting the ATM Link Attributes, page 3-39](#)
 - [Setting Links with L2 Access Nodes, page 3-40](#)
- After you have set up links, return to this section and perform the following steps to finishing creating the service request.
- Step 6** If you are using templates and data files with the service request, follow the guidelines in section [Using Templates and Data Files with an EVC ATM-Interworking Service Request, page 3-41](#).
- Step 7** When you have completed setting the attributes in the EVC Service Request Editor window, click the **Save** button to save the settings and create the EVC service request.
If any attributes are missing or incorrectly set, Prime Provisioning displays a warning in the lower left of the window. Make any corrections or updates needed (based on the information provided by Prime Provisioning), and click the **Save** button.
- Step 8** If you are ready to deploy the EVC Ethernet service request, see [Managing Service Requests, page 10-1](#).
-

For additional information on working with EVC service requests, see the following sections:

- [Using Templates and Data Files with an EVC ATM-Interworking Service Request, page 3-41.](#)
- [Saving the EVC ATM-Interworking Service Request, page 3-42.](#)
- [Modifying the EVC ATM-Interworking Service Request, page 3-42](#)
- [Deploying the EVC ATM-Ethernet Service Request, page 3-43.](#)

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. See the appropriate section, depending on which type of link you are setting up:

- [Setting Direct Connect Links, page 3-38](#). 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. For details on ATM links, see [Setting the ATM Link Attributes, page 3-39](#).
- [Setting Links with L2 Access Nodes, page 3-40](#). The Links with L2 Access Nodes section is where you set up links with L2 (Ethernet) access nodes. NPCs are involved. ATM interfaces cannot be in L2 access nodes.

**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 3-38](#). 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

For direct connect links, the CE is directly connected to the N-PE, with no intermediate L2 access nodes. 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.

To set up the direct connect links, perform the following steps.

-
- Step 1** Click **Add** to add a link.
- A new numbered row for the link attributes appears.
- Step 2** To select the PE device for the link, click the toggle button in the **Select Device** field in N-PE column. The Device Selection window appears. This window displays the list of currently defined PEs, including Device Name, Provider Name, and PE Region Name for each device. The Quick Filter option allows you to type in strings in filter fields to narrow the list of devices.
- Step 3** Choose the PE device for the link by clicking the radio button next to the device name. The EVC Service Request Editor window reappears displaying the name of the selected PE in the N-PE column.
- Step 4** To choose the UNI interface, click on the toggle button in the **Select One** field of the UNI column. The Direct Link Interface Selection window appears. This window 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.
- 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 5** Choose the UNI interface by clicking the radio button next to the interface name.
- Step 6** Check the **EVC check box** to mark the link for configuring service instance for the links.
- 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.
 - 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`.
- Step 7** Click **Edit** in the Link Attributes column to specify UNI 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: Service Instance Details and Standard UNI Details. If the EVC check box for the link is not checked, only the Standard UNI Details window is presented.

**Note**

If you are setting up an 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 3-39](#).

- Step 8** Click **Edit** in the Link Attributes column to specify the UNI attributes.
- Step 9** If applicable, set the attributes in the Service Instance Details window as described in [Table 3-20](#).
- Step 10** Click **Next** to save the settings in the Service Instance Details window.
The Standard UNI Details window appears.
- Step 11** If applicable, set the standard UNI link attributes as described in [Table 3-21](#).
- 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.
 - The attributes that appear in the Standard UNI Details window are dynamically configured by Prime Provisioning. 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 12** Click **OK** to save the Standard UNI settings and return to the EVC Service Request 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.
- Step 13** 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 14** To delete a link, check the check box in the first column of the row for that link and click the **Delete** button.
- Step 15** To complete the EVC ATM-Ethernet Interworking service request, see steps presented in [Creating an EVC ATM-Ethernet Interworking Service Request, page 3-36](#).

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 interface picker 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 Set attributes in the ATM UNI Details window as described in [Table 3-22](#).

Step 5 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 [Using Templates and Data Files with an EVC ATM-Interworking Service Request, page 3-41](#), for details on editing the link attributes.

Step 6 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 7 To delete a link, check the check box in the first column of the row for that link and click the **Delete** button.

Step 8 To complete the EVC ATM-Ethernet Interworking service request, see steps presented in [Creating an EVC ATM-Ethernet Interworking Service Request, page 3-36](#).

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 3-39](#).

The steps for setting up links with L2 access nodes are similar to those covered in the section [Setting Direct Connect Links, page 3-38](#). 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 than 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 3-38](#).
- The following points cover the use of the EVC (UNI) check box:
- The EVC (UNI) attribute is equivalent to the All L2 Access Links Default to EVC UNI attribute in the policy. When you enable the attribute in the policy, it is enabled in the service request.
 - The EVC (UNI) attribute only appears in the Links with L2 Access Nodes section of the EVC Service Request Editor window, in which you may have “n” number of U-PE and PE-AGG devices using the link. The Direct Connect Links section does not have this check box because EVC syntax is supported by default on N-PE devices of direct connect links.
 - An NPC link must be available on a U-PE or PE-AGG device/interface in order to use this feature.
 - This feature is only supported with IOS running on the U-PE or PE-AGG device. IOS XR is not supported.
 - When the EVC (UNI) check box is enabled and you click the Edit link, the Service Instance Details window appears. The EVC syntax-related attributes appear for the U-PE device as well as the N-PE device. The optimum number of attributes appear within the U-PE section. Attributes set in the U-PE section are not repeated in the N-PE section. Note that any VLAN matching criteria for the U-PE side are matched on the N-PE side also.
 - For descriptions of attributes that appear in GUI when the EVC (UNI) check box is enabled, see [Table 3-20](#).
- Step 4** To complete the EVC ATM-Ethernet Interworking service request, see steps presented in [Creating an EVC ATM-Ethernet Interworking Service Request, page 3-36](#).
-

Using Templates and Data Files with an EVC ATM-Interworking Service Request

The template mechanism in Prime Provisioning 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 [Chapter 11, “Managing Templates and Data Files”](#) for more information about using templates and data files.

Saving the EVC ATM-Interworking Service Request


To save an EVC ATM-Interworking service request, perform the following steps.

-
- Step 1** When you have finished setting the attributes for the EVC ATM-Interworking service request, click **Save** to create the service request.
- If the EVC ATM-Interworking 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.
-

Modifying the EVC ATM-Interworking Service Request

You can modify an EVC ATM-Interworking service request if you must change or modify the links or other settings of the service request.

To modify an EVC ATM-Interworking service request, perform the following steps.

-
- Step 1** Choose **Operate > Service Request Manager**.
- The Service Request Manager window appears, showing service request available in Prime Provisioning.
- 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 [Creating an EVC ATM-Ethernet Interworking Service Request, page 3-36](#), 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 ATM-Interworking Service Request, page 3-41](#).
- 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 ATM-Interworking Service Request](#), page 3-42.

Deploying the EVC ATM-Ethernet Service Request

You can deploy an EVC ATM-Ethernet service in two different ways:

- If a service request has been saved, you may deploy it through the Service Request Manager window (choose **Operate > Service Request Manager**). For steps on how to do this, see [Chapter 10, “Managing Service Requests.”](#)
- Alternatively, you can deploy an EVC ATM-Ethernet service request from within the Service Request Editor window (while creating the service request). The Deploy button at the bottom of the window allows you to save and deploy the service request in one step.

Defining Frame Relay Policies

To define a Frame Relay policy (with or without a CE present), perform the following steps.

- Step 1** Choose **Service Design > Create Policy**.
The Policy Editor window appears.
- Step 2** Choose **L2VPN** from the Policy Type drop-down list.
The Policy Editor window appears.
- Step 3** Enter a **Policy Name** for the policy.
- Step 4** Choose the **Policy Owner** for the policy.
There are three types of 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 Provisioning Role-Based Access Control (RBAC) comes into play. For example, an 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 5** 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 6** Choose the **Service Type** of the L2VPN policy (in this case, Frame Relay).
- Step 7** Check or uncheck the **CE Present** check box **as required**.
- Step 8** Click **Next**.
The Interface Type window appears.
- Step 9** Set the attributes in the Interface Type window as described in [Table E-2](#).

**Note**

Attributes that appear in the GUI are determined by the type of policy being defined and whether or not a CE has been specified.

- Step 10** When you have set the attributes, click **Next** to proceed to the next window (or else click **Finish** to save the policy).
- Step 11** If you would like to use user-defined attributes within this policy, click **Next** (before clicking **Finish**).
An additional window appears the policy workflow. 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 D, “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 12** If you would like to enable template association for this policy, click **Next** (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 [Chapter 11, “Managing Templates and Data Files”](#) for more information about using templates and data files. 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 13** To save the Frame Relay policy, click **Finish**.

To create a service request based on a Frame Relay policy, see [Managing Service Requests, page 10-1](#).

Defining ATM Policies

To define an ATM policy (with or without a CE present), perform the following steps.

- Step 1** Choose **Service Design > Create Policy**.
The Policy Editor window appears.
- Step 2** Choose **L2VPN** from the Policy Type drop-down list.
The Policy Editor window appears.
- Step 3** Enter a **Policy Name** for the policy.
- Step 4** Choose the **Policy Owner** for the policy.
There are three types of 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 Provisioning Role-Based Access Control (RBAC) comes into play. For example, an 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 5** 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 6 Choose the **Service Type** of the L2VPN policy (in this case, ATM).

Step 7 Check or uncheck the **CE Present** check box **as required**.

Step 8 Click **Next**.

The Interface Type window appears.

Step 9 Set the attributes in the Interface Type window as described in [Table E-2](#).



Note Attributes that appear in the GUI are determined by the type of policy being defined and whether or not a CE has been specified.

Step 10 When you have set the attributes, click **Next** to proceed to the next window (or else click **Finish** to save the policy).

Step 11 If you would like to use user-defined attributes within this policy, click **Next** (before clicking **Finish**).

An additional window appears the policy workflow. 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 D, “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 12 If you would like to enable template association for this policy, click **Next** (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 [Chapter 11, “Managing Templates and Data Files”](#) for more information about using templates and data files. 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 13 To save the ATM policy, click **Finish**.

Managing a VPLS Service Request

This section contains the basic steps to provision a VPLS service. It contains the following subsections:

- [Overview, page 3-46](#)
- [Creating a VPLS Service Request, page 3-46](#)
- [Using Templates and Data Files with a VPLS Service Request, page 3-51](#)
- [Saving the VPLS Service Request, page 3-52](#)
- [Modifying the VPLS Service Request, page 3-52](#)

Overview

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.

To create a service request, a service policy must already be defined, as described in [Creating a VPLS Policy, page E-35](#). 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.

You can also associate Prime Provisioning templates and data files with a service request. See [Chapter 11, “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 D, “Adding Additional Information to Services.”](#)

The following steps are involved in creating a service request for Layer 2 connectivity between customer sites:

1. Choose a VPLS policy.
2. Choose a VPN. For more information, see [Defining VPNs, page 3-9](#).
3. Add a link.
4. Choose a CE or UNI interface.
5. Choose a Named Physical Circuit (NPC) if more than one NPC exists from the CE or the UNI interface.
6. Edit the link attributes.

For sample configlets for VPLS scenarios, see [Sample Configlets, page 3-122](#).

Creating a VPLS Service Request

For information on creating specific types of VPLS service requests, see the following sections:

- [Creating a VPLS Service Request with a CE, page 3-46](#)
- [Creating a VPLS Service Request without a CE, page 3-48](#)

Creating a VPLS Service Request with a CE

To create a VPLS service request with a CE present, perform the following steps.



Note

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.

- Step 1** Choose **Operate > Create Service Request**.
The Service Request Editor window appears.

- Step 2** From the policy picker, choose a VPLS policy from the policies previously created (see [Creating a VPLS Policy, page E-35](#)).

The new service request inherits all the properties of that VPLS policy, such as all the editable and noneditable features and preset attributes.

The Edit VPLS Link window appears.

- Step 3** 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 Provisioning 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 2-55](#).

- Step 4** Choose a **VPN Name** in the Select column.

- Step 5** Click **Select**.

The Edit VPLS Link window appears with the VPN name displayed.

- Step 6** Click **Add Link**.

The window updates, allowing you specify the CE endpoints.

- Step 7** 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 8** Click **Select CE** in the CE column.

The Select CPE Device window appears.

This window displays the list of currently defined CEs.

- From the **Show CPEs with** drop-down list, you can display CEs by Customer Name, by Site, or by Device Name.
- You can use the **Find** button to either search for a specific CE, or to refresh the display.
- You can set the **Rows per page** to 5, 10, 20, 30, 40, or All.

- Step 9** In the Select column, choose a CE for the VPLS link.

- Step 10** Click **Select**.

The Edit VPLS Link window appears displaying the name of the selected CE in the CE column.

- Step 11** Choose the CE interface from the interface picker.



Note When you provision an ERMS (EVP-LAN) service (and when you choose a UNI for a particular device), Prime Provisioning 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 12** 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 13 Choose the name of the NPC from the Select column.

Step 14 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 15 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 16 The Circuit ID is created automatically, based on the VLAN data for the circuit.

Step 17 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.

Step 18 Set attributes in this window per your requirements.



Note For more information on setting attributes in this window, see the corresponding attributes for the VPLS policy as described in [Table E-5](#).

Step 19 Continue to specify additional CEs, as in previous steps, if desired.

Step 20 Click **OK**.

Step 21 Click **Save**.

The service request is created and saved into Prime Provisioning.

For additional information on working with VPLS service requests, see the following sections:

- [Using Templates and Data Files with a VPLS Service Request, page 3-51](#)
- [Saving the VPLS Service Request, page 3-52](#)
- [Modifying the VPLS Service Request, page 3-52](#)
- [Deploying, Monitoring, and Auditing Service Requests, page 3-52](#)

Creating a VPLS Service Request without a CE

To create a VPLS service request without a CE present, perform the following steps.



Note 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.

Step 1 Choose **Operate > Create Service Request**.

The Service Request Editor window appears.

Step 2 From the policy picker, choose a VPLS policy from the policies previously created (see [Creating a VPLS Policy, page E-35](#)).

The new service request inherits all the properties of that VPLS policy, such as all the editable and non-editable features and preset attributes.

The Edit VPLS Link window appears.

Step 3 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 Provisioning 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 2-55](#).

Step 4 Choose a **VPN Name** in the Select column.

Step 5 Click **Select**.

The Edit VPLS Link window appears with the VPN name displayed.

Step 6 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 7 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 8 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 9 In the **Select** column, choose the PE device name for the VPLS link.

Step 10 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 11 To choose the UNI interface, click on the toggle button in the **Select One** field of the UNI Interface column.

The Interface Selection window appears. This window 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.

Step 12 Choose the UNI interface by clicking the radio button next to the interface name.



Note When you provision an ERMS service (and when you choose a UNI for a particular device), Prime Provisioning 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 13 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 14 Choose the name of the NPC from the **Select** column.

Step 15 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 16 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 17 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 the corresponding attributes for the VPLS policy as described in [Table E-5](#).

Step 18 Continue to specify additional PEs, as in previous steps, if desired.

Step 19 Click **Save**.

The service request is created and saved into Prime Provisioning.

For additional information on working with VPLS service requests, see the following sections:

- [Using Templates and Data Files with a VPLS Service Request, page 3-51](#)
- [Saving the VPLS Service Request, page 3-52](#)
- [Modifying the VPLS Service Request, page 3-52.](#)
- [Deploying, Monitoring, and Auditing Service Requests, page 3-52](#)

Using Templates and Data Files with a VPLS Service Request

The template mechanism in Prime Provisioning 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 [Chapter 11, “Managing Templates and Data Files”](#) for more information about using templates and data files.

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 requests 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.
- Step 3** If you are ready to deploy the service request, see [Deploying, Monitoring, and Auditing Service Requests, page 3-52](#).
-

Modifying the VPLS Service Request

To modify a VPLS service request, perform the following steps.

-
- Step 1** Choose **Operate > 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.
- 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.
- Step 7** Click **OK**.
- The Edit VPLS Link window appears.
- Step 8** When you are finished editing the VPLS links, click **Save**.
-

Deploying, Monitoring, and Auditing Service Requests

To apply EVC policies to network devices, you must deploy the service request. When you deploy a service request, Prime Provisioning compares the device information in the Repository (the Prime Provisioning database) with the current device configuration and generates a configlet. Additionally, you can perform various monitoring and auditing tasks on service requests. Information about common tasks that apply to all types of Prime Provisioning service requests is provided in [Chapter 10, “Managing Service Requests.”](#)

This section covers specific issues related to managing service request tasks for EVC services.

Pre-Deployment Changes

You can change the Dynamic Component Properties Library (DCPL) parameter **actionTakenOnUNIVlanList** before you deploy an EVC 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 > 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 Provisioning create the minimum VLAN list. This is the default.
 - **abort** to have Prime Provisioning 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 Provisioning allow all VLANs.
- Step 6** Click **Set Property**.
-

Provisioning VPLS Autodiscovery on Devices using EVC Service Requests

This section describes how enable the VPLS autodiscovery in Prime Provisioning. It contains the following sections:

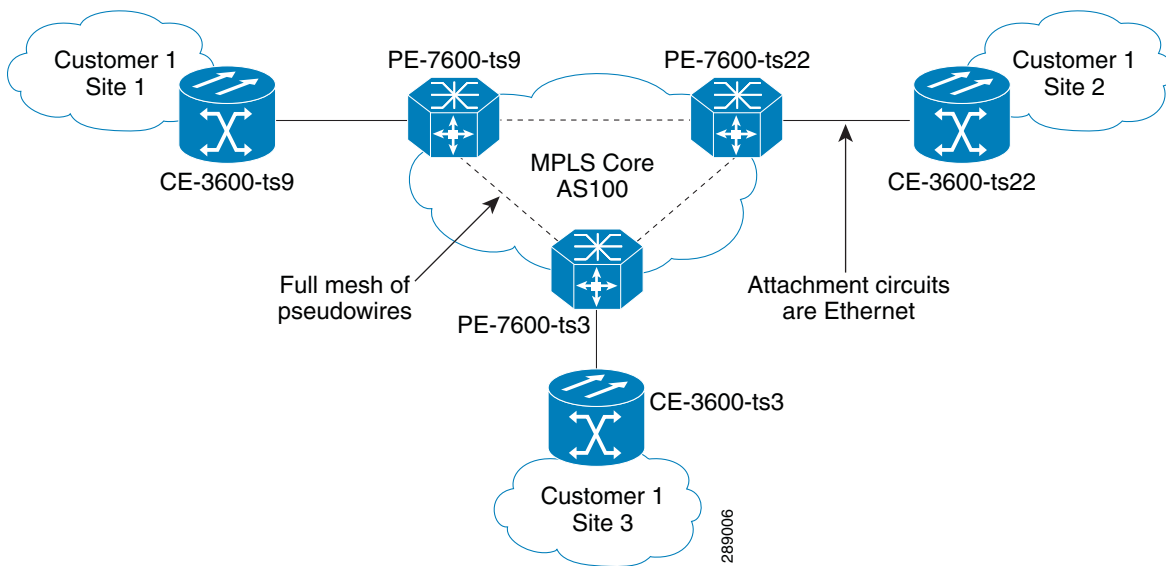
- [Overview, page 3-54](#)
- [Limitations and Restrictions for VPLS Autodiscovery, page 3-54](#)
- [Preconfiguring PE Devices to Support VPLS Autodiscovery, page 3-55](#)
- [Enabling VPLS Autodiscovery in the EVC Workflow, page 3-55](#)
- [Sample Configlets, page 3-56](#)

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 3-2 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 3-2 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 Provisioning. You must do this manually or through the use of templates. See [Preconfiguring PE Devices to Support VPLS Autodiscovery](#), page 3-55.
- 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 Provisioning.

- To use VPLS autodiscovery, all PE devices in the VPLS domain must 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 3-55](#).
- Split horizon should be enabled for when using VPLS autodiscovery.
- VPLS autodiscovery can only be configured in Prime Provisioning using EVC Ethernet service requests for which the MPLS Core Connectivity Type is set as VPLS. The feature is not supported for other Prime Provisioning 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 set up 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 3-56](#).

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 Provisioning for both IOS and IOS XR devices for VPLS autodiscovery.

Sample Configlet for IOS Device

```
! Setup VPLS instance,!
12 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

```
12vpn
bridge group abc
bridge-domain east
vfi vfname
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.

Policy and Service Request Attributes Reference Tables

This section provides reference information for attributes appearing in windows in EVC Ethernet, EVC ATM-Ethernet Interworking, EVC policies and service requests. To find attributes and descriptions refer to the appropriate section for the service:

- [EVC Ethernet Service Attributes, page 3-57](#)
- [EVC ATM-Ethernet Interworking Service Attributes, page 3-92](#)
- [Sample Configlets, page 3-122](#)

EVC Ethernet Service Attributes

This section describes policy and service request attributes for EVC Ethernet services:

- [EVC Ethernet Policy Attributes, page 3-57](#)
- [EVC Ethernet Service Request Attributes, page 3-74](#)

EVC Ethernet Policy Attributes

- [Service Options Window, page 3-57](#)
- [EVC Attributes Window, page 3-62](#)
- [Interface Attributes Window, page 3-68](#)



Note

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).

Service Options Window

[Table 3-2](#) describes the attributes in the Service Options window of the EVC Ethernet policy workflow.

Table 3-2 Service Options

Attribute	Description
CE Directly Connected to EVC	<p>Check the box if the CEs are directly connected to the N-PE. Usage notes:</p> <ul style="list-style-type: none"> • 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 Provisioning behavior. There is no change in NPC implementation to support EVC functionality.

Table 3-2 Service Options (continued)

Attribute	Description
All Links Terminate on EVC	<p>Check the box if links need to be configured with EVC features. Usage notes:</p> <ul style="list-style-type: none"> • 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. • 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 Provisioning policy types (L2VPN or VPLS) can be used instead of EVC.
All L2 Access Links default to EVC UNI	<p>Check the box to enable EVC syntax configuration on all access devices (U-PE and PE-AGG) throughout the circuit. This shows up in service request as EVC-related attributes for all of these device types. If this attribute is not enabled in the service request, EVC service-related syntax will only be available for N-PE devices.</p>
MPLS Core Connectivity Type	<p>From the drop-down list, choose the MPLS core connectivity type. The core option supports MPLS only. There is no L2TPv3 support for this service. The choices are:</p> <ul style="list-style-type: none"> • 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.
	<ul style="list-style-type: none"> • 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: <ul style="list-style-type: none"> – 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.

Table 3-2 Service Options (continued)

Attribute	Description
	<ul style="list-style-type: none"> • VPLS—Choose this option to allow connectivity between multiple N-PEs across the MPLS core. <ul style="list-style-type: none"> – This includes support for multi-segment pseudowire over an MPLS-TP enabled network. Some or all of the LSPs interconnecting the VPLS instances can be admitted onto existing MPLS-TP tunnels (which may have been provisioned using Prime Provisioning). The LSPs may be configured as multi-segment pseudowires, where each hop can be admitted onto an MPLS-TP tunnel. Prime Provisioning will automatically route the multi-segment pseudowire along the shortest path, taking into consideration any included and/or excluded nodes and/or tunnels. – The LSP/pseudowire labels may be statically allocated by Prime Provisioning. This eliminates the need for a directed protocol to be run within the VPLS to do label exchange and therefore further eliminates the need for IP connectivity between the endpoints in the VPLS. – The pool of MPLS labels is shared across VPLS and MPLS-TP services (if they come from the same MPLS static label range on the device). Otherwise Prime Provisioning uses the separate tunnel and service label ranges that are configured on the device. Labels already in use are discovered and removed from the label pool to ensure unique allocation of MPLS labels. <p>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.</p>
Configure With Bridge Domain	<p>Check the 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.</p> <ul style="list-style-type: none"> • PSEUDOWIRE as the MPLS Core Connectivity Type. There are two cases: <ul style="list-style-type: none"> A. With EVC: <ul style="list-style-type: none"> – 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: <ul style="list-style-type: none"> – 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.


Table 3-2 Service Options (continued)

Attribute	Description
	<p>Only pseudowires can be either configured directly under service instance of the corresponding EVC-capable interface or under SVIs associated to the bridge domain.</p> <ul style="list-style-type: none"> • LOCAL as the MPLS Core Connectivity Type: <ul style="list-style-type: none"> – 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 Provisioning allows only point-to-point local connects without bridge domain. • VPLS—Configure With Bridge Domain is checked by default and non-editable. <p>When the VPLS service option is selected, VPLS-specific service options appear.</p> <ul style="list-style-type: none"> – Check the Static VPLS (AutoPick MPLS Labels) check box to automatically allocate static labels. The static labels are allocated when the service request is saved. – Check the Configure Pseudowire Segment(s) check box to allow the VPLS service to be admitted onto MPLS-TP tunnels and “stitch” together tunnels to form a simulated end-to-end path.
Allow Spoke nodes	<p>This attribute is used to enable H-VPLS on EVC VPLS services. It allows the selection of spoke (leaf) nodes in the access of H-VPLS topology. If this check box is enabled, in the service request workflow, the user will be able to set the N-PE as hub or spoke nodes alone. (A spoke node can also be called a “leaf node,” which is connected to a hub node).</p> <p>This attribute only appears if the MPLS Core Connectivity Type is set as VPLS.</p>
Allow Spoke with Spoke nodes	<p>This attribute can also be used to enable H-VPLS on EVC VPLS services. It allows the selection of interior nodes in the access of H-VPLS topology, which can again be connected with other leaf nodes. Enabling this check box will enable “Allow spokes nodes” (see previous attribute) by default. Because of this, in the service request workflow, the user will be able to set the N-PE as a hub or spoke with additional spoke nodes.</p> <p>When you provision an H-VPLS service with the node as HUB, you can save the SR without selecting an UNI. But when the H-VPLS node is a SPOKE or SPOKE WITH SPOKE you need to select an interface.</p> <p>This attribute only appears if the MPLS Core Connectivity Type is set as VPLS.</p>
E Tree	<p>Check the box to choose one of the E Tree role options. When you choose E Tree, the split horizon attribute is hidden from the policy and SR level, and it is controlled internally using E Tree Role.</p> <p>This attribute only appears if the MPLS Core Connectivity Type is set as VPLS.</p>
E Tree Role	<p>Choose an option. The choices are:</p> <ul style="list-style-type: none"> • Root - Allows communication with all the end points. • Leaf - Allows communication only with the root nodes. <p>To change the E Tree role option while creating a Service Request, check the Editable box of this attribute.</p>

Table 3-2 Service Options (continued)

Attribute	Description
Split Horizon	<p>Check the box to enable split horizon with bridge domain. Usage notes:</p> <ul style="list-style-type: none"> • The Use Split Horizon attribute is disabled by default. • The Use Split Horizon attribute can be used only when the Configure With Bridge Domain check box is checked (enabled). • When 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.
Static Pseudowire (Autopick MPLS Labels)	<p>Choose a type. The choices are:</p> <ul style="list-style-type: none"> • All Dynamic—Labels will be allocated dynamically during provisioning. No static labels will be added into the configlet. • All Static—Labels will be allocated statically during provisioning. Every segment in a multi-segment pseudowire will have static labels assigned to it on per-segment basis. • Defaults—Prime Provisioning will automatically determine whether or not to apply static labels based on the core type of the segment. It will do this on a per segment basis. A multi-segment pseudowire over LDP defaults to dynamic pseudowire. Multi-segment pseudowire over MPLS-TP defaults to static pseudowire. <p>This attribute only supported for MPLS Core Connectivity Types of PSEUDOWIRE or VPLS.</p>
Configure Pseudowire Segment(s)	<p>Check the box to enable ability to configure pseudowire classes on a per segment basis in the service request based on this policy. Usage notes:</p> <ul style="list-style-type: none"> • The Configure Pseudowire Segment(s) attribute is only applicable for MPLS core connectivity types of PSEUDOWIRE and VPLS. With a VPLS core type, the attribute shows up in the Service Options window of the Policy Editor. With a PSEUDOWIRE core type, the attribute shows up in the Interface Attributes window in the block of other pseudowire-related attributes. • The Configure Pseudowire Segment(s) attribute is used in conjunction with the Static Pseudowire (Autopick MPLS Labels) attribute to configure the individual segments within a multi-segment pseudowire to be either dynamic or static. This allows you to override the default behavior of Prime Provisioning. • A segment can be a TP tunnel, a TE tunnel, or an LDP (dynamic) core. • The configuration is done subsequently in the service request based on the policy. When setting up the links in the service request, you can independently assign Pseudowire classes to ends of the segments of multi-segment pseudowires. For information on attaching pseudowire classes to links see Configuring Multi-segment Pseudowires, page 3-27. • The Configure Pseudowire Segment(s) attribute is not currently supported in EVC ATM-Ethernet Interworking policies and service requests.

Table 3-2 Service Options (continued)

Attribute	Description
UNI Multiplexing	<p>This attribute helps in creating EPL and EVPL policy. Usage notes:</p> <ul style="list-style-type: none"> To create an EVPL policy, check the UNI Multiplexing check box. <p> Note By default, the check box is always checked</p> <ul style="list-style-type: none"> To create an EPL policy, uncheck the UNI Multiplexing check box. <p>When you create an SR using EPL policy with an interface, then that interface is not available for further provisioning of any EPL or EVPL services. When you create an SR using EVPL policy with an interface, then that interface will be available for provisioning EVPL services but not for EPL services.</p>
Reserved Bandwidth	<p>Pseudowire bandwidth for EVC services can be managed by specifying this attribute value. Prime Provisioning compares the reserved bandwidth value with the bandwidth unallocated on the MPLS-TP tunnel. If this value is more than the tunnel bandwidth, provisioning throws an error and moves the SR to invalid state.</p> <p>This attribute only appears if the MPLS Core Connectivity Type is set as PSEUDOWIRE.</p>
Configure Pseudowire Headend	<p>When this check box is checked the interface acts as a pseudowire-ether interface and the special attributes related to pseudowire headend appears during the SR creation. When this check box is unchecked the interface of pseudowire-ether acts as a normal gigabit interface where sub interfaces can be configured.</p> <p>This attribute only appears if the MPLS Core Connectivity Type is set as PSEUDOWIRE.</p>

EVC Attributes Window

Table 3-3 describes the attributes available in the EVC Attributes window of EVC Ethernet policy workflow.

EVC attributes are organized under the following categories:

- **Service Attributes.** The EVC service attributes are the same no matter which MPLS Core Connectivity Type has been selected.
- **VLAN Match Criteria.** 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 Provisioning 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.
- **VLAN Rewrite Criteria.** 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.

Table 3-3 EVC Attributes

Attribute	Description
Service Attributes	
AutoPick Service Instance ID	<p>Check the 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 Provisioning link attributes during service request creation, Prime Provisioning will prompt the operator to specify the service instance ID. Usage notes:</p> <ul style="list-style-type: none"> • 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 Provisioning 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.
AutoPick Service Instance Name	<p>Check the box to have Prime Provisioning autogenerated a service instance name when you create a service request based on the policy. The autogenerated value is in the following pattern: <i>CustomerName_ServiceRequestJobID</i>. If the check box is unchecked, then you can enter a value during service request creation.</p>
Enable PseudoWire Redundancy	<p>Check the box to enable pseudowire redundancy (alternative termination device) under certain conditions. Usage notes:</p> <ul style="list-style-type: none"> • Enable Pseudo Wire Redundancy is only available if the MPLS Core Connectivity Type was set as PSEUDOWIRE in the Service Options window (see Defining the EVC Ethernet Policy, page 3-20). • Enabling this feature allows the user to do the following: <ul style="list-style-type: none"> – Configure two pseudowires between two direct links, or: – Add a backup peer such that pseudowires are configured between A–Z and A–Z'. In this case, the terminating links A, Z, and Z' must all be directly connected links. L2 access links are not supported as backup peers. • See Setting Up Pseudowire Redundancy and a Backup Peers, page 3-29, for more information on using this feature in service requests. • See Appendix B, “Terminating an Access Ring on Two N-PEs” and, specifically, the section Using N-PE Redundancy in FlexUNI/EVC Service Requests, page B-3, for notes on how this option can be used.

Table 3-3 EVC Attributes (continued)

Attribute	Description
Enable Trunk EFP	<p>Check the box to provide flexibility to make many Layer 2 flow points within one interface. This attribute appears only when the EVC check box is checked in Direct Connect Links section. Trunk EFP supports only ASR920, ASR902 and ASR903 IOS device. Usage notes:</p> <ul style="list-style-type: none"> • If the check box is checked, user will get “service instance trunk <id> ethernet” command. • If the check box is checked, the Match Inner and Outer Tags are disabled. • If the check box is checked, Autopick Outer VLAN and single value Outer VLAN are not allowed. • If the check box is checked, Outer VLAN ID only with ranges is allowed.
AutoPick VC ID	<p>Check the box to have Prime Provisioning 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:</p> <ul style="list-style-type: none"> • This attribute is available only if MPLS Core Connectivity of Type was set as PSEUDOWIRE or VPLS in the Service Options window (see Defining the EVC Ethernet Policy, page 3-20). • When AutoPick VC ID is checked, Prime Provisioning allocates a VC ID for pseudowires from the Prime Provisioning-managed VC ID resource pool. • If MPLS Core Connectivity of Type is VPLS, Prime Provisioning allocates the VPLS VPN ID from the Prime Provisioning-managed VC ID resource pool.
AutoPick VFI Name	<p>Check the box to have Prime Provisioning 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.</p> <p>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.</p>
AutoPick Bridge Domain/VLAN ID	<p>Check the box to have Prime Provisioning 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:</p> <ul style="list-style-type: none"> • AutoPick Bridge Domain/VLAN ID consumes a global VLAN ID on the device. • The bridge domain/VLAN ID is picked from the existing Prime Provisioning VLAN pool. Once the VLAN ID is assigned in the service request, Prime Provisioning makes the VLAN ID unavailable for subsequent service requests. • In the case of manual VLAN ID allocation, Prime Provisioning does not manage the VLAN ID if the ID lies outside the range of an Prime Provisioning-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 Provisioning-managed VLAN pool and the VLAN ID is already in use in the access domain, Prime Provisioning displays an error message indicating that the VLAN ID is in use. <p>For additional information on Access VLAN IDs, see Note on Access VLAN IDs, page 3-67.</p>

Table 3-3 EVC Attributes (continued)

Attribute	Description
AutoPick Bridge Group Name	Check the box to have Prime Provisioning 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. This attribute is applicable only for supported IOS XR devices.
AutoPick Bridge Domain Name	Check the check box to have Prime Provisioning autopick the domain name for the service request during service request creation. Usage notes: <ul style="list-style-type: none"> • 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: <ul style="list-style-type: none"> – For pseudowire and local connect core types: <i>ISC-Job-Job_ID</i>, where <i>Job_ID</i> is the service request job ID. – For VPLS core type: <i>ISC-VPN_Name-VPN_ID</i>, where <i>VPN_Name</i> is the name of the VPLS VPN being used, and <i>VPN_ID</i> is the VPN ID used in the service request. • This attribute is applicable only for supported IOS XR devices.
VLAN Matching Criteria Attributes	
Match	Choose an encapsulation type from the drop-down list. The choices are: <ul style="list-style-type: none"> • DOT1Q • DEFAULT • UNTAGGED • PRIORITY TAGGED • DOT1AD Selecting Default as the match criteria disables the Outer VLAN ID and Outer VLAN Ranges fields on the page. If Default is the CE encapsulation type, Prime Provisioning shows another field for the UNI port type.
Match Inner and Outer Tags	Check the 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 attribute causes the Inner VLAN Ranges attribute (covered in the next steps) to appear in the EVC Attribute window.
Inner VLAN Ranges	Check the 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.
Outer VLAN Ranges	Check the 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.

Table 3-3 EVC Attributes (continued)


Attribute	Description
AutoPick Outer VLAN	<p>Check the box to have Prime Provisioning 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. Usage notes:</p> <ul style="list-style-type: none"> • Use of the AutoPick Outer VLAN attribute requires that two elements have already been set up in Prime Provisioning. 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 2-39, and Resource Pools, page 2-43. • 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 Provisioning VLAN pool.
AutoPick Inner VLAN	<p>Check the box to have Prime Provisioning autopick the Inner VLAN ID from a previously created inner VLAN ID resource pool during service request creation. If this check box is unchecked, the operator will be prompted to specify an Inner VLAN ID during service request creation. Usage notes:</p> <ul style="list-style-type: none"> • Use of the AutoPick Inner VLAN attribute requires that two elements have already been set up in Prime Provisioning. 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 Inner 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 2-39, and Resource Pools, page 2-43. • AutoPick Inner VLAN can be used for interfaces that support EVC functionality. • AutoPick Inner VLAN consumes a VLAN ID on the interface that supports EVC. • The bridge domain VLAN ID is picked from the existing Prime Provisioning VLAN pool. <p> Note The Match Inner and Outer Tags check box should be enabled in the policy to make use of AutoPick Inner VLAN feature during service request creation.</p>
VLAN Rewriter Criteria Attributes	
Pop Outer	Check the 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.
Pop Inner	Check the 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.

Table 3-3 EVC Attributes (continued)

Attribute	Description
Push Outer	<p>Check the 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:</p> <ul style="list-style-type: none"> • 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 Provisioning-managed VLAN ID pools.
Push Inner	<p>Check the 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:</p> <ul style="list-style-type: none"> • 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 Provisioning-managed VLAN ID pools.
Translate Outer	<p>Check the 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.</p>
Translate Inner	<p>Check the 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.</p>

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 Provisioning by means of locally-significant VLAN pools. But once a VLAN ID is assigned for a link in the service request, Prime Provisioning 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 Provisioning 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.

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



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.

Interface Attributes Window

[Table 3-5](#) describes the attributes available in the EVC Attributes window of EVC Ethernet policy workflow. 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.



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 [Defining the EVC Ethernet Policy, page 3-20](#)). For completeness, all attributes available for the different core types are listed in the table. Attributes apply to all core types, unless otherwise noted.

Table 3-5 Interface Attributes

Attribute	Description
Standard UNI Port	Check the 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.
UNI Shutdown	Check the 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.
Keep Alive	Check the 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.
Link Media (optional)	Enter None, auto-select, rj45, or sfp.
Link Speed (optional)	Enter None, 10, 100, 1000, Auto, or nonegotiate.
Link Duplex (optional)	Enter None, Full, Half, or Auto.
Encapsulation	Choose a type. The choices are: <ul style="list-style-type: none"> • 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.
VLAN Translation	Specify the type for this policy by clicking the appropriate radio button. The choices are: <ul style="list-style-type: none"> • 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. <p>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 3-22.</p>

Table 3-5 Interface Attributes (continued)

Attribute	Description
Use Pseudowire Interface	<p>Check the Use Pseudowire Interface check box to define pseudowire interface details and to create cross connection between the service instance to the pseudowire interface.</p> <p>This feature is applicable only for the ASR920, ASR 902 and ASR 903 devices. Also, ensure that in the Policy Editor window, in the Service Options area Configure With Bridge Domain check box is not selected because you cannot use both the Use Pseudowire Interface and Configure With Bridge Domain Services at a time.</p>
Use PseudoWireClass	<p>Check the box to enable the selection of a pseudowire class. Usage notes:</p> <ul style="list-style-type: none"> • The pseudowire class name is used for provisioning pw-class commands on IOS and IOS XR devices. See Creating and Modifying Pseudowire Classes, page 3-15 for additional information on pseudowire class support. • 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 Provisioning. • The Use PseudoWireClass attribute is only available if the MPLS core connectivity type was set as PSEUDOWIRE in the Service Options window (see Defining the EVC Ethernet Policy, page 3-20).
E-Line Name	<p>Specify the point-to-point (p2p) E-line name. Usage notes:</p> <ul style="list-style-type: none"> • If no value is specified for the E-Line Name in either the policy or the service request based on the policy, Prime Provisioning autogenerates a default name as follows: <ul style="list-style-type: none"> – For PSEUDOWIRE core connectivity type, the format is: <i>DeviceName--VC_ID</i> – For LOCAL core connectivity type, the format is: <i>DeviceName--0--VLAN_ID</i> <p>If the default name is more than 32 characters, the device names are truncated.</p> • The E-Line Name attribute is not available if the MPLS core connectivity type was set as VPLS in the Service Options window (see Defining the EVC Ethernet Policy, page 3-20). • E-Line Name is only applicable for IOS XR devices.

Table 3-5 Interface Attributes (continued)

Attribute	Description
Configure Pseudowire Segment(s)	<p>Check the box to enable ability to configure pseudowire classes on a per segment basis in the service request based on this policy. Usage notes:</p> <ul style="list-style-type: none"> • The Configure Pseudowire Segment(s) attribute is only applicable for MPLS core connectivity types of PSEUDOWIRE and VPLS. With a VPLS core type, the attribute shows up in the Service Options window of the Policy Editor. With a PSEUDOWIRE core type, the attribute shows up in the Interface Attributes window in the block of other pseudowire-related attributes. • The Configure Pseudowire Segment(s) attribute is used in conjunction with the Static Pseudowire (Autopick MPLS Labels) attribute to configure the individual segments within a multi-segment pseudowire to be either dynamic or static. This allows you to override the default behavior of Prime Provisioning. • A segment can be a TP tunnel, a TE tunnel, or an LDP (dynamic) core. • The configuration is done subsequently in the service request based on the policy. When setting up the links in the service request, you can independently assign Pseudowire classes to ends of the segments of multi-segment pseudowires. For information on attaching pseudowire classes to links see Configuring Multi-segment Pseudowires, page 3-27. • The Configure Pseudowire Segment(s) attribute is not currently supported in EVC ATM-Ethernet Interworking policies and service requests.

Table 3-5 Interface Attributes (continued)

Attribute	Description
N-PE Pseudo-wire on SVI	<p>Check the box to have Prime Provisioning generate forwarding commands under SVIs (switch virtual interfaces). By default, this check box is not checked. In this case, Prime Provisioning generates forwarding commands under the service instance.</p> <p>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.</p> <p>Usage notes:</p> <ul style="list-style-type: none"> • Prime Provisioning 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 3-166 and EVC (Pseudowire Core Connectivity, no Bridge Domain, no Pseudowire on SVI), page 3-167. • 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 3-6 shows various use cases for hybrid configuration for EVC service requests.
Use Existing ACL Name	Check the box if you want to assign your own named access list to the port. By default, this check box is not checked and Prime Provisioning automatically assigns a MAC-based ACL on the customer facing UNI port, based on values you enter in UNI MAC addresses (below).
Port-Based ACL Name	Enter a Port-Based ACL Name (if you checked the Use Existing ACL Name check box). Prime Provisioning 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.
UNI MAC addresses	Enter one or more Ethernet 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.

Table 3-5 Interface Attributes (continued)

Attribute	Description
UNI Port Security	<p>Check the box if you want to provision port security-related CLIs to the UNI port by controlling the MAC addresses that are allowed to go through the interface.</p> <ul style="list-style-type: none"> • For Maximum Number of MAC address, enter the number of MAC addresses allowed for port security. • For Aging, enter the length of time the MAC address can stay on the port security table. • For Violation Action, choose what action will occur when a port security violation is detected: <ul style="list-style-type: none"> – 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. • In the Secure MAC Addresses field, enter one or more Ethernet MAC addresses.
Enable Storm Control	<p>Check the 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.</p>
Protocol Tunnelling	<p>Check the 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:</p> <ul style="list-style-type: none"> • Enable cdp—Enable Layer 2 tunnelling on Cisco Discover Protocol (CDP). • cdp shutdown threshold—Enter the number of packets per second to be received before the interface is shut down. • cdp drop threshold—Enter the number of packets per second to be received at which point the interface will start dropping CDP packets. • Enable vtp—Enable Layer 2 tunnelling on VLAN Trunk Protocol (VTP). • vtp shutdown threshold—Enter the number of packets per second to be received before the interface is shut down. • vtp drop threshold—Enter the number of packets per second to be received at which point the interface will start dropping VTP packets. • Enable stp—Enable Layer 2 tunnelling on Spanning Tree Protocol (STP). • stp shutdown threshold—Enter the number of packets per second to be received before the interface is shut down. • stp drop threshold—Enter the number of packets per second to be received at which point the interface will start dropping STP packets. • Recovery Interval—Enter the amount of time, in seconds, to wait before recovering a UNI port.

Table 3-5 Interface Attributes (continued)

Attribute	Description
MTU Size	<p>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 Provisioning 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.</p> <ul style="list-style-type: none"> 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 Provisioning uses 9216 in both cases. For the Cisco 7600 SVI (interface VLAN), the MTU size is 1500 to 9216. <p>MTU attribute support has been extended to Direct Access Links attribute section for VPLS Core Connectivity for EVC VPLS services. Earlier this attribute was available only in Links with L2 Access Nodes section. This attribute is applicable for both IOS and IOS-XR devices and MTU size range varies for both the devices.</p> <ul style="list-style-type: none"> For the IOS devices, the MTU range is 64-9216. For the IOS-XR devices, the MTU range is 46-65535.

Table 3-6 Use Cases for Hybrid Configuration for EVC Service Requests

Use Bridge Domain	EVC	N-PE Pseudowire on SVI	CLIs Generated
True	True	True	<ul style="list-style-type: none"> xconnect under VLAN interface. Service instance under main interface.
True	True	False	<ul style="list-style-type: none"> xconnect under service instance. Service instance under main interface.
False	True	N/A	<ul style="list-style-type: none"> 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.

EVC Ethernet Service Request Attributes

This section provides information about attributes available in the EVC Ethernet service request workflow:

- [Table 3-7, “Pseudowire Core Connectivity Attributes,” on page 75](#)
- [Table 3-8, “VPLS Core Connectivity Attributes,” on page 77](#)
- [Table 3-9, “Local Core Connectivity Attributes,” on page 80](#)

- [Table 3-10, “Service Instance Details Attributes,” on page 81](#)
- [Table 3-11, “Standard UNI Attributes,” on page 87](#)

Table 3-7 Pseudowire Core Connectivity Attributes

Attribute	Description
Job ID, SR ID	These 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 Provisioning database holds within the editing flow of the service request.
Policy	This 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.
Select VPN	<p>Click to choose a VPN for use with this service request. The Select VPN window appears with the VPNs defined in the system.</p> <p>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.</p> <ol style="list-style-type: none"> 1. Choose a VPN Name in the Select column. You may also use the New VPN Details section of the window to create a new VPN “on the fly.” This window provides a subset of the usual VPN creation features. Use the supplied fields to name the new VPN, select/create the customer, and so on. For more information about creating VPNs, see Setting Up Logical Inventory, page 2-55. 2. Click Select. <p>The EVC Service Request Editor window appears with the VPN name displayed.</p>
AutoPick VC ID	Check the box if you want Prime Provisioning 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 Provisioning allocates a VC ID for pseudowires from the Prime Provisioning-managed VC ID resource pool. In this case, the text field for the VC ID option is non-editable.
VC ID	<p>If AutoPick VC ID was unchecked, enter a VC ID in the VC ID field. Usage notes:</p> <ul style="list-style-type: none"> • The VC ID value must be an integer value corresponding to a VC ID. • When a VC ID is manually allocated, Prime Provisioning verifies the VC ID to see if it lies within Prime Provisioning’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 Provisioning prompts you to allocate a different VC ID. If the VC ID lies outside of the Prime Provisioning VC ID pool, Prime Provisioning 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.

Table 3-7 Pseudowire Core Connectivity Attributes (continued)

Attribute	Description
Enable PseudoWire Redundancy	<p>Check the box to enable pseudowire redundancy (alternative termination device) under certain conditions. Usage notes:</p> <ul style="list-style-type: none"> • Enable Pseudo Wire Redundancy is only available if the MPLS Core Connectivity Type was set as PSEUDOWIRE in the Service Options window (see Defining the EVC Ethernet Policy, page 3-20). • Enabling this feature allows the user to do the following: <ul style="list-style-type: none"> – Configure two pseudowires between two direct links, or: – Add a backup peer such that pseudowires are configured between A–Z and A–Z'. In this case, the terminating links A, Z, and Z' must all be directly connected links. L2 access links are not supported as backup peers. • See Setting Up Pseudowire Redundancy and a Backup Peers, page 3-29, for more information on using this feature in service requests. • See Appendix B, “Terminating an Access Ring on Two N-PEs” and, specifically, the section Using N-PE Redundancy in FlexUNI/EVC Service Requests, page B-3, for notes on how this option can be used.
Backup PW VC ID	<p>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 above. The backup VC ID behaves the same as the VC ID of the primary pseudowire.</p>
Static Pseudowire (Autopick MPLS Labels)	<p>Choose a type. The choices are:</p> <ul style="list-style-type: none"> • All Dynamic—Labels will be allocated dynamically during provisioning. No static labels will be added into the configlet. • All Static—Labels will be allocated statically during provisioning. Every segment in a multi-segment pseudowire will have static labels assigned to it on per-segment basis. • Defaults—Prime Provisioning will automatically determine whether or not to apply static labels based on the core type of the segment. It will do this on a per segment basis. A multi-segment pseudowire over LDP defaults to dynamic pseudowire. Multi-segment pseudowire over MPLS-TP defaults to static pseudowire. <p>This attribute only supported for MPLS Core Connectivity Types of PSEUDOWIRE or VPLS.</p>

Table 3-7 Pseudowire Core Connectivity Attributes (continued)

Attribute	Description
Configure Bridge Domain	<p>Check the 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:</p> <ul style="list-style-type: none"> • With EVC: <ul style="list-style-type: none"> – 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: <ul style="list-style-type: none"> – 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. <p>Pseudowires can be configured either directly under service instance of the corresponding EVC-capable interface or under SVIs associated to the bridge domain.</p>
Use Split Horizon	<p>Check the box to enable split horizon with bridge domain. Usage notes:</p> <ul style="list-style-type: none"> • 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.
Description	<p>Click the “Click here” link to enter a description label for the service request. This is useful for searching the Prime Provisioning database for the particular service request. A dialogue appears in which you can enter a description.</p>

Table 3-8 VPLS Core Connectivity Attributes

Attribute	Description
Job ID, SR ID	<p>These 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 Provisioning database holds within the editing flow of the service request.</p>
Policy	<p>This field is read-only. It displays the name of the policy on which the service request is based.</p>

Table 3-8 VPLS Core Connectivity Attributes (continued)

Attribute	Description
Select VPN	<p>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.</p> <p>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. 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.</p> <ol style="list-style-type: none"> 1. Choose a VPN Name in the Select column. You may also use the New VPN Details section of the window to create a new VPN “on the fly.” This window provides a subset of the usual VPN creation features. Use the supplied fields to name the new VPN, select/create the customer, and so on. For more information about creating VPNs, see Setting Up Logical Inventory, page 2-55. 2. Click Select. <p>The EVC Service Request Editor window appears with the VPN name displayed.</p>
AutoPick VPLS VPN ID	<p>Check the box if you want Prime Provisioning 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.</p> <ul style="list-style-type: none"> • When AutoPick VPLS VPN ID is checked, Prime Provisioning allocates a VPLS VPN ID from the Prime Provisioning-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.
VPLS VPN ID	<p>If AutoPick VPLS VPN ID was unchecked, enter a VPLS VPN ID in the VPLS VPN ID field. Usage notes:</p> <ul style="list-style-type: none"> • The VPLS VPN ID value must be an integer value corresponding to a VPN ID. • When a VPLS VPN ID is manually allocated, Prime Provisioning verifies the VPLS VPN ID to see if it lies within Prime Provisioning’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 Provisioning prompts you to allocate a different VPLS VPN ID. If the VPLS VPN ID lies outside of the VC ID pool, Prime Provisioning 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.

Table 3-8 VPLS Core Connectivity Attributes (continued)


Attribute	Description
AutoPick VFI Name	<p>Check the box if you want Prime Provisioning 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:</p> <ul style="list-style-type: none"> When AutoPick VFI name is checked, Prime Provisioning generates a VFI name in the following format: <i>VPN name-VC ID</i> <p> Note When the VFI name is automatically generated, any special characters associated with the VPN name, such as '&' ';' and '.', are converted into '_'</p> <ul style="list-style-type: none"> This attribute is useful when importing an existing service into Prime Provisioning 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.
VFI Name	If AutoPick VFI Name was unchecked, enter a VFI name in the VFI Name field.
Discovery Mode	<p>Choose the type for VPLS autodiscovery. The choices are:</p> <ul style="list-style-type: none"> 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. <p>For details on how this feature is supported in Prime Provisioning, device preconfiguration requirements, and limitations, see Provisioning VPLS Autodiscovery on Devices using EVC Service Requests, page 3-53.</p>
Static Pseudowire (Autopick MPLS Labels)	<p>Choose a type. The choices are:</p> <ul style="list-style-type: none"> All Dynamic—Labels will be allocated dynamically during provisioning. No static labels will be added into the configlet. All Static—Labels will be allocated statically during provisioning. Every segment in a multi-segment pseudowire will have static labels assigned to it on per-segment basis. Defaults—Prime Provisioning will automatically determine whether or not to apply static labels based on the core type of the segment. It will do this on a per segment basis. <p>This attribute only supported for MPLS Core Connectivity Types of PSEUDOWIRE or VPLS.</p>
Configure Bridge Domain	<p>The box is checked by default and cannot be changed. Usage notes:</p> <ul style="list-style-type: none"> 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.
Description	Click the “Click here” link to enter a description label for the service request. A dialogue appears in which you can enter a description.

Table 3-9 Local Core Connectivity Attributes

Attributes	Description
Job ID, SR ID	These 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 Provisioning database holds within the editing flow of the service request.
Policy	This field is read-only. It displays the name of the policy on which the service request is based.
Select VPN	<p>Click Select VPN to choose a VPN for use with this service request. The Select VPN window appears with the VIPs defined in the system.</p> <p>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.</p> <ol style="list-style-type: none"> 1. Choose a VPN Name in the Select column. You may also use the New VPN Details section of the window to create a new VPN “on the fly.” This window provides a subset of the usual VPN creation features. Use the supplied fields to name the new VPN, select/create the customer, and so on. For more information about creating VIPs, see Setting Up Logical Inventory, page 2-55. 2. Click Select. <p>The EVC Service Request Editor window appears with the VPN name displayed.</p>
Configure Bridge Domain	<p>Check the box to determine bridge domain characteristics. Usage notes:</p> <ul style="list-style-type: none"> • 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.) See the following comments for details on how Prime Provisioning degenerates 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: <i>CustomerNameTruncatedToMaxPossibleCharacters_ServiceRequestJobID</i> For example, if the customer name is North American Customer and the service request job ID is 56345, the degenerated connect name would be NorthAmer_56345. The CLI generated would be: <pre>connect NorthAmer_56345 GigabitEthernet7/0/5 11 GigabitEthernet7/0/4 18</pre> In this case, 11 and 18 are service instance IDs. <ul style="list-style-type: none"> • If the policy setting for Configure Bridge Domain is non-editable, the option in the service request will be read-only.

Table 3-9 Local Core Connectivity Attributes (continued)

Attributes	Description
Use Split Horizon	<p>Check the box to enable split horizon with bridge domain. Usage notes:</p> <ul style="list-style-type: none"> • 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.
Description	Click the “Click here” link to enter a description label for the service request. A dialogue appears in which you can enter a description.

Table 3-10 Service Instance Details Attributes

Attribute	Description
Ectopic Service Instance ID	<p>Check the box to specify that the service instance ID will be degenerated 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:</p> <ul style="list-style-type: none"> • 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 Provisioning 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.
Service Instance ID	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.
AutoPick Service Instance Name	<p>Check the 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:</p> <ul style="list-style-type: none"> • If the check box is checked, the Service Instance Name text field is disabled. • The service instance name is autogenerated in the following pattern: <i>CustomerName_ServiceRequestJobID</i>. • For example configlets, see EVC (AutoPick Service Instance Name), page 3-168, EVC (Pseudowire Core Connectivity, User-Provided Service Instance Name), page 3-170, and EVC (Local Core Connectivity, User-Provided Service Instance Name), page 3-183. • This attribute is not displayed for IOS XR devices.

Table 3-10 Service Instance Details Attributes (continued)

Attribute	Description
Enable Trunk EFP	<p>Check the box to provide flexibility to make many Layer 2 flow points within one interface. This attribute appears only when the EVC check box is checked in Direct Connect Links section. Trunk EFP supports only ASR920, ASR902 and ASR903 IOS device. Usage notes:</p> <ul style="list-style-type: none"> • If the check box is checked, user will get “service instance trunk <id> ethernet” command. • If the check box is checked, the Match Inner and Outer Tags are disabled. • If the check box is checked, Autopick Outer VLAN and single value Outer VLAN are not allowed. • If the check box is checked, Outer VLAN ID only with ranges is allowed.
Service Instance Name	<p>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:</p> <ul style="list-style-type: none"> • 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 Provisioning does not generate the global ethernet evc evcname command in the device configuration generated by the service request.
AutoPick Bridge Domain/VLAN ID	<p>Check the box to have Prime Provisioning 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. Usage notes:</p> <ul style="list-style-type: none"> • AutoPick Bridge Domain/VLAN ID consumes a global VLAN ID on the device. • The bridge domain VLAN ID is picked from the existing Prime Provisioning VLAN pool.
Bridge Domain/VLAN ID	<p>If the AutoPick Bridge Domain/VLAN ID check box is unchecked, enter an appropriate value in the Bridge Domain/VLAN ID field.</p> <p>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.</p> <p>When a VLAN ID is manually allocated, Prime Provisioning verifies the VLAN ID to see if it lies within Prime Provisioning’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 Provisioning prompts you to allocate a different VLAN ID. If the VLAN ID lies outside of the Prime Provisioning VLAN ID pool, Prime Provisioning does not perform any verification about whether the VLAN ID allocated. The operator must ensure the VLAN ID is available.</p> <p>During Manual entry and on automatic update using VLAN pool, enter the VLAN ID ranges between 1-65535 for L3VPN/MPLS services, instead of the range 1 - 4094, because IOS-XR devices allow configurable VLAN id ranges between the range 1-65535.</p>

Table 3-10 Service Instance Details Attributes (continued)

Attribute	Description
AutoPick Bridge Domain/VLAN ID Secondary N-PE	<p>Check the box to have Prime Provisioning 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.</p> <p>Usage notes:</p> <ul style="list-style-type: none"> • This attribute is only applicable in the case of a dual-homed ring (a ring that terminates on two different N-PEs). Prime Provisioning 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 Provisioning allocates the bridge domain VLAN IDs from both primary and secondary N-PE access domains. When both are in the same Access Domain, Prime Provisioning 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 Provisioning VLAN pool. • This attribute is not displayed for IOS XR devices.
Bridge Domain/VLAN ID Secondary N-PE	<p>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.</p>
Match	<p>Choose an encapsulation type from the drop-down list. The choices are:</p> <ul style="list-style-type: none"> • DOT1Q • DEFAULT • UNTAGGED • PRIORITY TAGGED • DOT1AD <p>Selecting Default as the match criteria disables the Outer VLAN ID and Outer VLAN Ranges fields on the page. If Default is the CE encapsulation type, Prime Provisioning shows another field for the UNI port type.</p>
Match Inner and Outer Tags	<p>Check the 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.</p>

Table 3-10 Service Instance Details Attributes (continued)

Attribute	Description
Inner VLAN ID and Outer VLAN ID	<p>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:</p> <ul style="list-style-type: none"> • You can specify single values, single ranges, multiples values, multiple ranges, or combinations of these. Examples: <ul style="list-style-type: none"> – 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.
Outer VLAN ID	<p>If the Match Inner and Outer Tags check box is unchecked, enter the outer VLAN tag in the Outer VLAN ID field. Usage notes:</p> <ul style="list-style-type: none"> • 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. • You may also have Prime Provisioning autopick the outer VLAN ID using the AutoPick Outer VLAN attribute.
Inner VLAN ID	<p>If the Match Inner and Outer Tags check box is checked, enter both the inner and outer VLAN tags in the Inner VLAN ID and Outer VLAN ID fields. Usage notes:</p> <ul style="list-style-type: none"> • The VLAN specified in Inner VLAN ID will be provisioned on the rest of the L2 access nodes (if the link has any), including the customer-facing UNI. • You may also have Prime Provisioning autopick the inner VLAN ID using the AutoPick Inner VLAN attribute.
AutoPick Outer VLAN	<p>Check the box to have Prime Provisioning 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. Usage notes:</p> <ul style="list-style-type: none"> • Use of the AutoPick Outer VLAN attribute requires that two elements have already been set up in Prime Provisioning. 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 2-39, and Resource Pools, page 2-43. • 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 Provisioning VLAN pool.

Table 3-10 Service Instance Details Attributes (continued)


Attribute	Description
AutoPick Inner VLAN	<p>Check the box to have Prime Provisioning autopick the Inner VLAN ID from a previously created Inner VLAN ID resource pool during service request creation. If this check box is unchecked, the operator will be prompted to specify an Inner VLAN ID during service request creation. Usage notes:</p> <ul style="list-style-type: none"> • Use of the AutoPick Inner VLAN attribute requires that two elements have already been set up in Prime Provisioning. 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 Inner 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 2-39, and Resource Pools, page 2-43. • AutoPick Inner VLAN can be used for interfaces that support EVC functionality. • AutoPick Inner VLAN consumes a VLAN ID on the interface that supports EVC. • The bridge domain VLAN ID is picked from the existing Prime Provisioning VLAN pool. <p> Note The Match Inner and Outer Tags check box should be enabled in the policy to make use of AutoPick Inner VLAN feature during service request creation.</p>
Rewrite Type	<p>Choose a type from the drop-down list. The choices are:</p> <ul style="list-style-type: none"> • Pop • Push • Translate <p>The subsequent attributes in the GUI change depending on the choice of Rewrite Type.</p>
	<p>If Pop is the Rewrite Type, two check boxes appear:</p> <ol style="list-style-type: none"> 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. <p>Note that if Pop Inner Tag is checked, Pop Outer Tag is automatically checked.</p>
	<p>If Push is the Rewrite Type, two text boxes appear:</p> <ol style="list-style-type: none"> 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.

Table 3-10 Service Instance Details Attributes (continued)

Attribute	Description
	<p>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.</p> <ol style="list-style-type: none"> 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. <ul style="list-style-type: none"> – 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. <ul style="list-style-type: none"> – 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.

Table 3-11 Standard UNI Attributes

Attributes	Description
N-PE/U-PE Information, Interface Name	These fields display the PE device and interface name selected in previous steps. These fields are read-only
Encapsulation	<p>Choose a type from the drop-down list. The choices are:</p> <ul style="list-style-type: none"> • 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. <p>This attribute allows you to deploy different types of UNI encapsulation on different links of a service. Usage notes:</p> <ul style="list-style-type: none"> • 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.
PE/UNI Interface Description	Enter a description for the interface, if desired.
UNI Shutdown	Check the 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).
VLAN Translation	<p>Specify the type of VLAN translation for the service request by clicking the appropriate radio button. The choices are:</p> <ul style="list-style-type: none"> • 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.

Table 3-11 Standard UNI Attributes (continued)

Attributes	Description
	<p>Usage notes:</p> <ul style="list-style-type: none"> • 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: <ul style="list-style-type: none"> – 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: <ul style="list-style-type: none"> – CE VLAN—Provide a value between 1 and 4096. – Auto Pick—Check this check box to have Prime Provisioning 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.
N-PE Pseudo-wire on SVI	<p>Check the box to have Prime Provisioning generate forwarding commands under SVIs (switch virtual interfaces). By default, this check box is not checked. In this case, Prime Provisioning generates forwarding commands under the service instance.</p> <p>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.</p>

Table 3-11 Standard UNI Attributes (continued)

Attributes	Description
	<p>Usage notes:</p> <ul style="list-style-type: none"> 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 Provisioning 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 3-166 and EVC (Pseudowire Core Connectivity, no Bridge Domain, no Pseudowire on SVI), page 3-167. For additional information on the N-PE Pseudo-wire on SVI attribute, see the corresponding coverage in the EVC policy section in the section Interface Attributes Window, page 3-68. The N-PE Pseudo-wire on SVI attribute is not supported for IOS XR devices. All the xconnect commands are configured on L2 subinterfaces.
AutoPick Bridge Group Name	<p>Check the box to have Prime Provisioning 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:</p> <ul style="list-style-type: none"> 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. If the AutoPick Bridge Group Name check box is checked and if the DCPL property is set to true, then the auto pick bridge group for L2 services will be displayed as “bridge group <VFI-NAME>” for EVC VPLS core types.
AutoPick Bridge Domain/VLAN ID	<p>Check the box to have Prime Provisioning 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:</p> <ul style="list-style-type: none"> AutoPick Bridge Domain/VLAN ID consumes a global VLAN ID on the device. The bridge domain VLAN ID is picked from the existing Prime Provisioning 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.

Table 3-11 Standard UNI Attributes (continued)

Attributes	Description
Bridge Domain/VLAN ID	<p>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:</p> <ul style="list-style-type: none"> • If AutoPick Bridge Domain/VLAN ID is checked, this field is non-editable. • When a VLAN ID is manually allocated, Prime Provisioning verifies the VLAN ID to see if it lies within Prime Provisioning'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 Provisioning prompts you to allocate a different VLAN ID. If the VLAN ID lies outside of the Prime Provisioning VLAN ID pool, Prime Provisioning 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.
AutoPick Bridge Domain Name	<p>Check the box to have Prime Provisioning 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:</p> <ul style="list-style-type: none"> • 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.
Bridge Domain Name	<p>If the AutoPick Bridge Domain Name check box is unchecked, enter a bridge domain name in the Bridge Domain Name text field. Usage notes:</p> <ul style="list-style-type: none"> • 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. <ul style="list-style-type: none"> – Create L2 services with autopick bridge domain name, with the format of L2-X or L3-X, where X will be the range between 1-60000 for the newly created service request. For example, if X is 4001, "bridge-domain name generates as "L2-40001", or "L3-40001." – As soon as the starting range value is set through the DCPL property for the first service request, Prime Provisioning will consider the set value as a starting range value for autopick bridge domain name, Automatically, for the next service request onwards the value is incremented with +1. – If the default value for bridgeDomainName property is zero, you can set the range between 1 -60000. Prime Provisioning will increment the bridge domain value from the set value.

Table 3-11 Standard UNI Attributes (continued)

Attributes	Description
Use BVI	<p>Check the box to select a bridge virtual interface (BVI) to provide pseudowire access into an L3VPN. When the Use BVI check box is checked, the Select BVI Interface attribute appears, which provides a drop-down list of available BV Is configured on the device. Usage notes:</p> <ul style="list-style-type: none"> • The Use BVI attribute is only supported for IOS XR devices. (It is equivalent to the N-PE Pseudo-wire on SVI attribute that is supported for IOS devices.) • Note the following prerequisites for using the Use BVI attribute: <ul style="list-style-type: none"> – In order to use this attribute, you must have previously configured L3VPN services on the device. – Such L3VPN services must have created BVI interfaces. (These interfaces are what provides pseudowire access into the L3VPN.) – Furthermore, you must have performed a manual Collect config task for the corresponding N-PE devices in Prime Provisioning so that the L2VPN service would be aware of the BVI interfaces that were configured in the L3VPN. • For example configlets for this feature, see EVC (Pseudowire Core Connectivity, Pseudowire Service with BVI), page 3-181.
Use Split Horizon	<p>Check the box to enable split horizon with bridge domain. Usage notes:</p> <ul style="list-style-type: none"> • 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.
Use PseudoWireClass	<p>Check the box to enable the selection of a pseudowire class. This attribute is unchecked by default. Usage notes:</p> <ul style="list-style-type: none"> • The pseudowire class name is used for provisioning pw-class commands on IOS and IOS XR devices. See Creating and Modifying Pseudowire Classes, page 3-15 for additional information on pseudowire class support. • 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 Provisioning. • The Use PseudoWireClass attribute is only available if the MPLS core connectivity type was set as PSEUDOWIRE in the Service Options window (see Defining the EVC Ethernet Policy, page 3-20). • 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.

Table 3-11 Standard UNI Attributes (continued)

Attributes	Description
L2VPN Group Name	<p>Choose one of the following from the drop-down list:</p> <ul style="list-style-type: none"> • ISC • VPNSC <p>Usage notes:</p> <ul style="list-style-type: none"> • This attribute is used for provisioning the L2VPN group name on IOS XR devices. • 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 3-19. • The L2VPN Group Name attribute is not available if the MPLS core connectivity type was set as VPLS in the Service Options window (see Defining the EVC Ethernet Policy, 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.
E-Line Name	<p>Enter the point-to-point (p2p) E-line name. Usage notes:</p> <ul style="list-style-type: none"> • If no value is specified for the E-Line Name, Prime Provisioning autogenerates a default name as follows: <ul style="list-style-type: none"> – For PSEUDOWIRE core connectivity type, the format is: <i>DeviceName--VC_ID</i> – For LOCAL core connectivity type, the format is: <i>DeviceName--VLAN_ID</i> <p>If the default name is more than 32 characters, the device names are truncated.</p> • The E-Line Name attribute is not available if the MPLS core connectivity type was set as VPLS in the Service Options window (see Defining the EVC Ethernet Policy, 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.

EVC ATM-Ethernet Interworking Service Attributes

This section describes policy and service request attributes for EVC ATM-Ethernet services:

- [EVC ATM-Ethernet Interworking Policy Attributes, page 3-92](#)
- [EVC ATM-Ethernet Interworking Service Request Attributes, page 3-106](#)

EVC ATM-Ethernet Interworking Policy Attributes

This section provides reference tables for attributes available in the EVC ATM-Ethernet policy workflow:

- [Service Options Window, page 3-93](#)

- [ATM Interface Attributes Window, page 3-96](#)
- [EVC Attributes Window, page 3-96](#)
- [Interface Attributes Window, page 3-101](#)

**Note**

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).

Service Options Window

[Table 3-12](#) describes the attributes in the Service Options Window of the EVC ATM-Interworking policy workflow.

Table 3-12 *Service Options*

Attribute	Description
CE Directly Connected to EVC	<p>Check the box if the CEs are directly connected to the N-PE. This check box is not checked by default. Usage notes:</p> <ul style="list-style-type: none"> • 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 Provisioning behavior. There is no change in NPC implementation to support EVC functionality.
All Links Terminate on EVC	<p>Check the box if all links need to be configured with EVC features. This check box is not checked by default. Usage notes:</p> <ul style="list-style-type: none"> • 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. • 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 Provisioning policy types (L2VPN or VPLS) can be used instead of EVC.
All L2 Access Links default to EVC UNI	<p>Check the box to enable EVC syntax configuration on all access devices (U-PE and PE-AGG) throughout the circuit. This shows up in service request as EVC-related attributes for all of these device types. If this attribute is not enabled, in the service request EVC service-related syntax will only be available for N-PE devices.</p>

Table 3-12 Service Options (continued)

Attribute	Description
MPLS Core Connectivity Type	<p>Choose an MPLS core connectivity type from the drop-down list. The core option supports MPLS only. There is no L2TPv3 support for this service. The choices are:</p> <ul style="list-style-type: none"> • 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. <ul style="list-style-type: none"> Local connect supports the following scenarios: <ul style="list-style-type: none"> – 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.

Table 3-12 Service Options (continued)

Attribute	Description
Configure With Bridge Domain	<p>Check the 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.</p> <ul style="list-style-type: none"> • PSEUDOWIRE as the MPLS Core Connectivity Type. There are two cases: <ul style="list-style-type: none"> A. With EVC: <ul style="list-style-type: none"> – 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: <ul style="list-style-type: none"> – 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. <p>Only pseudowires can be either configured directly under service instance of the corresponding EVC-capable interface or under SVIs associated to the bridge domain.</p> <ul style="list-style-type: none"> • LOCAL as the MPLS Core Connectivity Type: <ul style="list-style-type: none"> – 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 Provisioning allows only point-to-point local connects without bridge domain.
Split Horizon	<p>Check the box to enable split horizon with bridge domain. Usage notes:</p> <ul style="list-style-type: none"> • The Use Split Horizon attribute is disabled by default. • The Use Split Horizon attribute can be used only when the Configure With Bridge Domain check box is checked (enabled). • When 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.
Static Pseudowire (Autopick MPLS Labels)	<p>Choose a type. The choices are:</p> <ul style="list-style-type: none"> • All Dynamic—Labels will be allocated dynamically during provisioning. No static labels will be added into the configlet. • All Static—Labels will be allocated statically during provisioning. Every segment in a multi-segment pseudowire will have static labels assigned to it on per-segment basis. • Defaults—Prime Provisioning will automatically determine whether or not to apply static labels based on the core type of the segment. It will do this on a per segment basis. <p>This attribute only supported for MPLS Core Connectivity Types of PSEUDOWIRE or VPLS.</p>

Table 3-12 Service Options (continued)

Attribute	Description
Configure Pseudowire Segment(s)	<p>Check the box to enable ability to configure pseudowire classes on a per segment basis in the service request based on this policy. Usage notes:</p> <ul style="list-style-type: none"> The Configure Pseudowire Segment(s) attribute is only applicable for MPLS core connectivity types of PSEUDOWIRE and VPLS. With a VPLS core type, the attribute shows up in the Service Options window of the Policy Editor. With a PSEUDOWIRE core type, the attribute shows up in the Interface Attributes window in the block of other pseudowire-related attributes. The Configure Pseudowire Segment(s) attribute is used in conjunction with the Static Pseudowire (Autopick MPLS Labels) attribute to configure the individual segments within a multi-segment pseudowire to be either dynamic or static. This allows you to override the default behavior of Prime Provisioning. A segment can be a TP tunnel, a TE tunnel, or an LDP (dynamic) core. The configuration is done subsequently in the service request based on the policy. When setting up the links in the service request, you can independently assign Pseudowire classes to ends of the segments of multi-segment pseudowires. For information on attaching pseudowire classes to links. The Configure Pseudowire Segment(s) attribute is not currently supported in EVC ATM-Ethernet Interworking policies and service requests.

ATM Interface Attributes Window

Table 3-13 describes the attributes in the ATM Interface Attributes Window of the EVC ATM-Interworking policy workflow.

Table 3-13 ATM Interface Attributes

Description	Attribute
Transport Mode	<p>Choose the transport mode from the drop-down list. The choices are:</p> <ul style="list-style-type: none"> VP—Virtual path mode. This is the default. VC—Virtual circuit mode.
ATM Encapsulation	Choose the ATM encapsulation from the drop-down list. The only available option is AAL5SNAP.

EVC Attributes Window

Table 3-14 describes the attributes in the EVC Attributes Window of the EVC ATM-Interworking policy workflow. EVC attributes are organized under the following categories:

- Service Attributes**
- VLAN Match Criteria.** 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 Provisioning 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.

- **VLAN Rewrite Criteria.** 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.

Table 3-14 EVC Attributes

Description	Attribute
Service Attributes	
AutoPick Service Instance ID	<p>Check the 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 Provisioning link attributes during service request creation, Prime Provisioning will prompt the operator to specify the service instance ID. Usage notes:</p> <ul style="list-style-type: none"> • 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 Provisioning 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.
AutoPick Service Instance Name	<p>Check the box to have Prime Provisioning autogenerated a service instance name when you create a service request based on the policy. The autogenerated value is in the following pattern: <i>CustomerName_ServiceRequestJobID</i>. If the check box is unchecked, then you can enter a value during service request creation.</p>
Enable PseudoWire Redundancy	<p>Check the box to enable pseudowire redundancy (alternative termination device) under certain conditions. Enable Pseudo Wire Redundancy is only available if the MPLS Core Connectivity Type was set as PSEUDOWIRE in the Service Options window (see Service Options Window, page 3-93).</p>
AutoPick VC ID	<p>Check the box to have Prime Provisioning 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. When AutoPick VC ID is checked, Prime Provisioning allocates a VC ID for pseudowires from the Prime Provisioning-managed VC ID resource pool.</p>

Table 3-14 EVC Attributes (continued)

Description	Attribute
AutoPick Bridge Domain/VLAN ID	<p>Check the box to have Prime Provisioning 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:</p> <ul style="list-style-type: none"> • AutoPick Bridge Domain/VLAN ID consumes a global VLAN ID on the device. • The bridge domain/VLAN ID is picked from the existing Prime Provisioning VLAN pool. Once the VLAN ID is assigned in the service request, Prime Provisioning makes the VLAN ID unavailable for subsequent service requests. • In the case of manual VLAN ID allocation, Prime Provisioning does not manage the VLAN ID if the ID lies outside the range of an Prime Provisioning-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 Provisioning-managed VLAN pool and the VLAN ID is already in use in the access domain, Prime Provisioning displays an error message indicating that the VLAN ID is in use. <p>For additional information on Access VLAN IDs, see Note on Access VLAN IDs, page 3-100.</p>
VLAN Matching Criteria Attributes	
Match Inner and Outer Tags	<p>Check the 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 Ranges attribute to appear in the EVC Attribute window.</p>
Inner VLAN Ranges	<p>Check the 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.</p>
Outer VLAN Ranges	<p>Check the 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.</p>
AutoPick Outer VLAN	<p>Check the box to have Prime Provisioning 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. Usage notes:</p> <ul style="list-style-type: none"> • Use of the AutoPick Outer VLAN attribute requires that two elements have already been set up in Prime Provisioning. 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 2-39, and Resource Pools, page 2-43. • 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 Provisioning VLAN pool.

Table 3-14 EVC Attributes (continued)


Description	Attribute
AutoPick Inner VLAN	<p>Check the box to have Prime Provisioning autopick the Inner VLAN ID from a previously created Inner VLAN ID resource pool during service request creation. If this check box is unchecked, the operator will be prompted to specify an Inner VLAN ID during service request creation. Usage notes:</p> <ul style="list-style-type: none"> • Use of the AutoPick Inner VLAN attribute requires that two elements have already been set up in Prime Provisioning. 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 Inner 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 2-39, and Resource Pools, page 2-43. • AutoPick Inner VLAN can be used for interfaces that support EVC functionality. • AutoPick Inner VLAN consumes a VLAN ID on the interface that supports EVC. <p>The bridge domain VLAN ID is picked from the existing Prime Provisioning VLAN pool.</p> <p> Note The Match Inner and Outer Tags check box should be enabled in the policy to make use of AutoPick Inner VLAN feature during service request creation.</p>
VLAN Rewrite Criteria Attributes	
Pop Outer	Check the 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.
Pop Inner	Check the 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.
Push Outer	<p>Check the 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:</p> <ul style="list-style-type: none"> • 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 Provisioning-managed VLAN ID pools.

Table 3-14 EVC Attributes (continued)

Description	Attribute
Push Inner	<p>Check the 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:</p> <ul style="list-style-type: none"> • 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 Provisioning-managed VLAN ID pools.
Translate Outer	<p>Check the 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 3-15.</p>
Translate Inner	<p>Check the 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 3-15.</p>

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 Provisioning by means of locally-significant VLAN pools. But once a VLAN ID is assigned for a link in the service request, Prime Provisioning 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 Provisioning 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.

Table 3-15 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

Table 3-15 VLAN Translation Summary Table

Type	Match Outer Tag	Match Inner Tag	Translate Outer Tag	Translate Inner Tag	Push Outer Tag
2:1	True	True	Yes	No	N/A
2:2	True	True	Yes	Yes	N/A

**Note**

[Table 3-15](#) 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.

Interface Attributes Window

[Table 3-16](#) describes the attributes in the Interface Attributes Window of the EVC ATM-Interworking policy workflow. 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 [Defining the EVC ATM-Ethernet Interworking Policy, page 3-34](#)). 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.

Table 3-16 Interface Attributes

Attribute	Description
Standard UNI Port	<p>Check the 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.</p> <p>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.</p>
UNI Shutdown	Check the 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.
Keep Alive	Check the 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.
Link Media (optional)	Enter None, auto-select, rj45, or sfp.
Link Speed (optional)	Enter None, 10, 100, 1000, Auto, or nonegotiate.
Link Duplex (optional)	Enter None, Full, Half, or Auto.
Encapsulation	<p>Choose a type. The choices are:</p> <ul style="list-style-type: none"> • 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.
VLAN Translation	<p>Specify the type for this policy by clicking the appropriate radio button. The choices are:</p> <ul style="list-style-type: none"> • 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. <p>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 3-35.</p>

Table 3-16 Interface Attributes (continued)

Attribute	Description
Use PseudoWireClass	<p>Check the box to enable the selection of a pseudowire class. This attribute is unchecked by default.</p> <p>Usage notes:</p> <ul style="list-style-type: none"> The pseudowire class name is used for provisioning pw-class commands on IOS and IOS XR devices. See Creating and Modifying Pseudowire Classes, page 3-15 for additional information on pseudowire class support. 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 Provisioning. The Use PseudoWireClass attribute is only available if the MPLS core connectivity type was set as PSEUDOWIRE in the Service Options window (see Defining the EVC ATM-Ethernet Interworking Policy, page 3-34).
L2VPN Group Name	<p>Choose one of the following from the drop-down list:</p> <ul style="list-style-type: none"> ISC VPNSC <p>Usage notes:</p> <ul style="list-style-type: none"> This attribute is used for provisioning the L2VPN group name on IOS XR devices. 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 3-19. L2VPN Group Name is only applicable for IOS XR devices.
E-Line Name	<p>Specify the point-to-point (p2p) E-line name. Usage notes:</p> <ul style="list-style-type: none"> If no value is specified for the E-Line Name in either the policy or the service request based on the policy, Prime Provisioning autogenerates a default name as follows: <ul style="list-style-type: none"> For PSEUDOWIRE core connectivity type, the format is: <i>DeviceName--VC_ID</i> For LOCAL core connectivity type, the format is: <i>DeviceName--VLAN_ID</i> <p>If the default name is more than 32 characters, the device names are truncated.</p> E-Line Name is only applicable for IOS XR devices.

Table 3-16 Interface Attributes (continued)

Attribute	Description
N-PE Pseudo-wire on SVI	<p>Check the box to have Prime Provisioning generate forwarding commands under SVIs (switch virtual interfaces). By default, this check box is not checked. In this case, Prime Provisioning generates forwarding commands under the service instance.</p> <p>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.</p> <p>Usage notes:</p> <ul style="list-style-type: none"> • Prime Provisioning 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 3-166 and EVC (Pseudowire Core Connectivity, no Bridge Domain, no Pseudowire on SVI), page 3-167. • 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 3-17 shows various use cases for hybrid configuration for EVC service requests. [Move outside table]
Use Existing ACL Name	<p>Check the box if you want to assign your own named access list to the port. By default, this check box is not checked and Prime Provisioning automatically assigns a MAC-based ACL on the customer facing UNI port, based on values you enter in UNI MAC addresses.</p>
Port-Based ACL Name	<p>Enter a Port-Based ACL Name (if you checked the Use Existing ACL Name check box).</p> <p>Prime Provisioning 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.</p>
UNI MAC addresses	<p>Enter one or more Ethernet 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.</p>

Table 3-16 Interface Attributes (continued)

Attribute	Description
UNI Port Security	<p>Check the box if you want to provision port security-related CLIs to the UNI port by controlling the MAC addresses that are allowed to go through the interface.</p> <ul style="list-style-type: none"> • For Maximum Number of MAC address, enter the number of MAC addresses allowed for port security. • For Aging, enter the length of time the MAC address can stay on the port security table. • For Violation Action, choose what action will occur when a port security violation is detected: <ul style="list-style-type: none"> – 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. • In the Secure MAC Addresses field, enter one or more Ethernet MAC addresses.
Enable Storm Control	<p>Check the 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.</p>
Protocol Tunnelling	<p>Check the 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:</p> <ul style="list-style-type: none"> • Enable cdp—Enable Layer 2 tunnelling on Cisco Discover Protocol (CDP). • cdp shutdown threshold—Enter the number of packets per second to be received before the interface is shut down. • cdp drop threshold—Enter the number of packets per second to be received at which point the interface will start dropping CDP packets. • Enable vtp—Enable Layer 2 tunnelling on VLAN Trunk Protocol (VTP). • vtp shutdown threshold—Enter the number of packets per second to be received before the interface is shut down. • vtp drop threshold—Enter the number of packets per second to be received at which point the interface will start dropping VTP packets. • Enable stp—Enable Layer 2 tunnelling on Spanning Tree Protocol (STP). • stp shutdown threshold—Enter the number of packets per second to be received before the interface is shut down. • stp drop threshold—Enter the number of packets per second to be received at which point the interface will start dropping STP packets. • Recovery Interval—Enter the amount of time, in seconds, to wait before recovering a UNI port.

Table 3-16 Interface Attributes (continued)

Attribute	Description
MTU Size	<p>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 Provisioning 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.</p> <p>In Cisco Prime Provisioning 6.3, different platforms support different ranges.</p> <ul style="list-style-type: none"> 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 Provisioning 6.3 uses 9216 in both cases. For the Cisco 7600 SVI (interface VLAN), the MTU size is 1500 to 9216.

Table 3-17 Use Cases for Hybrid Configuration for EVC Service Requests

Use Bridge Domain	EVC	N-PE Pseudowire on SVI	CLIs Generated
True	True	True	<ul style="list-style-type: none"> xconnect under VLAN interface. Service instance under main interface.
True	True	False	<ul style="list-style-type: none"> xconnect under service instance. Service instance under main interface.
False	True	N/A	<ul style="list-style-type: none"> 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.

EVC ATM-Ethernet Interworking Service Request Attributes

This section describes attributes available in the EVC ATM-Ethernet Interworking service request workflow:

- [Table 3-18, “Pseudowire Core Connectivity Attributes,” on page 107](#)
- [Table 3-19, “Local Core Connectivity Attributes,” on page 109](#)
- [Table 3-20, “Service Instance Details Attributes,” on page 111](#)
- [Table 3-21, “Standard UNI Attributes,” on page 115](#)
- [Table 3-22, “ATM UNI Attributes,” on page 119](#)

Table 3-18 Pseudowire Core Connectivity Attributes

Attribute	Description
Job ID, SR ID	These 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 Provisioning database holds within the editing flow of the service request.
Policy	This 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.
Select VPN	<p>Click to choose a VPN for use with this service request. The Select VPN window appears with the VPNs defined in the system.</p> <p>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.</p> <ol style="list-style-type: none"> 1. Choose a VPN Name in the Select column. <p>You may also use the New VPN Details section of the window to create a new VPN “on the fly.” This window provides a subset of the usual VPN creation features. Use the supplied fields to name the new VPN, select/create the customer, and so on. For more information about creating VPNs, see Setting Up Logical Inventory, page 2-55.</p> 2. Click Select. <p>The EVC Service Request Editor window appears with the VPN name displayed.</p>
AutoPick VC ID	Check the box if you want Prime Provisioning 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 Provisioning allocates a VC ID for pseudowires from the Prime Provisioning-managed VC ID resource pool. In this case, the text field for the VC ID option is non-editable.
VC ID	<p>If AutoPick VC ID was unchecked, enter a VC ID in the VC ID field. Usage notes:</p> <ul style="list-style-type: none"> • 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 Provisioning verifies the VC ID to see if it lies within Prime Provisioning’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 Provisioning prompts you to allocate a different VC ID. If the VC ID lies outside of the Prime Provisioning VC ID pool, Prime Provisioning 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.

Table 3-18 Pseudowire Core Connectivity Attributes (continued)

Attribute	Description
Enable PseudoWire Redundancy	<p>Check the box to enable pseudowire redundancy (alternative termination device) under certain conditions. Usage notes:</p> <ul style="list-style-type: none"> • Enable Pseudo Wire Redundancy is only available if the MPLS Core Connectivity Type was set as PSEUDOWIRE in the Service Options window (see Defining the EVC ATM-Ethernet Interworking Policy, page 3-34). • Enabling this feature allows the user to do the following: <ul style="list-style-type: none"> – Configure two pseudowires between two direct links, or: – Add a backup peer such that pseudowires are configured between A–Z and A–Z'. In this case, the terminating links A, Z, and Z' must all be directly connected links. L2 access links are not supported as backup peers. • See Setting Up Pseudowire Redundancy and a Backup Peers, page 3-29, for more information on using this feature in service requests.
Backup PW VC ID	<p>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 above. The backup VC ID behaves the same as the VC ID of the primary pseudowire.</p>
Static Pseudowire (Autopick MPLS Labels)	<p>Choose a type. The choices are:</p> <ul style="list-style-type: none"> • All Dynamic—Labels will be allocated dynamically during provisioning. No static labels will be added into the configlet. • All Static—Labels will be allocated statically during provisioning. Every segment in a multi-segment pseudowire will have static labels assigned to it on per-segment basis. • Defaults—Prime Provisioning will automatically determine whether or not to apply static labels based on the core type of the segment. It will do this on a per segment basis. <p>This attribute only supported for MPLS Core Connectivity Types of PSEUDOWIRE or VPLS.</p>
Configure Bridge Domain	<p>Check the 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:</p> <ul style="list-style-type: none"> • With EVC: <ul style="list-style-type: none"> – 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: <ul style="list-style-type: none"> – 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. <p>Pseudowires can be configured either directly under service instance of the corresponding EVC-capable interface or under SVIs associated to the bridge domain.</p>

Table 3-18 Pseudowire Core Connectivity Attributes (continued)

Attribute	Description
Use Split Horizon	<p>Check the box to enable split horizon with bridge domain. Usage notes:</p> <ul style="list-style-type: none"> 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.
Description	Click the “Click here” link to enter a description label for the service request. This is useful for searching the Prime Provisioning database for the particular service request.

Table 3-19 Local Core Connectivity Attributes

Attribute	Description
Job ID, SR ID	These 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 Provisioning database holds within the editing flow of the service request.
Policy	This field is read-only. It displays the name of the policy on which the service request is based.
Select VPN	<p>Click to choose a VPN for use with this service request. The Select VPN window appears with the VPNs defined in the system.</p> <p>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.</p> <ol style="list-style-type: none"> Choose a VPN Name in the Select column. You may also use the New VPN Details section of the window to create a new VPN “on the fly.” This window provides a subset of the usual VPN creation features. Use the supplied fields to name the new VPN, select/create the customer, and so on. For more information about creating VPNs, see Setting Up Logical Inventory, page 2-55. Click Select. <p>The EVC Service Request Editor window appears with the VPN name displayed.</p>

Table 3-19 Local Core Connectivity Attributes (continued)

Attribute	Description
Configure Bridge Domain	<p>Check the box to determine bridge domain characteristics. Usage notes:</p> <ul style="list-style-type: none"> 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. See the following comments for details on how Prime Provisioning autogenerates the connect name. <p>Because the device only accepts a maximum of 15 characters for the connect name, the connect name is generated using the following format:</p> <p style="text-align: center;"><i>CustomerNameTruncatedToMaxPossibleCharacters_ServiceRequestJobID</i></p> <p>For example, if the customer name is NorthAmericanCustomer and the service request job ID is 56345, the autogenerated connect name would be NorthAmer_56345.</p> <p>The CLI generated would be:</p> <pre>connect NorthAmer_56345 ATM7/0/5 11 ATM7/0/4 18</pre> <p>In this case, 11 and 18 are service instance VPIs.</p> <ul style="list-style-type: none"> If the policy setting for Configure Bridge Domain is non-editable, the option in the service request will be read-only.
Use Split Horizon	<p>Check the box to enable split horizon with bridge domain. Usage notes:</p> <ul style="list-style-type: none"> 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.
Description	<p>Click the “Click here” link to enter a description label for the service request. A dialogue appears in which you can enter a description.</p>

Table 3-20 Service Instance Details Attributes

Attribute	Description
AutoPick Service Instance ID	<p>Check the 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. Usage notes:</p> <ul style="list-style-type: none"> 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 Provisioning 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.
Service Instance ID	<p>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.</p>
AutoPick Service Instance Name	<p>Check the box to specify that the service instance name will be autogenerated. If the check box is unchecked, you can specify the service instance name. Usage notes:</p> <ul style="list-style-type: none"> If the check box is checked, the Service Instance Name text field is disabled. The service instance name is autogenerated in the following pattern: <i>CustomerName_ServiceRequestJobID</i>. For example configlets, see EVC (No AutoPick Service Instance Name, No Service Instance Name), page 3-169, EVC (Pseudowire Core Connectivity, User-Provided Service Instance Name), page 3-170, and EVC (Local Core Connectivity, User-Provided Service Instance Name), page 3-183. This attribute is not displayed for IOS XR devices.
Service Instance Name	<p>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:</p> <ul style="list-style-type: none"> 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 Provisioning does not generate the global ethernet evc evcname command in the device configuration generated by the service request.
AutoPick Bridge Domain/VLAN ID	<p>Check the box to have Prime Provisioning 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. Usage notes:</p> <ul style="list-style-type: none"> AutoPick Bridge Domain/VLAN ID consumes a global VLAN ID on the device. The bridge domain VLAN ID is picked from the existing Prime Provisioning VLAN pool. This attribute is not displayed for IOS XR devices.

Table 3-20 Service Instance Details Attributes (continued)

Attribute	Description
Bridge Domain/VLAN ID	<p>If the AutoPick Bridge Domain/VLAN ID check box is unchecked, enter an appropriate value in the Bridge Domain/VLAN ID field.</p> <p>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.</p> <p>When a VLAN ID is manually allocated, Prime Provisioning verifies the VLAN ID to see if it lies within Prime Provisioning'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 Provisioning prompts you to allocate a different VLAN ID. If the VLAN ID lies outside of the Prime Provisioning VLAN ID pool, Prime Provisioning does not perform any verification about whether the VLAN ID allocated. The operator must ensure the VLAN ID is available.</p>
AutoPick Bridge Domain/VLAN ID Secondary N-PE	<p>Check the box to have Prime Provisioning 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. Usage notes:</p> <ul style="list-style-type: none"> • This attribute is only applicable in the case of a dual-homed ring (a ring that terminates on two different N-PEs). Prime Provisioning 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 Provisioning allocates the bridge domain VLAN IDs from both primary and secondary N-PE access domains. When both are in the same Access Domain, Prime Provisioning 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 Provisioning VLAN pool. • This attribute is not displayed for IOS XR devices.
Bridge Domain/VLAN ID Secondary N-PE	<p>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.</p>
Match Inner and Outer Tags	<p>Check the 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.</p>

Table 3-20 Service Instance Details Attributes (continued)

Attribute	Description
Match Inner and Outer Tags	<p>Check the box to enter the inner and outer VLAN tags in the Inner VLAN ID and Outer VLAN ID fields. Usage notes:</p> <ul style="list-style-type: none"> • You can specify single values, single ranges, multiples values, multiple ranges, or combinations of these. Examples: <ul style="list-style-type: none"> – 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.
Outer VLAN ID	<p>If the Match Inner and Outer Tags check box is unchecked, enter the outer VLAN tag in the Outer VLAN ID field.</p> <ul style="list-style-type: none"> • 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. • You may also have Prime Provisioning autopick the outer VLAN ID.
Inner VLAN ID	<p>If the Match Inner and Outer Tags check box is checked, enter both the inner and the outer VLAN tags in the Inner VLAN ID and Outer VLAN ID fields. Usage notes:</p> <ul style="list-style-type: none"> • The VLAN specified in Inner VLAN ID will be provisioned on the rest of the L2 access nodes (if the link has any), including the customer-facing UNI. • You may also have Prime Provisioning autopick the inner VLAN ID using the AutoPick Inner VLAN attribute.
AutoPick Outer VLAN	<p>Check the box to have Prime Provisioning 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. Usage notes:</p> <ul style="list-style-type: none"> • Use of the AutoPick Outer VLAN attribute requires that two elements have already been set up in Prime Provisioning. 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 2-39, and Resource Pools, page 2-43. • 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 Provisioning VLAN pool.

Table 3-20 Service Instance Details Attributes (continued)


Attribute	Description
AutoPick Inner VLAN	<p>Check the box to have Prime Provisioning autopick the Inner VLAN ID from a previously created Inner VLAN ID resource pool during service request creation. If this check box is unchecked, the operator will be prompted to specify an Inner VLAN ID during service request creation. Usage notes:</p> <ul style="list-style-type: none"> • Use of the AutoPick Inner VLAN attribute requires that two elements have already been set up in Prime Provisioning. 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 Inner 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 2-39, and Resource Pools, page 2-43. • AutoPick Inner VLAN can be used for interfaces that support EVC functionality. • AutoPick Inner VLAN consumes a VLAN ID on the interface that supports EVC. <p>The bridge domain VLAN ID is picked from the existing Prime Provisioning VLAN pool.</p> <p> Note The Match Inner and Outer Tags check box should be enabled in the policy to make use of AutoPick Inner VLAN feature during service request creation.</p>
Rewrite Type	<p>Choose a type from the drop-down list. The choices are:</p> <ul style="list-style-type: none"> • Pop • Push • Translate <p>The subsequent attributes in the GUI change depending on the choice of Rewrite Type, as covered in the next steps.</p> <p>If Pop is the Rewrite Type, two check boxes appear:</p> <ol style="list-style-type: none"> 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. <p>Note that if Pop Inner Tag is checked, Pop Outer Tag is automatically checked.</p> <p>If Push is the Rewrite Type, two text boxes appear:</p> <ol style="list-style-type: none"> 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.

Table 3-20 Service Instance Details Attributes (continued)

Attribute	Description
	<p>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).</p> <ol style="list-style-type: none"> 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. <ul style="list-style-type: none"> – 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. <ul style="list-style-type: none"> – 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.

Table 3-21 Standard UNI Attributes

Attribute	Description
N-PE/U-PE Information, Interface Name	These fields display the PE device and interface name selected in previous steps. These fields are read-only.
Encapsulation	<p>Choose a type from the drop-down list. The choices are:</p> <ul style="list-style-type: none"> • 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. <p>This attribute allows you to deploy different types of UNI encapsulation on different links of a service. 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.</p>
PE/UNI Interface Description	Enter a description for the interface, if desired.
UNI Shutdown	Check the 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).

Table 3-21 Standard UNI Attributes (continued)

Attribute	Description
VLAN Translation	<p>Specify the type of VLAN Translation for the service request by clicking the appropriate radio button. The choices are:</p> <ul style="list-style-type: none"> • 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.
	<p>Usage notes:</p> <ul style="list-style-type: none"> • 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: <ul style="list-style-type: none"> – 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: <ul style="list-style-type: none"> – CE VLAN—Provide a value between 1 and 4096. – Auto Pick—Check this check box to have Prime Provisioning 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.
N-PE Pseudo-wire on SVI	<p>Check the box to have Prime Provisioning generate forwarding commands under SVIs (switch virtual interfaces). By default, this check box is not checked. In this case, Prime Provisioning generates forwarding commands under the service instance.</p> <p>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.</p>

Table 3-21 Standard UNI Attributes (continued)

Attribute	Description
	<p>Usage notes:</p> <ul style="list-style-type: none"> 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 Provisioning 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 3-166 and EVC (Pseudowire Core Connectivity, no Bridge Domain, no Pseudowire on SVI), page 3-167. For additional information on the N-PE Pseudo-wire on SVI attribute, see the corresponding coverage in the EVC policy section in the section Interface Attributes Window, page 3-101. 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.
Use PseudoWireClass	<p>Check the box to enable the selection of a pseudowire class. This attribute is unchecked by default. Usage notes:</p> <ul style="list-style-type: none"> The pseudowire class name is used for provisioning pw-class commands on IOS and IOS XR devices. See Creating and Modifying Pseudowire Classes, page 3-15 for additional information on pseudowire class support. 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 Provisioning. The Use PseudoWireClass attribute is only available if the MPLS core connectivity type was set as PSEUDOWIRE in the Service Options window (see Service Options Window, page 3-93).
AutoPick Bridge Group Name	<p>Check the box to have Prime Provisioning 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:</p> <ul style="list-style-type: none"> 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.

Table 3-21 Standard UNI Attributes (continued)

Attribute	Description
AutoPick Bridge Domain/VLAN ID	<p>Check the box to have Prime Provisioning 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:</p> <ul style="list-style-type: none"> • AutoPick Bridge Domain/VLAN ID consumes a global VLAN ID on the device. • The bridge domain VLAN ID is picked from the existing Prime Provisioning 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.
Bridge Domain/VLAN ID	<p>If the AutoPick Bridge Domain/VLAN ID check box is unchecked, enter an ID number in the text field. Usage notes:</p> <ul style="list-style-type: none"> • If AutoPick Bridge Domain/VLAN ID is checked, this field is non-editable. • When a VLAN ID is manually allocated, Prime Provisioning verifies the VLAN ID to see if it lies within Prime Provisioning'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 Provisioning prompts you to allocate a different VLAN ID. If the VLAN ID lies outside of the Prime Provisioning VLAN ID pool, Prime Provisioning 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. • If you enter the starting value in Bridge Domain Name, Prime Provisioning takes the entered value as a starting value for auto pick bridge domain name with the L2-X format (For example: "bridge-domain L2-40001"L2-40001) at the time of creating service requests and thus increments with +1 for the next service onwards. • For L3 VPN services, Prime Provisioning allows you to set the BD name with numeric value within a range of 1-60000. You can set the initial value using DCPL property. For example, If you enter the starting value in Bridge Domain Name, Prime Provisioning takes the property value as 40001 and chooses auto-pick BD name by generating the configuration as "bridge-domain L3-40001" and for the next new service request, the BD name is incremented to the numeric value by 1, that is L3-40002, L3-40003.
L2VPN Group Name	<p>Choose one of the following from the drop-down list:</p> <ul style="list-style-type: none"> • ISC • VPNSC <p>Usage notes:</p> <ul style="list-style-type: none"> • This attribute is used for provisioning the L2VPN group name on IOS XR devices. • 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 3-19. • L2VPN Group Name is only applicable for IOS XR devices.

Table 3-21 Standard UNI Attributes (continued)

Attribute	Description
E-Line Name	<p>Enter the point-to-point (p2p) E-line name. Usage notes:</p> <ul style="list-style-type: none"> If no value is specified for the E-Line Name, Prime Provisioning autogenerates a default name as follows: <ul style="list-style-type: none"> For PSEUDOWIRE core connectivity type, the format is: <i>DeviceName--VC_ID</i> For LOCAL core connectivity type, the format is: <i>DeviceName--VLAN_ID</i> <p>If the default name is more than 32 characters, the device names are truncated.</p> <ul style="list-style-type: none"> E-Line Name is only applicable for IOS XR devices.

Table 3-22 ATM UNI Attributes

Attribute	Description
Transport Mode	<p>Choose the Transport Mode from the drop-down list. The choices are:</p> <ul style="list-style-type: none"> VP—Virtual path mode. This is the default. VC—Virtual circuit mode.
ATM Encapsulation	<p>Choose the ATM Encapsulation from the drop-down list. The choice is:</p> <ul style="list-style-type: none"> AAL5SNAP
ATM VCD/Sub-Interface #	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.
ATM VPI	To specify the ATM virtual path identifier (VPI), enter a value in the ATM VPI field. The value can be from 0 to 255.
ATM VCI	To specify the ATM virtual channel identifier (VCI), a value in the ATM VCI field. The value can be from 32 to 65535.
UNI Shutdown	Check the 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).
Use Existing PW Class	<p>Check the box to enable the selection of a pseudowire class. This attribute is unchecked by default. Usage notes:</p> <ul style="list-style-type: none"> 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 Provisioning 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.

Table 3-22 ATM UNI Attributes (continued)

Attribute	Description
N-PE Pseudo-wire on SVI	<p>Check the box to have Prime Provisioning generate forwarding commands under SVIs (switch virtual interfaces). By default, this check box is not checked. In this case, Prime Provisioning generates forwarding commands under the service instance.</p> <p>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.</p>
	<p>Usage notes:</p> <ul style="list-style-type: none"> • 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 Provisioning 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 3-166 and EVC (Pseudowire Core Connectivity, no Bridge Domain, no Pseudowire on SVI), page 3-167. • For additional information on the N-PE Pseudo-wire on SVI attribute, see the corresponding coverage in the EVC policy section in the section Interface Attributes Window, page 3-101. • 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.
PW Tunnel Selection	<p>Check the 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:</p> <ul style="list-style-type: none"> • 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.
Interface Tunnel	<p>If you checked the PW Tunnel Selection check box, enter the TE tunnel ID in the Interface Tunnel text field. Prime Provisioning 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 Provisioning does not check the validity of the tunnel ID number. That is, Prime Provisioning does not verify the existence of the tunnel.</p>

Table 3-22 ATM UNI Attributes (continued)

Attribute	Description
AutoPick Bridge Domain/VLAN ID	<p>Check the box to have Prime Provisioning 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:</p> <ul style="list-style-type: none"> • AutoPick Bridge Domain/VLAN ID consumes a global VLAN ID on the device. • The bridge domain VLAN ID is picked from the existing Prime Provisioning VLAN pool.
Bridge Domain/VLAN ID	<p>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:</p> <ul style="list-style-type: none"> • If AutoPick Bridge Domain/VLAN ID is checked, this field is non-editable. • When a VLAN ID is manually allocated, Prime Provisioning verifies the VLAN ID to see if it lies within Prime Provisioning'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 Provisioning prompts you to allocate a different VLAN ID. If the VLAN ID lies outside of the Prime Provisioning VLAN ID pool, Prime Provisioning does not perform any verification about whether the VLAN ID allocated. The operator must ensure the VLAN ID is available.

Sample Configlets

This section provides sample configlets for EVC service provisioning in Prime Provisioning. It contains the following subsections:

- [Overview, page 3-124](#)
- [ERS \(EVPL\) \(Point-to-Point\), page 3-125](#)
- [ERS \(EVPL\) \(Point-to-Point, UNI Port Security\), page 3-126](#)
- [ERS \(EVPL\) \(1:1 VLAN Translation\), page 3-127](#)
- [ERS \(EVPL\) \(2:1 VLAN Translation\), page 3-128](#)
- [ERS \(EVPL\) and EWS \(EPL\) \(Local Connect on E-Line\), page 3-131](#)
- [EWS \(EPL\) \(Point-to-Point\), page 3-133](#)
- [EWS \(EPL\) \(Point-to-Point, UNI Port Security, BPDU Tunneling\), page 3-134](#)
- [EWS \(EPL\) \(Hybrid\), page 3-136](#)
- [ATM over MPLS \(VC Mode\), page 3-141](#)
- [ATM over MPLS \(VP Mode\), page 3-142](#)
- [Frame Relay over MPLS, page 3-144](#)
- [Frame Relay \(DLCI Mode\), page 3-145](#)
- [VPLS \(Multipoint, ERMS/EVP-LAN\), page 3-146](#)
- [VPLS \(Multipoint, EMS/EP-LAN\), BPDU Tunneling\), page 3-147](#)
- [EVC \(Pseudowire Core Connectivity, UNI Port Security\), page 3-148](#)
- [EVC \(Pseudowire Core Connectivity, UNI, without Port Security, with Bridge Domain\), page 3-149](#)
- [EVC \(Pseudowire Core Connectivity, UNI, and Pseudowire Tunneling\), page 3-150](#)
- [EVC \(Pseudowire Core Connectivity, With Pseudowire Headend Support\), page 3-151](#)
- [EVC \(Pseudowire Core Connectivity, Without Pseudowire Headend Support\), page 3-152](#)
- [EVC \(VPLS Core Connectivity, UNI Port Security\), page 3-153](#)
- [EVC \(VPLS Core Connectivity, no UNI Port Security\), page 3-154](#)
- [EVC \(VPLS Core Connectivity, With E-Tree Role, Communication between the Spokes of Different Hubs\), page 3-159](#)
- [EVC \(VPLS Core Connectivity, With E-Tree Role, Communication between the Spokes of Same HUB\), page 3-160](#)
- [EVC \(VPLS Core Connectivity, EFPs in same UNI, Switchport, CPT\), page 3-161](#)
- [EVC \(VPLS Core Connectivity, EFPs in Different UNI, Service Instance, CPT\), page 3-162](#)
- [EVC \(VPLS Core Connectivity, With MTU Attribute Support for VPLS Services\), page 3-163](#)
- [EVC \(Local Connect Core Connectivity, UNI Port Security\), page 3-164](#)
- [EVC \(Local Connect Core Connectivity, UNI, no Port Security, Bridge Domain\), page 3-165](#)
- [EVC \(Pseudowire Core Connectivity, Bridge Domain, Pseudowire on SVI\), page 3-166](#)
- [EVC \(Pseudowire Core Connectivity, no Bridge Domain, no Pseudowire on SVI\), page 3-167](#)
- [EVC \(No AutoPick Service Instance Name, No Service Instance Name\), page 3-169](#)
- [EVC \(Pseudowire Core Connectivity, User-Provided Service Instance Name\), page 3-170](#)

- EVC (Pseudowire Core Connectivity, Pseudowire Redundancy, “A” – “Z”), page 3-171
- EVC (Pseudowire Core Connectivity, Pseudowire Redundancy, “A”, “Z”, and “Z ”), page 3-172
- EVC (Pseudowire Core Connectivity, Pseudowire Redundancy, “A”, “Z”, and “Z ”, where “Z” = “Z ”), page 3-173
- EVC (Pseudowire Core Connectivity, Service Instance Syntax on L2 Access Nodes), page 3-174
- EVC (Pseudowire Core Connectivity, Mixture of Switchport and Service Instance Syntax on L2 Access Nodes, Push Outer Enabled), page 3-175
- EVC (Pseudowire Core Connectivity, Service Instance Syntax on L2 Access Nodes, Push Both Enabled), page 3-177
- EVC (Pseudowire Core Connectivity, Static Pseudowire, IOS Device), page 3-178
- EVC (Pseudowire Core Connectivity, Static Pseudowire, IOS Device, Pseudowire Redundancy), page 3-179
- EVC (Pseudowire Core Connectivity, Static Pseudowire, IOS Device, Bridge Domain Disabled), page 3-180
- EVC (Pseudowire Core Connectivity, Pseudowire Service with BVI), page 3-181
- EVC (Pseudowire Core Connectivity, Static Pseudowire, OAM Class Set in DCPL Property), page 3-182
- EVC (Local Core Connectivity, User-Provided Service Instance Name), page 3-183
- EVC (VPLS Core Connectivity, User-Provided Service Instance Name), page 3-184
- EVC (ATM-Ethernet Interworking, Pseudowire Core Connectivity, Point-to-Point Circuit), page 3-185
- EVC (ATM-Ethernet Interworking, Pseudowire Core Connectivity, Multipoint Circuit), page 3-186
- EVC (ATM-Ethernet Interworking, Pseudowire Core Connectivity, Pseudowire Redundancy), page 3-187
- EVC (ATM-Ethernet Interworking, Local Core Connectivity, Point-to-Point Circuit), page 3-188
- EVC (ATM-Ethernet Interworking, Local Core Connectivity, Multipoint Circuit), page 3-189
- EVC (ATM-Ethernet Interworking, Local Core Connectivity, Multipoint Circuit), page 3-190
- EVC (ATM-Ethernet Interworking, Local Core Connectivity, Point-to-Point Circuit), page 3-191
- EVC (ATM-Ethernet Interworking, Pseudowire Core Connectivity, End-to-End Circuit), page 3-192
- EVC (ATM-Ethernet Interworking, Pseudowire Core Connectivity, Multipoint Circuit), page 3-193
- EVC (ATM-Ethernet Interworking, Local Core Connectivity, Point-to-Point Circuit), page 3-194
- EVC (ATM-Ethernet Interworking, Pseudowire Core Connectivity, End-to-End Circuit, with Bridge Domain), page 3-195
- EVC (ATM-Ethernet Interworking, Pseudowire Core Connectivity, End-to-End Circuit, with Bridge Domain), page 3-196
- EVC (ATM-Ethernet Interworking, Pseudowire Core Connectivity, End-to-End Circuit, no Bridge Domain), page 3-197

Overview

The configlets provided in this section show the CLIs generated by Prime Provisioning 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 Provisioning 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
<pre>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 bpdupfilter 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</pre>	<pre>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</pre>

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	N-PE
<pre> 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 bpdupfilter 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 </pre>	<pre> 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 </pre>

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	N-PE
<pre> ! 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 bpdudfilter 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 </pre>	<pre> 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 </pre>

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
<pre> 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 bpdupfilter 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 </pre>	<pre> 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 </pre>

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	N-PE
<pre> ! 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 ! </pre>	<pre> ! 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 ! ! </pre>

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	N-PE
<pre> ! 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 bpdudfilter enable ! interface GigabitEthernet1/0/1 no ip address switchport switchport trunk encapsulation dot1q switchport mode trunk switchport trunk allowed vlan 3200 ! </pre>	<pre> ! 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 bpdudfilter enable ! interface Vlan3300 no ip address xconnect 192.169.105.40 7502 encapsulation mpls no shutdown ! </pre>

Comments

None.

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
	<pre>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 c1-test-12-crs1-1--0--559 interface GigabitEthernet0/0/0/2.559 interface GigabitEthernet0/0/0/4.559 ! ! !</pre>

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)	<p>Template Content:</p> <pre>interface Loopback0 description LocalLoopbackAddress=\$L2VPNLocalLoopback LocalHostName=\$L2VPNLocalHostName RemoteLoopbackAddress=\$L2VPNRemoteLoopback RemoteHostName=\$L2VPNRemoteHostName</pre> <p>Configlets:</p> <pre>interface Loopback0 description LocalLoopbackAddress= 192.169.105.40 LocalHostName=c1-test-12-7600-2 RemoteLoopbackAddress=192.169.105.80 RemoteHostName= c1-test-12-7600-4</pre>

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	N-PE
<pre> 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 </pre>	<pre> 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 </pre>

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
<pre> 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 bpdupfilter 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 </pre>	<pre> 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 </pre>

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
<pre> 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 bpdupfilter 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 </pre>	<pre> 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 </pre>

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
<pre> 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 </pre>	<pre> 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 </pre>

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	N-PE
<pre> ! 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 ! </pre>	<pre> ! 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-12-12404-2--1000 interface GigabitEthernet0/3/1/1.700 neighbor 192.169.105.30 pw-id 1000 pw-class PW_AD7-AD3_Cutsomer2 ! </pre>

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 Provisioning-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	N-PE
<pre> ! 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 bpdupfilter enable ! interface GigabitEthernet1/0/1 no ip address switchport switchport trunk encapsulation dot1q switchport mode trunk switchport trunk allowed vlan 3201 ! </pre>	<pre> ! 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 bpdupfilter enable ! interface Vlan3301 no ip address xconnect 192.169.105.40 7502 encapsulation mpls no shutdown ! </pre>

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)	<pre>interface ATM2/0.34234 point-to-point pvc 213/423 l2transport encapsulation aa15 xconnect 99.99.4.99 89025 encapsulation mpls</pre>

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)	<pre>pseudowire-class ISC-pw-tunnel-123 encapsulation mpls preferred-path interface tunnel123 disable-fallback ! interface ATM2/0 atm pvp 131 12transport xconnect 99.99.4.99 89024 pw-class ISC-pw-tunnel-123</pre>

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)	<pre>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 !</pre>

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)	<pre>interface Serial1/1 exit ! connect C1_89001 Serial1/1 135 12transport xconnect 99.99.4.99 89001 encapsulation mpls</pre>

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)	<pre>pseudowire-class ISC-pw-dynamic-default encapsulation l2tpv3 ip local interface Loopback10 ip dfbit set ! interface Serial13/2 encapsulation frame-relay exit ! connect ISC_1054 Serial13/2 86 l2transport xconnect 10.9.1.1 1054 encapsulation l2tpv3 pw-class ISC-pw-dynamic-default</pre>

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	N-PE
<pre> 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 bpdupfilter 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 </pre>	<pre> 12 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 </pre>

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 BPDU's 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
<pre> 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 bpdupfilter enable </pre>	<pre> 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 </pre>

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
<pre> 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 bpdupfilter 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 </pre>	<pre> interface GigabitEthernet4/0/1 no shut service instance 10 ethernet encapsulation dot1q 500 rewrite ingress tag push dot1q 555 symmetric xconnect 192.169.105.20 505 encapsulation mpls </pre>

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
<pre>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 bpdudfilter 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</pre>	<pre>vlan 100 interface GigabitEthernet2/0/0 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</pre>

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)	<pre>pseudowire-class ISC-pw-tunnel-2147 encapsulation mpls preferred-path interface Tunnel2147 disable-fallback interface GigabitEthernet4/0/0 service instance 1 ethernet encapsulation dot1q 11 second-dot1q 41 rewrite ingress tag pop 2 symmetric xconnect pw-class ISC-pw-tunnel-2147</pre>

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 (Pseudowire Core Connectivity, With Pseudowire Headend Support)

Configuration

- Service: EVC/Metro Ethernet.
- Feature: EVC with pseudowire core connectivity, with pseudowire headend support.
- 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)	<pre>interface PW-Ether4909 no shutdown l2vpn xconnect group ISC p2p Demo_Eline_Head interface PW-Ether4909 neighbor ipv4 10.10.10.10 pw-id 1734</pre>

Comments

- When “Configure PWHE” check box is checked, the interface acts as a pseudowire-ether interface.

EVC (Pseudowire Core Connectivity, Without Pseudowire Headend Support)

Configuration

- Service: EVC/Metro Ethernet.
- Feature: EVC with pseudowire core connectivity, without pseudowire headend support.
- 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)	<pre>interface PW-Ether4909.294 12transport description EVC-JOBID:2 encapsulation dot1q 294 no shutdown 12vpn bridge group bgrname bridge-domain bdrname interface PW-Ether4909.294 neighbor 192.18.156.7 pw-id 3651</pre>

Comments

- When “Configure PWHE” check box is unchecked, the interface of pseudowire acts similar to that of gigabit interface where sub-interfaces can be configured.

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	N-PE
<pre> 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 bpdupfilter 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 </pre>	<pre> 12 vfi attest-226 manual vpn id 226 neighbor 192.169.105.20 encapsulation mpls vlan 200 bridge-domain 200 split-horizon interface GigabitEthernet4/0/1 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 </pre>

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	N-PE
<pre> 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 bpdupfilter 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 </pre>	<pre> 12 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 </pre>

Comments

- UNI on U-PE.
- The rewrite operation translates the incoming VLAN tag 500 to two tags, 222 and 41.

EVC DOT1Q Encapsulation

Configuration

- Service: EVC/Metro Ethernet.
- Feature: EVC with PW/LC/VPLS core connectivity with UNI service instance ID 507 and outer VLAN as 508. Bridge-Domain is used as 508.
- Additional Properties: Encapsulation is Dot1Q at UNI.
- Device configuration:
 - The N-PE is a Cisco 7600 with IOS 15.3(3)S1.
 - Interface(s): Gi2/7.
 - The U-PE is a Cisco ME1200 with ME1200 OS Software Build 15.4-2.SN
 - Interface(s): Gi1/4 as UNI and Gi1/5 as Uplink interface

■ Sample Configlets

Configlets

U-PE (ME1200) (Sheet 1 of 2)		N-PE (7600)
UNI Interface	Uplink Interface	
<pre> <soapRequest soapAction="test#addECE" urlPath="SandinoSoap/evc"> <inputMsgDoc> <ece_configuration> <ece_id>507</ece_id> <control> <ingress_match> <uni_ports> <GigabitEthernet_1_UNI>>false</Gigabi tEthernet_1_UNI> <GigabitEthernet_2_UNI>>false</Gigabi tEthernet_2_UNI> <GigabitEthernet_3_UNI>>false</Gigabi tEthernet_3_UNI> <GigabitEthernet_4_UNI>>true</Gigabit Ethernet_4_UNI> <GigabitEthernet_5_UNI>>false</Gigabi tEthernet_5_UNI> <GigabitEthernet_6_UNI>>false</Gigabi tEthernet_6_UNI> </uni_ports> <outer_tag_match> <match_type> <c_tagged/> </match_type> <match_fields> <vlan_id_filter> <specific>508</specific> </vlan_id_filter> <inner_pcp> <val_any/> </inner_pcp> <inner_dei> <any/> </inner_dei> </match_fields> </outer_tag_match> <inner_tag_match> <match_type> <any/> </match_type> <match_fields> <vlan_id_filter> <specific>1</specific> </vlan_id_filter> <inner_pcp> <val_any/> </inner_pcp> <inner_dei> <any/> </inner_dei> </match_fields> </inner_tag_match> <mac_params> <smac_filter> <any/> </smac_filter> <dmac_filter> <any/> </dmac_filter> </mac_params> <frame_type> <any/> </frame_type> </ingress_match> <actions> </pre>	<pre> <soapRequest soapAction="test#addEVC" urlPath="SandinoSoap/evc "> <inputMsgDoc> <evcConfiguration> <instance>508</instance> <policer_id>1</policer_i d> <nni_vid>508</nni_vid> <internal_vid>508</inter nal_vid> <nni_ports> <GigabitEthernet_1_NNI>f alse</GigabitEthernet_1_ NNI> <GigabitEthernet_2_NNI>f alse</GigabitEthernet_2_ NNI> <GigabitEthernet_3_NNI>f alse</GigabitEthernet_3_ NNI> <GigabitEthernet_4_NNI>f alse</GigabitEthernet_4_ NNI> <GigabitEthernet_5_NNI>t rue</GigabitEthernet_5_N NI> <GigabitEthernet_6_NNI>f alse</GigabitEthernet_6_ NNI> </nni_ports> <learning>>true</learning > </evcConfiguration> </inputMsgDoc> </soapRequest> </pre>	<pre> interface GigabitEthernet2/7 service instance 507 ethernet description EVC-JOBID:7 encapsulation dot1q 508 </pre>

U-PE (ME1200) (continued) (Sheet 2 of 2)		
UNI Interface	Uplink Interface	N-PE (7600)
<pre> <direction> <both/> </direction> <rule_type> <both/> </rule_type> <tx_lookup> <vid_only/> </tx_lookup> <evc_id> <specific>508</specific> </evc_id> <policer_id> <none/> </policer_id> <tag_pop_count>0</tag_pop_count> <policy_id>0</policy_id> <class> <disabled/> </class> <drop_precedence> <disabled/> </drop_precedence> </actions> <egress_outer_tag> <mode> <disabled/> </mode> <vlan_id>4095</vlan_id> <pcp_mode> <classified/> </pcp_mode> <pcp_value>0</pcp_value> <dei_mode> <classified/> </dei_mode> <dei_value>0</dei_value> </egress_outer_tag> <egress_inner_tag> <type> <untagged/> </type> <vlan_id>4095</vlan_id> <pcp_mode> <classified/> </pcp_mode> <pcp_value>0</pcp_value> <dei_mode> <classified/> </dei_mode> <dei_value>0</dei_value> </egress_inner_tag> </control> </ece_configuration> </inputMsgDoc> </soapRequest> </pre>		

Comments

- UNI on U-PE.
- The rewrite operation translates the incoming VLAN tag 500 to two tags, 222 and 41.

EVC (VPLS Core Connectivity, With E-Tree Role, Communication between the Spokes of Different Hubs)

Configuration

- Service: EVC/Metro Ethernet.
- Feature: EVC with VPLS core connectivity, with E-Tree Role, Communication between the Spokes of Different Hubs.
- Device configuration:
 - Cisco 7600 with IOS Version 15.4(1)S as NPE(LEAF and ROOT).



Note

Other devices that support similar configurations are Cisco ASR9k with IOS XR 4.3.1, Cisco ME3600 with IOS Version 15.3(3)S.

Configlets

NPE(LEAF)	NPE(ROOT)
<pre>bridge-domain 3511 12 vfi syjhjf manual vpn id 1649 vlan 3511 exit interface GigabitEthernet7/0/11 service instance 214 ethernet description EVC-JOBID:15 encapsulation dot1q 1124 bridge-domain 3511 split-horizon interface Vlan3511 no ip address description EVC-JOBID:15 xconnect vfi syjhjf no shutdown</pre>	<pre>bridge-domain 2019 12 vfi egysd manual vpn id 5773 vlan 2019 exit interface GigabitEthernet7/0/12 service instance 633 ethernet description EVC-JOBID:16 encapsulation dot1q 2075 bridge-domain 2019 interface Vlan2019 no ip address description EVC-JOBID:16 xconnect vfi egysd no shutdown</pre>

Comments

- When the E-Tree role of the device is set as root, the Split Horizon attribute is hidden from policy and it is controlled internally using E Tree role.
- When the E-Tree role of the device is set as leaf, the Split Horizon attribute appears in the policy and prevents the communication between the spokes of different HUBs.

EVC (VPLS Core Connectivity, With E-Tree Role, Communication between the Spokes of Same HUB)

Configuration

- Service: EVC/Metro Ethernet.
- Feature: EVC with VPLS core connectivity, with E-Tree Role, Communication between the Spokes of Same HUB.
- Device configuration:
 - Cisco 7600 with IOS Version 15.4(1)S as NPE (LEAF and ROOT).



Note

Other devices that support similar configurations are Cisco ASR9k with IOS XR 4.3.1 and Cisco ME3600 with IOS Version 15.3(3)S.

Configlets

LEAF	ROOT
<pre>12 vfi gwded manual vpn id 1028 neighbor 192.18.156.7 encapsulation mpls</pre>	<pre>12 vfi fhrerr manual vpn id 6338 neighbor 192.18.156.7 encapsulation mpls no-split-horizon</pre>

Comments

- when the E-Tree role of the device is set as root, no-split horizon attribute appears in the policy. This attribute is controlled internally using E Tree role.
- when the E-Tree role is set as leaf, split horizon attribute does not appear in the policy and it prevents the communication between spokes of the same HUB.

EVC (VPLS Core Connectivity, EFPs in same UNI, Switchport, CPT)

Configuration

- Service: EVC/Metro Ethernet.
- Feature: Interface VLAN is not supported from CPT device. EVC-PW cannot be used in Single Home Ring scenarios as there is no support for second interface through “Interface VLAN”. As an alternative, EVC-VPLS is to be used in DHR and SHR using “L2 VFI”.
- Device configuration:
 - The N-PE is a CPT 200 platform with the version as 15.2(1)SB1, with EFPs in same UNI.
 - The U-PE is a CPT50 Ring platform with the version as 15.2(20140201:215234), with switchport (non -flex) configuration.

Configlets

U-PE	N-PE
<pre>bridge-domain 822 mode p2p exit interface GigabitEthernet1/2 no ip address switchport switchport trunk encapsulation dot1q switchport mode trunk switchport trunk allowed vlan 822 interface TenGigabitEthernet56/46 no ip address switchport trunk encapsulation dot1q switchport mode trunk switchport trunk allowed vlan 822 keepalive no spanning-tree bpdupfilter enable switchport nonegotiate</pre>	<pre>bridge-domain 435 mode vpls ! interface GigabitEthernet7/0/17 no shutdown service instance 2311 ethernet description EVC-JOBID:1 encapsulation dot1q 543 bridge-domain 435 service instance 2312 ethernet description EVC-JOBID:1 encapsulation dot1q 522 bridge-domain 435 ! 12 vfi vpls-100 manual vpn id 533 bridge-domain 435 neighbor 1.1.1.1 encapsulation mpls neighbor 4.4.4.4 encapsulation mpls (optional backup circuit)!</pre>

Comments

- UNI on N-PE.
- UPE device with switch port configuration.

EVC (VPLS Core Connectivity, EFPs in Different UNI, Service Instance, CPT)

Configuration

- Service: EVC/Metro Ethernet.
- Feature: Interface VLAN is not supported from CPT device. EVC-PW cannot be used in Single Home Ring scenarios as there is no support for second interface through “Interface VLAN”. As an alternative, EVC-VPLS is to be used in DHR and SHR using “L2 VFI”.
- Device configuration:
 - The N-PE is a CPT 200 platform with the version as 15.2(1)SB1, with EFPs in different UNI.
 - The U-PE is a CPT50 Ring platform with the version as 15.2(20140201:215234), with service instance(flex) configuration.

Configlets

U-PE	N-PE
<pre>bridge-domain 454 mode vpls exit interface GigabitEthernet56/24 service instance 554 ethernet no description encapsulation dot1q 454 bridge-domain 454 exit interface TenGigabitEthernet56/48 service instance 554 ethernet no description encapsulation dot1q 454 bridge-domain 454 exit</pre>	<pre>bridge-domain 658 mode vpls ! interface GigabitEthernet7/0/14 service instance 1544 ethernet description EVC-JOBID:1 encapsulation dot1q 412 bridge-domain 658 ! interface GigabitEthernet7/0/19 service instance 1545 ethernet description EVC-JOBID:1 encapsulation dot1q 722 bridge-domain 658 ! l2 vfi vpls-100 manual vpn id 533 bridge-domain 435 neighbor 2.2.2.2 encapsulation mpls neighbor 3.3.3.3 encapsulation mpls (opt- tional backup circuit) !</pre>

Comments

- UNI on N-PE.
- UPE device with service instance configuration.

EVC (VPLS Core Connectivity, With MTU Attribute Support for VPLS Services)

Configuration

- Service: EVC/Metro Ethernet.
- Feature: EVC with VPLS core connectivity, with MTU Attribute Support. MTU attribute provides support to provision MTU value for EVC VPLS services in Direct Access Links. Earlier this attribute was available only in Links with L2 Access.
- Device configuration:
- The MTU size provided is provisioned under interface VLAN CLI for IOS Version 15.3(3)S devices and under bridge-domain CLI for IOS XR Version 4.3.1 devices.

Configlets

IOS device CLI	IOS XR device CLI
<pre> bridge-domain 2252 12 vfi DEMO_VPN_MTU-37893 manual vpn id 37893 neighbor 171.16.150.57 encapsulation mpls vlan 2252 exit interface GigabitEthernet2/2 service instance 1942 ethernet description EVC-JOBID:1 encapsulation dot1q 3321 bridge-domain 2252 exit interface Vlan2252 no ip address description EVC-JOBID:1 mtu 2500 xconnect vfi DEMO_VPN_MTU-37893 no shutdown </pre>	<pre> interface GigabitEthernet0/1/0/13.1332 12transport description EVC-JOBID:1 encapsulation dot1q 1332 no shutdown 12vpn bridge group DEMO_CUST_MTU bridge-domain ISC-DEMO_VPN_MTU-37893 mtu 2500 interface GigabitEther- net0/1/0/13.1332 vfi DEMO_VPN_MTU-37893 neighbor 171.16.150.47 pw-id 37893 </pre>

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	N-PE
<pre> 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 bpdupfilter 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 </pre>	<pre> 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 </pre>

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
<pre>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 bpdupfilter 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</pre>	<pre>interface GigabitEthernet2/0/0 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 GigabitEthernet2/0/10 no shut service instance 15 ethernet encapsulation dot1q 24 rewrite ingress tag pop 1 symmetric bridge-domain 200 split-horizon</pre>

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	N-PE
<pre>vlan 452 exit ! interface FastEthernet1/0/10 no ip address switchport trunk allowed vlan add 452 ! interface FastEthernet1/0/13 no spanning-tree bpdupfilter enable switchport no keepalive no ip address switchport switchport trunk encapsulation dot1q switchport mode trunk switchport trunk allowed vlan 452 switchport nonegotiate</pre>	<pre>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</pre>

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
<pre>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</pre>	<pre>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</pre>

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
<pre>! 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</pre>	<pre>! vlan 3524 exit ! ethernet evc C1_1 ! interface GigabitEthernet7/0/0 service instance 3 ethernet C1_1 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</pre>

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	N-PE
<pre> ! 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 </pre>	<pre> ! interface GigabitEthernet7/0/2 service instance 43 ethernet encapsulation dot1q 566 xconnect 1.1.1.1 453366 encapsulation mpls </pre>

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 (Pseudowire Core Connectivity, User-Provided Service Instance Name)

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	N-PE
<pre> ! 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 </pre>	<pre> ! 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 </pre>

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 (Pseudowire Core Connectivity, Pseudowire Redundancy, “A” – “Z”)

Configuration

- EVC/Metro Ethernet.
- Feature: EVC with Pseudowire core connectivity, pseudowire redundancy (“A” – “Z”) with backup peer command on both ends.
- Device configuration:
 - The N-PE 1 is a Cisco 7600 with IOS version 12.2(33).
Interface(s): GigabitEthernet7/0/13.
 - The N-PE 2 is a Cisco 7600 with IOS version 12.2(33).
Interface(s): GigabitEthernet3/2.

Configlets

N-PE 1 (“A” End)	N-PE 2 (“Z” End)
<pre>pseudowire-class PrimeF-pwc-Vpn1-staticLabels encapsulation mpls protocol none interface GigabitEthernet7/0/13 service instance 6560 ethernet description EVC-JOBID:22 encapsulation dot1q 4018 xconnect 10.10.10.10 451341 encapsulation mpls manual pw-class PrimeF-pwc-Vpn1-staticLabels mpls label 16 5001 backup peer 10.10.10.10 333612 mpls label 18 8001</pre>	<pre>pseudowire-class PrimeF-pwc-Vpn1-staticLabels protocol none interface GigabitEthernet3/2 no shutdown service instance 5551 ethernet description EVC-JOBID:22 encapsulation dot1q 2551 xconnect 1.1.1.1 451341 encapsulation mpls manual pw-class PrimeF-pwc-Vpn1-staticLabels mpls label 5001 16 backup peer 1.1.1.1 333612 mpls label 8001 18</pre>

Comments

- None.

EVC (Pseudowire Core Connectivity, Pseudowire Redundancy, "A", "Z", and "Z'")

Configuration

- EVC/Metro Ethernet.
- Feature: EVC with Pseudowire core connectivity, pseudowire redundancy ("A", "Z", and "Z'") with backup peer command on "A" end only.
- Device configuration:
 - The N-PE 1 is a Cisco 7600 with IOS version 12.2(33).
Interface(s): GigabitEthernet7/0/13.
 - The N-PE 2 is a Cisco 7600 with IOS version 12.2(33).
Interface(s): GigabitEthernet7/0/18.
 - The N-PE 3 is a Cisco 7600 with IOS version 12.2(33).
Interface(s): GigabitEthernet3/2.

Configlets

N-PE 1 ("A" End)	N-PE 2 ("Z" End)
<pre>pseudowire-class PrimeF-pwc-Vpn1-staticLabels encapsulation mpls protocol none interface GigabitEthernet7/0/13 service instance 6560 ethernet description EVC-JOBID:22 encapsulation dot1q 4018 xconnect 10.10.10.10 451341 encapsulation mpls manual pw-class PrimeF-pwc-Vpn1-staticLabels mpls label 16 5001 backup peer 6.6.7.1 333612 mpls label 18 8001</pre>	<pre>pseudowire-class PrimeF-pwc-Vpn1-staticLabels encapsulation mpls protocol none interface GigabitEthernet7/0/18 service instance 7580 ethernet description EVC-JOBID:22 encapsulation dot1q 3486 xconnect 1.1.1.1 451341 encapsulation mpls manual pw-class PrimeF-pwc-Vpn1-staticLabels mpls label 5001 16</pre>
N-PE 3 ("Z'" End)	
<pre>pseudowire-class PrimeF-pwc-Vpn1-staticLabels protocol none interface GigabitEthernet3/2 no shutdown service instance 5551 ethernet description EVC-JOBID:22 encapsulation dot1q 2551 xconnect 1.1.1.1 333612 encapsulation mpls manual pw-class PrimeF-pwc-Vpn1-staticLabels mpls label 8001 18</pre>	

Comments

- None.

EVC (Pseudowire Core Connectivity, Pseudowire Redundancy, "A", "Z", and "Z'", where "Z'" = "Z'")

Configuration

- EVC/Metro Ethernet.
- Feature: EVC with Pseudowire core connectivity, pseudowire redundancy, with backup peer on "A" only, "Z" and "Z'" (prime) are the same device but two service instances created on separate interfaces.
- Device configuration:
 - The N-PE 1 is a Cisco 7600 with IOS version 12.2(33).
Interface(s): GigabitEthernet7/0/11.
 - The N-PE 2 is a Cisco 7600 with IOS version 12.2(33).
Interface(s): GigabitEthernet7/0/16, GigabitEthernet7/0/17.

Configlets

N-PE 1 ("A" End)	N-PE 2 ("Z" End)
<pre>interface GigabitEthernet7/0/11 service instance 4775 ethernet description EVC-JOBID:23 encapsulation dot1q 1985 xconnect 10.10.10.10 125412 encapsulation mpls manual pw-class PrimeF-pwc-Vpn1-staticLabels mpls label 16 5001 backup peer 10.10.10.10 333212 mpls label 18 5002</pre>	<pre>interface GigabitEthernet7/0/16 service instance 6987 ethernet description EVC-JOBID:23 encapsulation dot1q 2987 xconnect 1.1.1.1 125412 encapsulation mpls manual pw-class PrimeF-pwc-Vpn1-staticLabels mpls label 5001 16 interface GigabitEthernet7/0/17 service instance 6665 ethernet description EVC-JOBID:23 encapsulation dot1q 2665 xconnect 1.1.1.1 333212 encapsulation mpls manual pw-class PrimeF-pwc-Vpn1-staticLabels mpls label 5002 18</pre>

Comments

- None.

EVC (Pseudowire Core Connectivity, Service Instance Syntax on L2 Access Nodes)

Configuration

- EVC/Metro Ethernet.
- Feature: EVC with Pseudowire core connectivity, service instance syntax on L2 access node, EVC UNI enabled, configured with Bridge Domain disabled.
- Device configuration:
 - The U-PE is a Cisco 7600 with IOS version 15.2(4)S.
Interface(s): GigabitEthernet5/15, GigabitEthernet5/16.
 - The PE-AGG is a Cisco ASR903 with IOS version 15.2(4)S1a
Interface(s): GigabitEthernet0/5, GigabitEthernet0/6.
 - The N-PE is a Cisco 7600 with IOS version 15.2(4)S.
Interface(s): GigabitEthernet2/0/15.

Configlets

U-PE	PE-AGG
<pre> bridge-domain 3600 exit vlan 3600 exit interface GigabitEthernet5/15 service instance 2321 ethernet encapsulation dot1q 3600 bridge-domain 3600 exit interface GigabitEthernet5/16 service instance 2321 ethernet encapsulation dot1q 3600 bridge-domain 3600 exit </pre>	<pre> bridge-domain 3600 exit interface GigabitEthernet0/5 service instance 400 ethernet encapsulation dot1q 3600 rewrite ingress tag pop 1 symmetric bridge-domain 3600 exit interface GigabitEthernet0/6 service instance 400 ethernet encapsulation dot1q 3600 rewrite ingress tag pop 1 symmetric bridge-domain 3600 exit </pre>
N-PE	
<pre> interface GigabitEthernet2/0/15 service instance 3700 ethernet description EVC-JOBID:6 encapsulation dot1q 3600 xconnect 171.16.150.56 45000 encapsulation mpls </pre>	

Comments

- The bridge domain VLAN input for the U-PE and PE-AGG will be reused from the outer VLAN (if push is not enabled at the UNI).

EVC (Pseudowire Core Connectivity, Mixture of Switchport and Service Instance Syntax on L2 Access Nodes, Push Outer Enabled)

Configuration

- EVC/Metro Ethernet.
- Feature: EVC with Pseudowire core connectivity, mixture of switchport and service instance syntax on L2 access nodes, EVC UNI enabled, push outer enabled, configured with bridge domain disabled.
- Device configuration:
 - The U-PE is a Cisco ME3600 with IOS Version 15.3(1)S.
Interface(s): GigabitEthernet5/15, GigabitEthernet5/16.
 - The PE-AGG is a Cisco 7600 with IOS version 15.2(4)S
Interface(s): GigabitEthernet0/5, GigabitEthernet0/6.
 - The N-PE is a Cisco 7600 with IOS version 15.2(4)S.
Interface(s): GigabitEthernet2/0/15.

Configlets

U-PE	PE-AGG
<pre> bridge-domain 3600 exit vlan 3600 exit interface GigabitEthernet5/15 switchport trunk allowed vlan none switchport mode trunk service instance 2321 ethernet encapsulation dot1q 1500 rewrite ingress tag push dot1q 3600 symmetric bridge-domain 3600 exit interface GigabitEthernet5/16 switchport trunk allowed vlan none switchport mode trunk service instance 2321 ethernet encapsulation dot1q 3600 bridge-domain 3600 exit </pre>	<pre> bridge-domain 3600 exit vlan 3600 exit interface GigabitEthernet0/5 switchport trunk allowed vlan add 3600 exit interface GigabitEthernet0/6 service instance 400 ethernet encapsulation dot1q 3600 rewrite ingress tag pop 1 symmetric bridge-domain 3600 exit </pre>
N-PE	
<pre> interface GigabitEthernet2/0/15 service instance 3700 ethernet description EVC-JOBID:6 encapsulation dot1q 3600 second-dot1q 1500 xconnect 171.16.150.56 45000 encapsulation mpls </pre>	

Comments

- In case push outer is enabled at the UNI, the same will be matched at the up-link of the U-PE, PE-AGG, and N-PE interfaces
- In the PE-AGG device, GigabitEthernet0/5 is a non-EVC interface. Hence, switchport gets provisioned into the interface. This is identified automatically during provisioning.

EVC (Pseudowire Core Connectivity, Service Instance Syntax on L2 Access Nodes, Push Both Enabled)

Configuration

- EVC/Metro Ethernet.
- Feature: EVC with Pseudowire core connectivity, service instance syntax on L2 access nodes, EVC UNI enabled, encapsulation untagged, Push Both enabled, configured with bridge domain disabled.
- Device configuration:
 - The U-PE is a Cisco 7600 with IOS version 15.2(4)S.
Interface(s): GigabitEthernet5/15, GigabitEthernet5/16.
 - The PE-AGG is a Cisco ASR901 with IOS Version 15.2(2)SNH1
Interface(s): GigabitEthernet0/5, GigabitEthernet0/6.
 - The N-PE is a Cisco 7600 with IOS version 15.2(4)S
Interface(s): GigabitEthernet2/0/15.

Configlets

U-PE	PE-AGG
<pre>bridge-domain 3600 exit vlan 3600 exit interface GigabitEthernet5/15 service instance 2321 ethernet encapsulation untagged rewrite ingress tag push dot1q 3600 second-dot1q 3700 symmetric bridge-domain 3600 exit interface GigabitEthernet5/16 service instance 2321 ethernet encapsulation dot1q 3600 bridge-domain 3600 exit</pre>	<pre>bridge-domain 3600 exit interface GigabitEthernet0/5 service instance 400 ethernet encapsulation dot1q 3600 rewrite ingress tag pop 1 symmetric bridge-domain 3600 exit interface GigabitEthernet0/6 service instance 400 ethernet encapsulation dot1q 3600 rewrite ingress tag pop 1 symmetric bridge-domain 3600 exit</pre>
N-PE	
<pre>interface GigabitEthernet2/0/15 service instance 3700 ethernet description EVC-JOBID:6 encapsulation dot1q 3600 second-dot1q 3700 xconnect 171.16.150.56 45000 encapsulation mpls</pre>	

Comments

- In case push both is enabled at the UNI (which is applicable for encapsulation UNTAGGED), only the outer VLAN will be matched at the U-PE up-link and PE-AGG interfaces, whereas both outer and inner VLAN will be matched at the N-PE.

EVC (Pseudowire Core Connectivity, Static Pseudowire, IOS Device)

Configuration

- EVC/Metro Ethernet.
- Feature: EVC with Pseudowire core connectivity, IOS device, static pseudowire (all static), configured with bridge domain disabled.
- Device configuration:
 - N-PE 1 is a Cisco 7600 with IOS Version 15.2(4)S.
Interface(s): GigabitEthernet7/0/17.
 - N-PE 2 is a Cisco ASR9K with IOS-XR Version 4.2.2.
Interface(s): GigabitEthernet0/1/0/25.

Configlets

N-PE 1	N-PE 2
<pre>interface GigabitEthernet7/0/17 service instance 3801 ethernet description EVC-JOBID:66 encapsulation dot1q 3801 xconnect 192.168.101.1 5466 encapsulation mpls manual mpls label 21 4018</pre>	<pre>interface GigabitEthernet0/1/0/25.3801 l2transport encapsulation dot1q 3801 no shutdown l2vpn xconnect group ISC p2p isc-cl-test-l2-asr9006-1--5466 interface GigabitEthernet0/1/0/25.3801 neighbor 1.1.1.1 pw-id 5466 mpls static label local 4018 remote 21</pre>

Comments

- None.

EVC (Pseudowire Core Connectivity, Static Pseudowire, IOS Device, Pseudowire Redundancy)

Configuration

- EVC/Metro Ethernet.
- Feature: EVC with pseudowire core connectivity, IOS device, static pseudowire (all static), configured with bridge domain enabled, pseudowire redundancy, pseudowire class disabled.
- Device configuration:
 - N-PE 1 is a Cisco 7600 with IOS Version 15.2(4)S.
Interface(s): GigabitEthernet4/0/14.
 - N-PE 2 is a Cisco 7600 with IOS Version 15.2(4)S.
Interface(s): GigabitEthernet2/0/19.

Configlets

N-PE 1	N-PE 2
<pre> bridge-domain 367 exit vlan 367 exit interface GigabitEthernet4/0/14 service instance 697 ethernet description EVC-JOBID:9 encapsulation dot1q 147 bridge-domain 367 interface Vlan367 no ip address description EVC-JOBID:9 xconnect 192.169.105.20 6589632 encapsulation mpls manual mpls label 8003 2102 backup peer 192.169.105.20 47851 mpls label 8004 2103 no shutdown </pre>	<pre> bridge-domain 23 exit vlan 23 exit interface GigabitEthernet2/0/19 service instance 398 ethernet description EVC-JOBID:9 encapsulation dot1q 434 bridge-domain 23 interface Vlan23 no ip address description EVC-JOBID:9 xconnect 192.169.105.10 6589632 encapsulation mpls manual mpls label 2102 8003 backup peer 192.169.105.10 47851 mpls label 2103 8004 no shutdown </pre>

Comments

- None.

EVC (Pseudowire Core Connectivity, Static Pseudowire, IOS Device, Bridge Domain Disabled)

Configuration

- EVC/Metro Ethernet.
- Feature: EVC with Pseudowire core connectivity, static pseudowire (all static), IOS device, configured with bridge domain disabled, pseudowire class enabled.
- Device configuration:
 - N-PE 1 is a Cisco 7600 with IOS Version 15.2(4)S.
Interface(s): GigabitEthernet4/0/14.
 - N-PE 2 is a Cisco ASR9K with IOS-XR Version 4.2.2.
Interface(s): GigabitEthernet0/1/0/25.

Configlets

N-PE 1	N-PE 2
<pre>pseudowire-class d-staticpw encapsulation mpls protocol none interface GigabitEthernet7/0/17 service instance 3801 ethernet description EVC-JOBID:66 encapsulation dot1q 3801 xconnect 192.168.101.1 5466 encapsulation mpls manual pw-class d-staticpw mpls label 21 4018</pre>	<pre>interface GigabitEthernet0/1/0/25.3801 l2transport encapsulation dot1q 3801 no shutdown l2vpn pw-class d-staticpw encapsulation mpls xconnect group ISC p2p isc-cl-test-l2-asr9006-1--5466 interface GigabitEthernet0/1/0/25.3801 neighbor 1.1.1.1 pw-id 5466 mpls static label local 4018 remote 21 pw-class d-staticpw</pre>

Comments

- None.

EVC (Pseudowire Core Connectivity, Pseudowire Service with BVI)

Configuration

- EVC/Metro Ethernet.
 - Feature: EVC with Pseudowire core connectivity, configured with bridge domain enabled, use BVI enabled.
 - Device configuration:
 - The N-PE is a Cisco ASR9K with IOS XR Version 4.2.2.
- Interface(s): GigabitEthernet0/1/0/0.

Configlets

N-PE

```
l2vpn
bridge group cisco
bridge-domain domain50
  Interface GigabitEthernet0/1/0/0.50
  routed interface bvi 20
  neighbor 1.2.3.4 pw-id 55
```

Comments

- As a prerequisite, an interface BVI should be created for the L3VPN service and a config-collect task should be performed for this ASR9K device.
- This feature is only applicable for ASR9K devices.

EVC (Pseudowire Core Connectivity, Static Pseudowire, OAM Class Set in DCPL Property)

Configuration

- EVC/Metro Ethernet.
- Feature: EVC with Pseudowire core connectivity, EVC N-PE enabled, pseudowire class enabled, OAM class set in DCPL property.
- Device configuration:
 - The N-PE 1 is a Cisco 7600 with IOS Version 15.2(4)S.
 - Interface(s): TenGigabitEthernet2/1.

Configlets

```

N-PE
-----
pseudowire-class d-static-oam-1
  encapsulation mpls
  protocol none
  status protocol notification static
d-static-oam-1
interface TenGigabitEthernet2/1
  service instance 456 ethernet
  description EVC-JOBID:4
  encapsulation dot1q 765
  xconnect 1w-class92.16.5.58 6544
encapsulation mpls manual p d-static-oam-1
  mpls label 4001 32
  
```

Comments

- EVC static pseudowire can also be provisioned with OAM class enabled. The prerequisite for this is that the OAM class needs to be created manually by the user, and the same should be provided as DCPL property “Provisioning\Service\fsm\SetStaticOamClassName” in Prime Provisioning.
- This is only applicable for IOS platforms.

EVC (Local Core Connectivity, User-Provided Service Instance Name)

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	N-PE
<pre> 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 </pre>	<pre> 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 </pre>

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 (VPLS Core Connectivity, User-Provided Service Instance Name)

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	N-PE
<pre> ! vlan 452 exit ! interface FastEthernet1/0/10 no ip address switchport trunk allowed vlan add 452 ! interface FastEthernet1/0/13 no spanning-tree bpdupfilter 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 </pre>	<pre> 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 </pre>

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)
<pre>! 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 !</pre>	<pre>! ethernet evc 1-3_51 ! interface GigabitEthernet4/0/2 no ip address 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 !</pre>

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)	N-PE 2 (Ethernet)
<pre>! 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</pre>	<pre>! 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</pre>

Comments

- None.

EVC (ATM-Ethernet Interworking, Pseudowire Core Connectivity, Pseudowire Redundancy)

Configuration

- EVC/ATM-Ethernet Interworking.
- Feature: EVC for ATM-Ethernet interworking with pseudowire core connectivity, with Pseudowire Redundancy. Pseudowire Redundancy support has been extended to ATM-Ethernet Interworking service type for both IOS and IOS-XR devices.
- Device configuration:
 - The A-END is an ATM LINK with IOS XR Version 4.3.1 devices.
Interface(s): ATM0/3/0/0.789, l2transport pvc 78/89
 - The Z-END is an ETHERNET LINK with IOS XR Version 4.3.1 devices.
Interface(s): GigabitEthernet0/1/0/12.809 l2transport.
 - The Z Backup - END is an ETHERNET LINK with IOS Version 15.3(3)S devices.
Interface(s): GigabitEthernet7/0/15 service instance 567 Ethernet.

Configlets

ATM LINK (IOS XR DEVICE): A - END	ETHERNET LINK (IOS-XR DEVICE): Z - END
<pre>interface ATM0/3/0/0.789 l2transport pvc 78/89 encapsulation aal5snap no shutdown l2vpn xconnect group ISC p2p D-ELINE-DEMO interface ATM0/3/0/0.789 neighbor ipv4 192.168.12.21 pw-id 6898 backup neighbor 10.10.10.10 pw-id 6899</pre>	<pre>interface GigabitEthernet0/1/0/12.809 l2transport description EVC-JOBID:15 encapsulation dot1q 809 rewrite ingress tag pop 1 symmetric no shutdown l2vpn xconnect group ISC p2p D-ELINE-DEMO interface GigabitEthernet0/1/0/12.809 neighbor ipv4 192.168.80.210 pw-id 6898</pre>

ETHERNET LINK (IOS DEVICE): Z Backup - END

```
interface GigabitEthernet7/0/15
  service instance 567 ethernet
  description EVC-JOBID:15
  encapsulation dot1q 567
  rewrite ingress tag pop 1 symmetric
xconnect 192.168.80.210 6899 encapsulation
mpls
```

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
<pre> ! interface ATM1/0/1 no shutdown ! interface ATM4/1/0 no shutdown ! interface ATM1/0/1.99 point-to-point pvc 99/99 12transport encapsulation aal0 ! interface ATM4/1/0.98 point-to-point pvc 98/98 12transport encapsulation aal0 ! connect ATM-to-ATM ATM1/0/1 99/99 ATM4/1/0 98/98 ! </pre>	

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
<pre> ! 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 ! </pre>	

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
<pre> ! 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 ! </pre>	

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
<pre>! 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</pre>	

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)	N-PE 2 (Ethernet)
<pre>! 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 !</pre>	<pre>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 ! ! !</pre>

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)
<pre>interface ATM4/1/0.8790 point-to-point pvc 150/3454 12transport encapsulation aal5snap xconnect 192.169.105.10 760 pw-class ISC-pw-tunnel-1</pre>	<pre>interface GigabitEthernet4/0/17.600 encapsulation dot1Q 600 xconnect 192.169.105.20 760 pw-class ISC-pw-tunnel-1</pre>

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
<pre>! 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</pre>	

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)
<pre>! 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</pre>	<pre>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 ! ! !</pre>

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)	N-PE 2 (Ethernet)
<pre>! 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 !</pre>	<pre>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 ! ! !</pre>

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)
<pre>! 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 !</pre>	<pre>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 ! ! ! !</pre>

Comments

- None.

EVPL(Priority Tagged to Tagged, "A" – "Z")

Configuration

- EVC/Metro Ethernet.
- Feature: EVC Priority Tagged to Untagged ("A" – "Z") .
- Device configuration:
 - The N-PE 1 is a Cisco ME3600 with IOS version 15.4(3)S.
Interface(s): GigabitEthernet0/1.
 - The N-PE 2 is a Cisco 7600 with IOS version 15.4(1)S.
Interface(s): GigabitEthernet3/2.

Configlets

N-PE 1 ("A" End)	N-PE 2 ("Z" End)
<pre> ethernet evc <Srv-Name-1> ! interface GigabitEthernet0/1 switchport trunk allowed vlan none switchport mode trunk service instance 1 ethernet <Srv-Name-1> encapsulation priority-tagged rewrite ingress tag pop 1 symmetric xconnect 7600-loopback ip <VCID> encaps mpls </pre>	<pre> ethernet evc <Srv-Name> ! interface TenGigabitEthernet1/1 service instance 100 ethernet <name> encapsulation dot1q 100 rewrite ingress tag pop 1 symmetric xconnect ME3600-loopback ip <VCID> encaps mpls </pre>

Comments

- None.

EVPL(Priority Tagged to Untagged, "A" – "Z")

Configuration

- EVC/Metro Ethernet.
- Feature: EVC Priority Tagged to Untagged ("A" – "Z").
- Device configuration:
 - The N-PE 1 is a Cisco ME3600 with IOS version 15.4(3)S.
Interface(s): GigabitEthernet0/1.
 - The N-PE 2 is a Cisco 7600 with IOS version 15.4(1)S.
Interface(s): GigabitEthernet3/2.

Configlets

N-PE 1 ("A" End)	N-PE 2 ("Z" End)
<pre> ethernet evc <Srv-Name-1> ! interface GigabitEthernet0/1 switchport trunk allowed vlan none switchport mode trunk service instance 1 ethernet <Srv-Name-1> encapsulation priority-tagged rewrite ingress tag pop 1 symmetric xconnect 7600-loopback ip <VCID> encap mpls </pre>	<pre> ethernet evc <Srv-Name> ! interface TenGigabitEthernet1/1 service instance 1 ethernet <name> encapsulation untagged xconnect ME3600-loopback ip <VCID> encap mpls </pre>

Comments

None.



Managing TDM-CEM Services (RAN Backhaul)

This chapter describes how to use Prime Provisioning to manage CEM classes, work with TDM-CEM policies, and manage TDM-CEM service requests. It contains the following sections:

- [Overview of RAN Backhaul Services, page 4-1](#)
- [Overview of the CEM TDM Service, page 4-3](#)
- [Prerequisites, page 4-4](#)
- [Managing CEM Classes, page 4-4](#)
- [Creating a TDM-CEM Policy, page 4-7](#)
- [Using Template Variables in TDM-CEM Services, page 4-11](#)
- [Managing TDM-CEM Service Requests, page 4-11](#)
- [Sample Configlets for TDM-CEM Services, page 4-21](#)

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

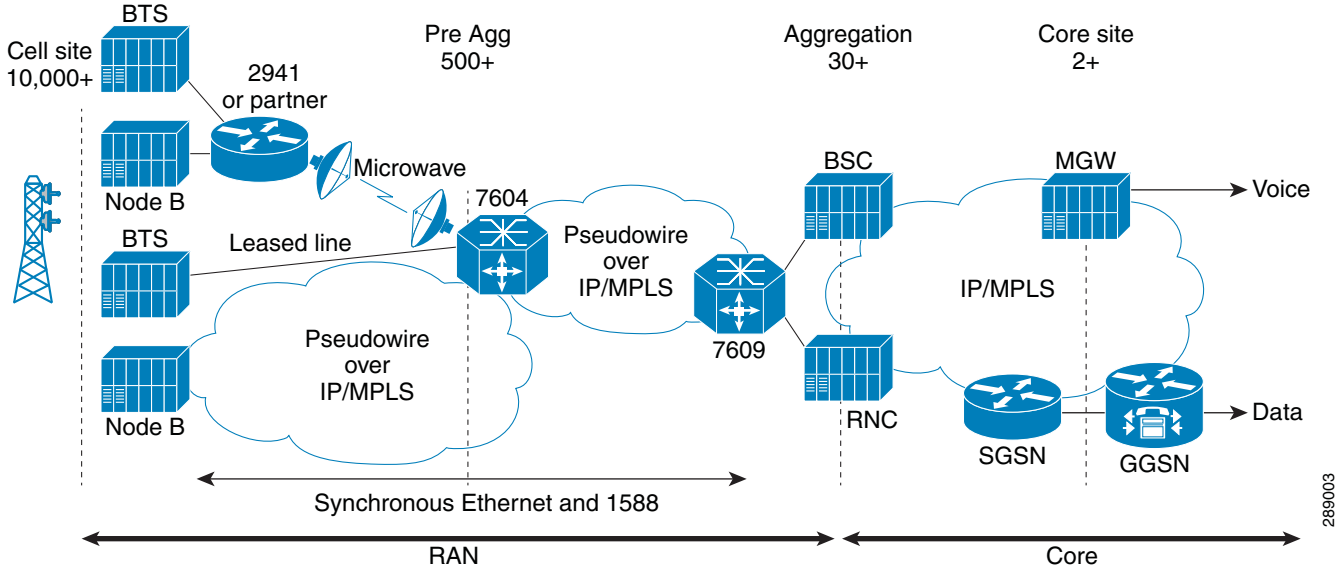
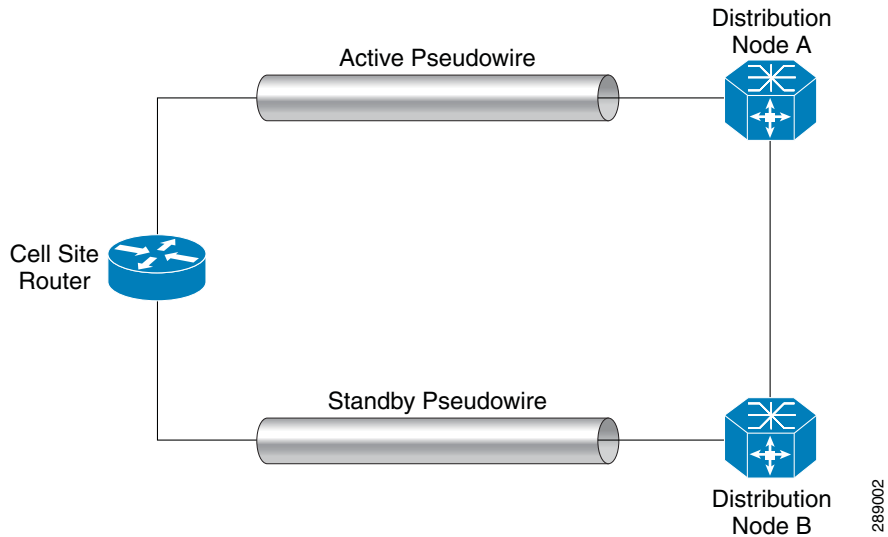


Figure 4-2 is an abstracted topology view that is used in this chapter when discussing how to configure RAN backhaul services in the Prime Provisioning GUI.

Figure 4-2 Abstracted RAN Backhaul Topology



Prime Provisioning is able to provision Internet Protocol based Radio Access Networking (IP RAN) services. This is performed through the Ethernet Virtual Circuit (EVC) policies and service requests within Prime Provisioning to configure the following services to enable RAN backhaul services:

- Circuit Emulation Time Delay Multiple Access (TDM-CEM)
- Pseudowire provisioning of Asynchronous Transfer Mode (ATM)

In addition, the EVC service requests use CEM and pseudowire class objects to bundle common attributes for reuse on every node where the service is provisioned.

The basic workflow for configuring and managing RAN backhaul services in Prime Provisioning, involves the following tasks:

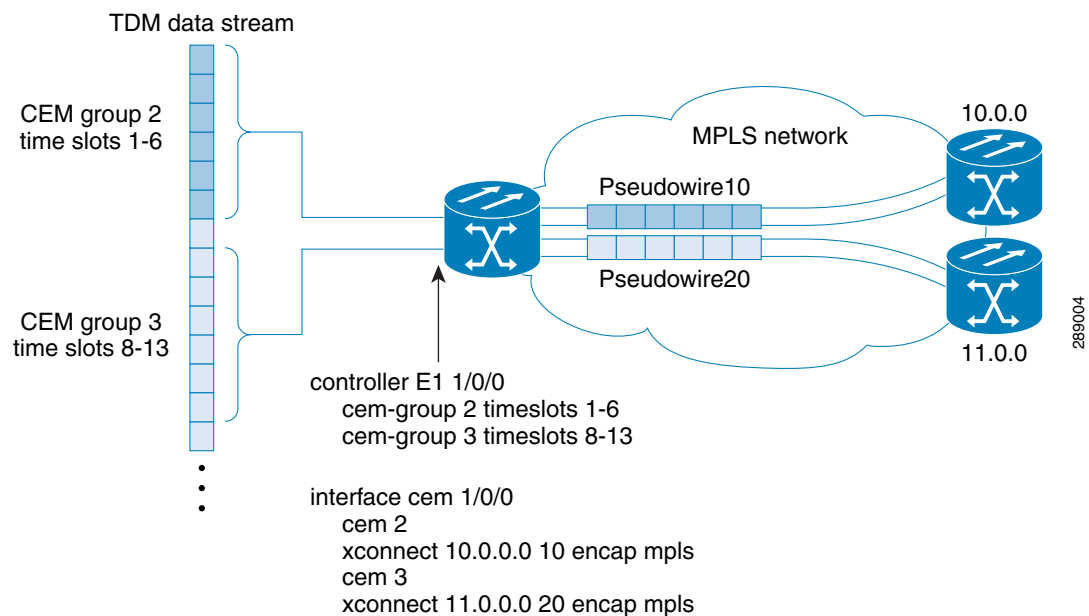
1. Verify prerequisites and preform necessary setup tasks.
2. Create CEM and/or pseudowire classes to be used in RAN backhaul policies and service requests.
3. Create the TDM-CEM or ATM policy.
4. Create template(s) for use in the TDM-CEM or ATM service request.
5. Create the TDM-CEM or ATM service request.
6. Deploy the service request to the device(s) on the network.

In this chapter, the above workflow tasks are documented for the TDM-CEM service.

Overview of the CEM TDM Service

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

Prerequisites

To create TDM-CEM policies and service requests, you must first define the service-related elements in Prime Provisioning, such as target devices and network links. Normally, you create these elements once. For some coverage of these tasks, see [Setting Up the Prime Provisioning Services, page 3-7](#).

Also see other chapters of this guide for how to perform basic infrastructure set up and discovery tasks. The information in the following chapters assumes you have already performed these preliminary tasks.

Managing 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 TDM-CEM 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

TDM-CEM policies and service requests can also use pseudowire classes. Information about creating and managing pseudowire classes is covered in another section of this guide. For more information, see [Creating and Modifying Pseudowire Classes, page 3-15](#).

This section covers the following topics:

- [Creating a CEM Class Object, page 4-4](#)
- [Editing a CEM Class Object, page 4-5](#)
- [Deleting a CEM Class Object, page 4-6](#)
- [Sample Configlets for CEM Classes, page 4-6](#)

Creating a CEM Class Object

Perform the following steps to create a CEM class.

- Step 1** From the top-level menu in the Prime Provisioning 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 appropriate 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.
- **Dummy Mode**—The bit pattern that replaces lost and corrupted frames. To enable this mode, use the command `dummy-mode [last-frame | user-defined]`.
Example: `Router(config-cem)# dummy-mode last-frame`
The Dummy Mode options are:
 - **User Defined:** If you select this mode, it is mandatory to provide a dummy pattern.
 - **Last Frame:** The default Dummy mode. If you select this mode, the Dummy Pattern field is hidden and the dummy pattern value is automatically selected from the service.
 - **None:** If you select this mode, no attributes related to the mode and the pattern are included.
- **Dummy Pattern (hexadecimal pattern)**—If the Dummy Mode is set to User Defined, you can use the Dummy Pattern option to configure the dummy pattern. The range for this pattern is from 0x0 to 0xFF. The default value for the dummy pattern is 0xFF.
Example:
`Router(config-cem)# dummy-pattern 0x55`

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 Provisioning GUI, choose **Inventory > Logical Inventory > CEM class**.

The CEM Class window appears showing any CEM classes already created in Prime Provisioning.

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 TDM-CEM 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 Provisioning GUI, choose **Inventory > Logical Inventory > CEM Class**.

The CEM Class window appears showing any CEM classes already created in Prime Provisioning.

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 TDM-CEM 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 TDM-CEM Policy

This section describes how to create a TDM-CEM policy.

You must define a TDM-CEM 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 TDM-CEM policy whenever you create a new type of service or a service with different parameters.

You can also associate Prime Provisioning templates and data files with a policy. See [Using Templates with Policies, page 11-20](#) 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 D, “Adding Additional Information to Services.”](#)

To start defining a TDM-CEM 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 Provisioning 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 Provisioning 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 TDM-CEM 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** from the drop-down list.

The choices are:

- **SAToP_UNFRAMED**—Structure-agnostic TDM over packet. This mode used to encapsulate T1 or E1 or SERIAL 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 T1 or E1 or SERIAL 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.
- **SERIAL**— SERIAL data circuit-terminating equipment (DCE). Sub-Rate (<64kbps) serial line speeds, only one time slot is used in the TDM frame.



Note Sub-rate (below DS0 bandwidth) Async (R232) data over MPLS MPLS using T1/E1 CESoP is not supported.

Step 3 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 4 Click **Next**.

The Policy Editor window appears.

Step 5 Continue with the steps contained in the next section, [Setting the Service Attributes, page 4-9](#).

Setting the Service Attributes

To set the service attributes for the TDM-CEM 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 B, “Terminating an Access Ring on Two N-PEs”](#) and, specifically, the section [Using N-PE Redundancy in FlexUNI/EVC Service Requests, page B-3](#), for notes on how this option can be used.
- Step 2** Check the **AutoPick VC ID** check box to have Prime Provisioning 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 Provisioning allocates a VC ID for pseudowires from the Prime Provisioning-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 TDM-CEM 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 **PseudoWireClass** commands on IOS and IOS XR devices. See [Creating and Modifying Pseudowire Classes, page 3-15](#) 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 Provisioning.
- Step 2** Select an L2VPN group name for the CEM class from the drop-down list. Your options are:
- ISC
 - VPNSC
 - Other options derived from a configurable DCPL property.
- Usage Notes:
- This attribute is used for provisioning the L2VPN group name on IOS XR devices. 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](#).
 - The L2VPN Group Name attribute is not available if the MPLS core connectivity type is set as VPLS in the window.

- Step 3** To enable Prime Provisioning to generate an E-Line name automatically, check the **Autopick E-Line Name** checkbox. Or, specify the point-to-point (p2p) E-line name for the CEM class manually. This attribute is not available if the MPLS core connectivity type is set to VPLS in the window. If you enable Prime Provisioning to generate the E-Line name automatically, the E-line name generated is of the format **HostNameOfDeviceAtTerminalA_VCID**. In case of the Z-backup nodes, the format used is **HostNameOfDeviceAtTerminalA_BackupVCID**. E-Line Name is only applicable for IOS XR devices.
- Step 4** 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 [Managing 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 Provisioning.
 - Use CEM Class is applicable to IOS and IOS-XR devices.
- Step 5** Click **Next**.
- The Policy Editor window appears.
- Step 6** Continue with the steps contained in the next section, [Adding User-Defined Fields into the TDM-CEM Policy Workflow, page 4-10](#).
-

Adding User-Defined Fields into the TDM-CEM 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 D, “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 Provisioning 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 Provisioning.



Note

Template variable support is available for TDM-CEM services. An example template and data file is available containing the CEM-related variables. See the next section [Using Template Variables in TDM-CEM Services, page 4-11](#), 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 11-20](#).

- 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 TDM-CEM policy, click **Finish**.
-

To create a service request based on an CDM TEM policy, see [Managing TDM-CEM Service Requests, page 4-11](#).

Using Template Variables in TDM-CEM Services

This section describes how to access and use the example CEM template in Prime Provisioning. To create a data file for the example CEM template, perform the following steps.

-
- Step 1** In the Prime Provisioning 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 a new one. In either case, the Data File Editor window appears. You can use this file to map the template variables required for provisioning TDM-CEM 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 TDM-CEM Service Requests

This section describes the various tasks of the workflow for managing TDM-CEM service requests. It contains the following sections:

- [Creating a TDM-CEM Service Request, page 4-12](#)
- [Setting the Service Request Details, page 4-12](#)
- [Selecting Devices, page 4-14](#)

- [Modifying the TDM-CEM Service Request, page 4-18](#)
- [Using Templates and Data Files with a TDM-CEM Service Request, page 4-19](#)
- [Saving the TDM-CEM Service Request, page 4-19](#)

Creating a TDM-CEM Service Request

To begin creating the TDM-CEM 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 TDM-CEM policy from the policies previously created (see [Creating a TDM-CEM 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 Provisioning 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 Provisioning 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 Provisioning allocates a VC ID for pseudowires from the Prime Provisioning-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:

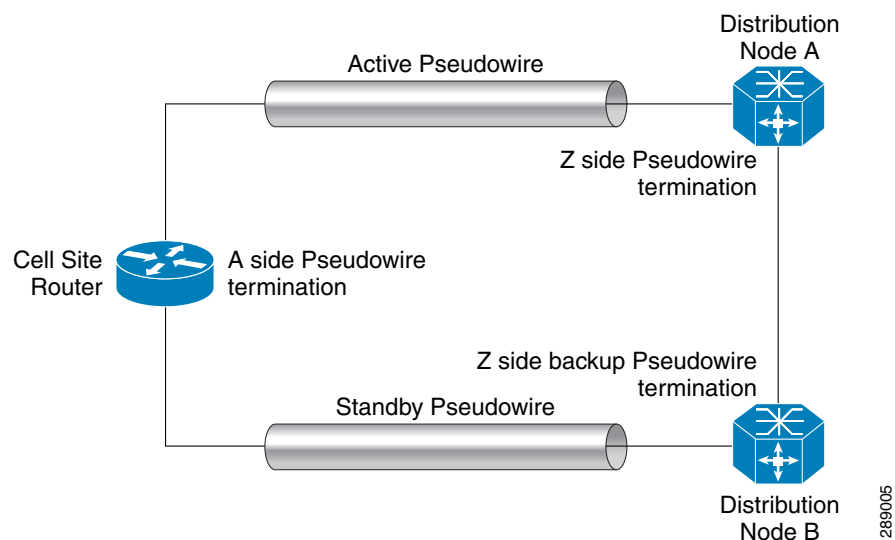
- The VC ID value must be an integer value corresponding to a VC ID.
- When a VC ID is manually allocated, Prime Provisioning verifies the VC ID to see if it lies within Prime Provisioning's VC ID pool. If the VC ID is in the pool but not allocated, the VC ID is allocated to the service request. If the VC ID is in the pool and is already in use, Prime Provisioning prompts you to allocate a different VC ID. If the VC ID lies outside of the Prime Provisioning VC ID pool, Prime Provisioning 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 B, "Terminating an Access Ring on Two N-PEs"](#) and, specifically, the section [Using N-PE Redundancy in FlexUNI/EVC Service Requests, page B-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.
- **SERIAL**— SERIAL data circuit-terminating equipment (DCE). Sub-Rate (<64kbps) serial line speeds, only one time slot is used in the TDM frame.



Note Sub-rate (below DS0 bandwidth) Async (R232) data over MPLS MPLS using T1/E1 CESoP is not supported.

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 [Managing 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 Provisioning.
- 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 Provisioning, devices added for TDM-CEM 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, 760X, ASR901, ASR903, and ME36xx 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 the container type is SERIAL, only SERIAL 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/SERIAL controller attributes.

- Step 5** Set T1/E1/SERIAL controller attributes for the “A” side terminal device.

Attributes for T1/E1controller:

- **Auto-Pick CEM Group ID**—By default, the check box is always checked and the CEM group ID text box is disabled. The auto-generated CEM group ID appears in the CEM group ID text box after deploying the service request. To enter the value manually, uncheck the check box and enter the value in the CEM group ID text box.
- **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.
- **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 attribute in the policy was set to CESoPN_TIMESLOT. It does not appear if the attribute was set to SAToP_UNFRAMED.

Attributes for SERIALcontroller:

- **Channel Group**—The value is always default. The default value is 0.
- **Raw Socket Packet Length**—A value from 2 to 1400.

- **Raw Socket Packet Timer**—A value from 3 to 1000.
- **Number of Data Bits**—The values are 5, 6, 7 and 8.
- **Terminal Parity**—NONE, EVEN, ODD, MARK and SPACE from the drop-down list.
- **Async line stop bits**—The values are 1, 1.5 and 2.
- **Transmit and Receive Speeds**—A value from 0 to 4294967295.
- **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.
- **L2VPN Group Name**— Select an L2VPN group name for the CEM class from the drop-down list. Your options are SC, VPNSC, and other options derived from a configurable DCPL property. This attribute is used for provisioning the L2VPN group name on IOS XR devices. 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. The L2VPN Group Name attribute is not available if the MPLS core connectivity type is set as VPLS in the window.
- **Autopick E-Line Name**— Select this checkbox to enable Prime Provisioning to generate an E-Line name automatically. Or, specify the point-to-point (p2p) E-line name for the CEM class manually. This attribute is not available if the MPLS core connectivity type is set to VPLS in the window. If you enable Prime Provisioning to generate the E-Line name automatically, the E-line name generated is of the format **HostNameOfDeviceAtTerminalA_VCID**. In case of the Z-backup nodes, the format used is **HostNameOfDeviceAtTerminalA_BackupVCID**. E-Line Name is only applicable for IOS XR devices.

- Step 6** After setting the attributes for the T1/E1/SERIAL 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 (IOS)	Channelization Mode (IOS-XR)
E1	SDH	Au-4	C-12	C-12-E1
T1	SDH	Au-3	C-11	C-11-T1
T1	SONET	N/A	STS-1	VT15-T1
SERIAL	SONET	N/A	N/A	N/A

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.

- **Auto-Pick CEM Group ID**—When the Auto-Pick CEM Group ID check box is checked the CEM group ID text box is disabled. The auto-generated CEM group ID appears in the CEM group ID text box after deploying the service request.
- **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.
- **Resource Availability**—Click this link to check the resources that have been configured or provisioned on the same controller that belongs to the same device through a different service in Prime Provisioning. This feature is applicable to EVC-TDM service blades on IOS and XR platforms only. It is not applicable to ATM services.
- **Width**—Width of an E1 controller. The value can range from 1 to 3. By default, the value is 3. This field is hidden when an IOS device is selected.
- **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 from 1 to 3, or 1 to 7, depending on the container type (E1 or T1). 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 from 1 to 3. The user must set the value in the text box. There is no default value.
- **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 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 Provisioning displays a warning. Make any corrections or updates needed (based on the information provided by Prime Provisioning), and click the **Save** button.

For information on modifying an EVC service request see the section [Modifying the TDM-CEM Service Request, page 4-18](#). For additional information about saving an TDM-CEM service request, see [Saving the TDM-CEM Service Request, page 4-19](#).

Modifying the TDM-CEM Service Request

You can modify a TDM-CEM 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 Provisioning.

- 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 a TDM-CEM Service Request, page 4-19](#).
- Step 6** When you are finished editing the TDM-CEM service request, click **Save**.
For additional information about saving an TDM-CEM service request, see [Saving the TDM-CEM Service Request, page 4-19](#).
-

Using Templates and Data Files with a TDM-CEM Service Request

Prime Provisioning 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 Provisioning 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 11-24](#).

Saving the TDM-CEM Service Request

To save an TDM-CEM 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 TDM-CEM 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 TDM-CEM service request, see [Deploying Service Requests, page 10-9](#).

For sample configlets for TDM-CEM services, see the section [Sample Configlets for TDM-CEM Services, page 4-21](#).

Creating an E1-E1, T1-T1 and SERIAL- SERIAL circuit using Prime Provisioning

Prime Provisioning enables you to provision an E1-E1 circuit by selecting E1 controllers at both Terminal A and Terminal Z ends of the PW terminal. To select the E1 controller at Z-backup terminal in the service request, the Enable PseudoWire Redundancy check box must be selected during Service Request creation.

The table below lists the different possible combinations for either E1-E1 or T1-T1 or SERIAL-SERIAL service provisioning.

Table 4-2 *Combinations for E1-E1 or T1-T1 Service Provisioning*

Controller at A	Controller at Z	Controller at Z-Backup
E1 (IOS)	E1 (IOS/IOS-XR)	E1 (if PW redundancy enabled) (IOS/IOS-XR)
T1 (IOS)	T1 (IOS/IOS-XR)	T1 (if PW redundancy enabled) (IOS/IOS-XR)
SERIAL (IOS)	SERIAL (IOS)	SERIAL (if PW redundancy enabled) (IOS)

To view the sample configlets generated between E1 controllers, see:

- [TDM-CEM between E1 controllers \(IOS-XR device\), page 4-29](#)
- [TDM-CEM between T1 controllers \(IOS device\), page 4-30](#)

Sample Configlets for TDM-CEM Services

This section provides sample configlets for RAN backhaul service provisioning in Prime Provisioning. It contains the following subsections:

- [Overview, page 4-21](#)
- [TDM-CEM using SAToP PW3, page 4-22](#)
 - [TDM-CEM using framing type SDH \(IOS-XR device\), page 4-23](#)
 - [TDM-CEM using framing type SONET \(IOS device\), page 4-24](#)
 - [TDM-CEM using framing type SONET \(IOS-XR device\), page 4-25](#)
- [TDM-CEM using CESoPSN, page 4-27](#)
 - [TDM-CEM between E1 controllers \(IOS device\), page 4-28](#)
 - [TDM-CEM between E1 controllers \(IOS-XR device\), page 4-29](#)
 - [TDM-CEM between T1 controllers \(IOS device\), page 4-30](#)
 - [TDM-CEM between T1 controllers \(IOS-XR device\), page 4-31](#)
 - [TDM-CEM between SERIAL controllers \(IOS device\), page 4-32](#)

Overview

The configlets provided in this section show the CLIs generated by Prime Provisioning 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 Provisioning 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.

TDM-CEM using SAToP PW3

Configuration

- Service: RAN Backhaul.
- Feature: This sections contains sample configlets that would be generated for TDM-CEM 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
<pre>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</pre>	<pre>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</pre>

Comments

- None.

TDM-CEM using framing type SDH (IOS-XR device)

Configuration

- Service: RAN Backhaul.
- Feature: This sections contains sample configlets generated for TDM-CEM SAToP PW3 service on a cell site router with an IOS-XR image and two distribution nodes (Z and Z-backup).
- Device configuration:
 - The cell site router could be a device belonging to any of MWR 2941-DC, ASR901, ASR903, 7600X or ME36XX platform running on an IOS image"
 - Controller: E1 0/4
 - Interface(s): CEM 0/4
 - Distribution node A is an ASR9K series device running on IOS-XR image with XR version being: 4.3.0 and above.
 - Contoller: SONET 0/2/0/0
 - Interface(s): CEM0/2/0/0/1/2/2/1:5
 - Distribution node B is an ASR9K series device running on IOS-XR image with XR version being: 4.3.0 and above.
 - Contoller: SONET 0/2/0/0
 - Interface(s): CEM0/2/0/0/1/2/2/1:5

Configlets

Cell Site Router

```

pseudowire-class c76a3-1
  encapsulation mpls
  preferred-path interface Tunnel200
pseudowire-class c76a3-2
  encapsulation mpls
  preferred-path interface Tunnel200
controller E1 0/4
  cem-group 9 timeslots 12-15
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 Z

```

controller SONET0/2/0/0
  au 1
    width 3
    mode tug3
    tug3 2
    mode c12-e1
controller E1 0/2/0/0/1/2/2/1
  cem-group framed 5 timeslots 12
interface CEM0/2/0/0/1/2/2/1:5
  l2transport
l2vpn
  pw-class c76a3-1
  encapsulation mpls
  preferred-path interface tunnel-te
100
  xconnect group ISC
  p2p ems2941q_3090102001
    interface CEM0/2/0/0/1/2/2/1:5
      neighbor 70.70.70.1 pw-id 3090102001
pw-class c76a3-1

```

Node Z-backup (incase of PW Redundancy):

```

controller SONET0/2/0/0
  au 1
    width 3
    mode tug3
    tug3 2
    mode c12-e1
controller E1 0/2/0/0/1/2/2/1
  cem-group framed 5 timeslots 12
interface CEM0/2/0/0/1/2/2/1:5
  l2transport
l2vpn
  pw-class c76a3-2
  encapsulation mpls
  preferred-path interface tunnel-te
100
  xconnect group ISC
  p2p ems2941q_3090403001
    interface CEM0/2/0/0/1/2/2/1:5
      neighbor 70.70.70.1 pw-id 3090403001
pw-class c76a3-2

```

Comments

- None.

TDM-CEM using framing type SONET (IOS device)

Configuration

- Service: RAN Backhaul.
- Feature: This sections contains sample configlets generated for TDM-CEM SAToP PW3 service on a cell site router with an IOS image and two distribution nodes (Z and Z-backup).
- Device configuration:
 - The cell site router could be a device belonging to either of MWR 2941-DC, ASR901, ASR903, 7600X or ME36XX platforms".
 - Controller: T1 0/5
 - Interface(s): CEM 0/0

- Distribution node A is a 760X series device with IOS image.
Contoller: SONET 3/2/0
Interface(s): CEM 3/2/0
- Distribution node B is a 760X series device with IOS image.
Contoller: SONET 3/2/0
Interface(s): CEM 3/2/0

Configlets

Cell Site Router

```
pseudowire-class c76a3-1
  encapsulation mpls
  preferred-path interface Tunnel200
controller T1 0/5
  cem-group 9 timeslots 20
interface CEM0/5
  cem 9
  xconnect 20.10.10.100 2090102001 pw-class c76a3-1
  backup peer 10.0.0.4 2090403001 pw-class c76a3-2
```

Distribution Node Z

```
pseudowire-class c76a3-1
  encapsulation mpls
  preferred-path interface Tunnel200
controller SONET 3/2/0
  sts-1 1
    mode vt-15
    vtg 2 t1 1 cem-group 9 timeslots 12
interface CEM3/2/0
  cem 9
xconnect 100.100.100.2 2090102001 pw-class
c76a3-1
```

Node Z-backup (incase of PW Redundancy):

```
pseudowire-class c76a3-2
  encapsulation mpls
  preferred-path interface Tunnel100
controller SONET 3/2/0
  sts-1 1
    mode vt-15
    vtg 2 t1 1 cem-group 8 timeslots 13
interface CEM3/2/0
  cem 9
  xconnect 100.100.100.2 2090403001
pw-class c76a3-2
```

Comments

- None.

TDM-CEM using framing type SONET (IOS-XR device)

Configuration

- Service: RAN Backhaul.
- Feature: This sections contains sample configlets generated for TDM-CEM SAToP PW3 service on a cell site router with an IOS image and two distribution nodes (Z and Z-backup).
- Device configuration:
 - The cell site router could be a device belonging to either of MWR 2941-DC, ASR901, ASR903, 7600X or ME36XX platforms".
Controller: T1 0/5
Interface(s): CEM 0/0

- Distribution node A is an ASR9K series device running on IOS-XR image with XR version being: 4.3.0 and above.
Contoller: SONET 0/2/0/0
Interface(s): CEM0/2/0/0/1/2/1/1
- Distribution node B is an ASR9K series device running on IOS-XR image with XR version being: 4.3.0 and above.
Contoller: SONET 0/2/0/0
Interface(s): CEM0/2/0/0/1/2/1/1

Configlets**Cell Site Router**

```
pseudowire-class c76a3-1
  encapsulation mpls
  preferred-path interface Tunnel200
controller T1 0/5
  cem-group 9 timeslots 12
interface CEM0/5
  cem 9
  xconnect 20.10.10.100 2090102001 pw-class c76a3-1
  backup peer 10.0.0.4 2090403001 pw-class c76a3-2
```

Distribution Node Z

```
controller SONET0/2/0/0
  sts 2
  mode vt15-t1
controller T1 0/2/0/0/2/4/2
  cem-group framed 12 timeslots 12
interface CEM0/2/0/0/2/4/2:12
  l2transport
l2vpn
  pw-class c76a3-1
  encapsulation mpls
  preferred-path interface tunnel-te
100
  xconnect group ISC
  p2p ems2941g_1525
  interface CEM0/2/0/0/2/4/2
  neighbor 192.18.156.7 pw-id
2090102001
pw-class c76a3-1
```

Node Z-backup (incase of PW Redundancy):

```
controller SONET0/2/0/0
  sts 2
  mode vt15-t1
controller T1 0/2/0/0/2/4/2
  cem-group framed 12 timeslots 12
interface CEM0/2/0/0/2/4/2:12
  l2transport
l2vpn
  pw-class c76a3-1
  encapsulation mpls
  preferred-path interface tunnel-te
200
  xconnect group ISC
  p2p ems2941g_1525
  interface CEM0/2/0/0/2/4/2
  neighbor 192.18.156.7 pw-id
2090403001
pw-class c76a3-2
```

TDM-CEM using CESoPSN

Configuration

- Service: RAN Backhaul.
- Feature: This sections contains sample configlets that would be generated for TDM-CEM 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
<pre>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</pre>	<pre>pseudowire-class c76a3-2 encapsulation mpls preferred-path interface Tunnel1340 ! 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</pre>

Comments

- None.

TDM-CEM between E1 controllers (IOS device)

Configuration

- Service: RAN Backhaul.
- Feature: This sections contains sample configlets that is generated for TDM-CEM CESoPSN service deployed between E1 controllers on an IOS device.
- Device configuration- The cell site router is a router with an IOS image.

Configlets**Cell Site Router**

```
pseudowire-class PW_4
 encapsulation mpls
 preferred-path interface Tunnel200
pseudowire-class PW_7
 encapsulation mpls
 preferred-path interface Tunnel300
controller E1 0/5
 cem-group 9 unframed
interface CEM0/5
 cem 9
  xconnect 20.10.10.100 10234 pw-class PW_4
  backup peer 10.0.0.4 2090403001 pw-class PW_7
```

Distribution Node A	Distribution Node B
<pre>pseudowire-class PW_5 encapsulation mpls preferred-path interface Tunnel200 controller E1 0/5 cem-group 9 unframed interface CEM0/5 cem 9 xconnect 20.10.10.100 10234 pw-class PW_5</pre>	<pre>pseudowire-class PW_6 encapsulation mpls preferred-path interface Tunnel200 controller E1 0/7 cem-group 9 unframed interface CEM0/7 cem 9 xconnect 20.10.10.100 2090403001 pw-class PW_6</pre>

Comments

- None.

TDM-CEM between E1 controllers (IOS-XR device)

Configuration

- Service: RAN Backhaul.
- Feature: This sections contains sample configlets that is generated for TDM-CEM CESoPSN service deployed between E1 controllers on an IOS-XR device.
- Device configuration- The cell site router is a router with an IOS-XR image.

Configlets

Cell Site Router
<pre>pseudowire-class PW_4 encapsulation mpls preferred-path interface Tunnel200 pseudowire-class PW_7 encapsulation mpls preferred-path interface Tunnel300 controller E1 0/5 cem-group 9 unframed interface CEM0/5 cem 9 xconnect 20.10.10.100 10234 pw-class PW_4 backup peer 10.0.0.4 2090403001 pw-class PW_7</pre>

Distribution Node A	Distribution Node B
<pre> controller E1 0/2/0/0/1/2/1/1 cem-group unframed interface CEM0/2/0/0/1/2/1/1 l2transport l2vpn pw-class PW_1H encapsulation mpls preferred-path interface tunnel-te 100 xconnect group ISC p2p ems2941q_10234 interface CEM0/2/0/0/1/2/1/1 neighbor 70.70.70.1 pw-id 10234 pw-class PW_1H </pre>	<pre> controller E1 0/2/0/0/1/2/1/1 cem-group unframed interface CEM0/2/0/0/1/2/1/1 l2transport l2vpn pw-class PW_1H2 encapsulation mpls preferred-path interface tunnel-te 101 xconnect group ISC p2p ems2941q_2090403001 interface CEM0/2/0/0/1/2/1/1 neighbor 70.70.70.1 pw-id 2090403001 pw-class PW_1H2 </pre>

Comments

- None.

TDM-CEM between T1 controllers (IOS device)

Configuration

- Service: RAN Backhaul.
- Feature: This sections contains sample configlets that is generated for TDM-CEM CESoPSN service deployed between T1 controllers.
- Device configuration- The cell site router is a router with an IOS image.

Configlets**Cell Site Router**

```

pseudowire-class PW_4
  encapsulation mpls
  preferred-path interface Tunnel200
pseudowire-class PW_7
  encapsulation mpls
  preferred-path interface Tunnel300
controller T1 0/5
  cem-group 9 unframed
interface CEM0/5
  cem 9
  xconnect 20.10.10.100 10234 pw-class PW_4
    backup peer 10.0.0.4 2090403001 pw-class PW_7

```

Distribution Node A	Distribution Node B
<pre>pseudowire-class PW_5 encapsulation mpls preferred-path interface Tunnel200 controller T1 0/5 cem-group 9 unframed interface CEM0/5 cem 9 xconnect 20.10.10.100 10234 pw-class PW_5</pre>	<pre>pseudowire-class PW_6 encapsulation mpls preferred-path interface Tunnel200 controller T1 0/7 cem-group 9 unframed interface CEM0/7 cem 9 xconnect 20.10.10.100 2090403001 pw-class PW_6</pre>

Comments

- None.

TDM-CEM between T1 controllers (IOS-XR device)

Configuration

- Service: RAN Backhaul.
- Feature: This sections contains sample configlets that is generated for TDM-CEM CESoPSN service deployed between T1 controllers.
- Device configuration- The cell site router is a router with an IOS-XR image.

Configlets

Cell Site Router
<pre>pseudowire-class PW_4 encapsulation mpls preferred-path interface Tunnel200 pseudowire-class PW_7 encapsulation mpls preferred-path interface Tunnel300 controller T1 0/5 cem-group 9 unframed interface CEM0/5 cem 9 xconnect 20.10.10.100 10234 pw-class PW_4 backup peer 10.0.0.4 2090403001 pw-class PW_7</pre>

Distribution Node A	Distribution Node B
<pre> controller T1 0/2/0/0/1/2/1/1 cem-group unframed interface CEM0/2/0/0/1/2/1/1 l2transport l2vpn pw-class PW_1H encapsulation mpls preferred-path interface tunnel-te 100 xconnect group ISC p2p ems2941q_10234 interface CEM0/2/0/0/1/2/1/1 neighbor 70.70.70.1 pw-id 10234 pw-class PW_1H </pre>	<pre> controller T1 0/2/0/0/2/1/1 cem-group unframed interface CEM0/2/0/0/2/1/1 l2transport l2vpn pw-class PW_1H2 encapsulation mpls preferred-path interface tunnel-te 101 xconnect group ISC p2p ems2941q_2090403001 interface CEM0/2/0/0/2/1/1 neighbor 70.70.70.1 pw-id 2090403001 pw-class PW_1H2 </pre>

Comments

- None.

TDM-CEM between SERIAL controllers (IOS device)

Configuration

- Service: RAN Backhaul.
- Feature: This sections contains sample configlets that is generated for TDM-CEM CESoPSN service deployed between SERIAL controllers.
- Device configuration- The cell site router is a router with an IOS image.

Configlets

Terminal - A	Terminal - Z
<pre> pseudowire-class newpw-cls encapsulation mpls controller SERIAL 0/5/0 physical-layer async channel-group 0 interface Serial0/5/0 no ip address encapsulation trans xconnect 192.168.18.123 1098 encapsulation mpls pw-class newpw-cls line 0/5/0 raw-socket packet-length 98 raw-socket packet-timer 78 databits 6 parity odd stopbits 1 speed 900 </pre>	<pre> pseudowire-class newpw-cls encapsulation mpls controller SERIAL 0/3/13 physical-layer async channel-group 0 interface Serial0/3/13 no ip address encapsulation trans xconnect 192.18.156.10 1098 encapsulation mpls pw-class newpw-cls line 0/3/13 raw-socket packet-length 98 raw-socket packet-timer 87 databits 5 parity none stopbits 1 speed 900 </pre>



Managing ATM Services (RAN Backhaul)

This chapter describes how to work with pseudowire classes, create ATM policies and ATM/IMA interfaces using templates, and manage ATM service requests. It contains the following sections:

- [Overview of RAN Backhaul Services, page 5-1](#)
- [Overview of the ATM Service, page 5-3](#)
- [Managing Pseudowire Classes, page 5-4](#)
- [Creating an ATM Policy, page 5-4](#)
- [Using Template Variables in ATM Services, page 5-7](#)
- [Creating an ATM/IMA Interface Using Templates, page 5-7](#)
- [Managing an ATM Service Request, page 5-10](#)
- [Sample Configlets for ATM Services, page 5-17](#)

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 5-1](#) shows an example RAN backhaul topology.

Figure 5-1 Example RAN Backhaul Topology

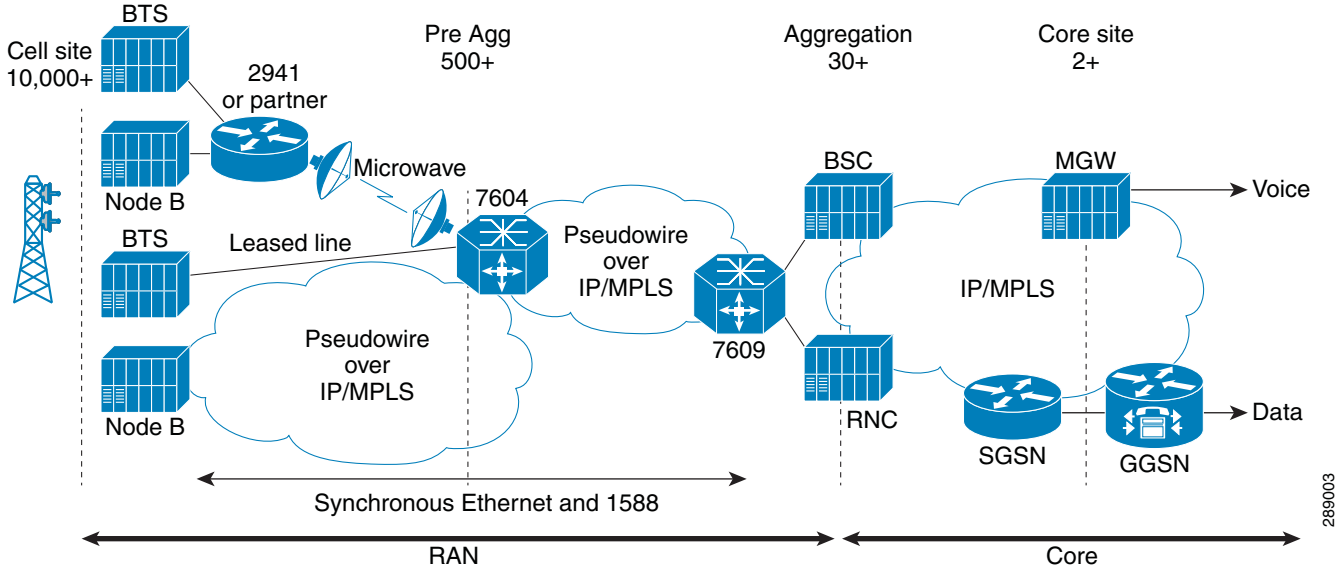
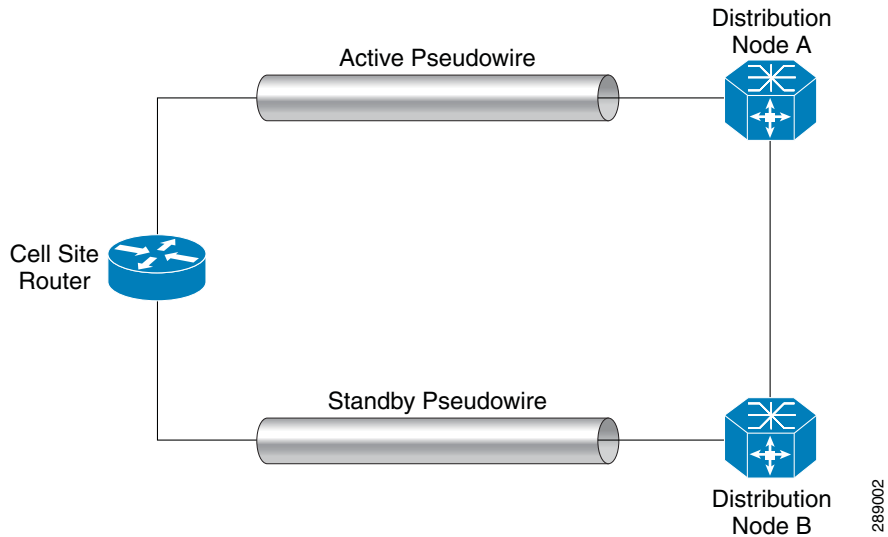


Figure 5-2 is an abstracted topology view that is used in this chapter when discussing how to configure RAN backhaul services in the Prime Provisioning GUI.

Figure 5-2 Abstracted RAN Backhaul Topology



Prime Provisioning is able to provision Internet Protocol based Radio Access Networking (IP RAN) services. This is performed through the Ethernet Virtual Circuit (EVC) policies and service requests within Prime Provisioning to configure the following services to enable RAN backhaul services:

- Circuit Emulation Time Delay Multiple Access (TDM-CEM)
- Pseudowire provisioning of Asynchronous Transfer Mode (ATM)

In addition, the EVC service requests use CEM and pseudowire class objects to bundle common attributes for reuse on every node where the service is provisioned.

The basic workflow for configuring and managing RAN backhaul services in Prime Provisioning, involves the following tasks:

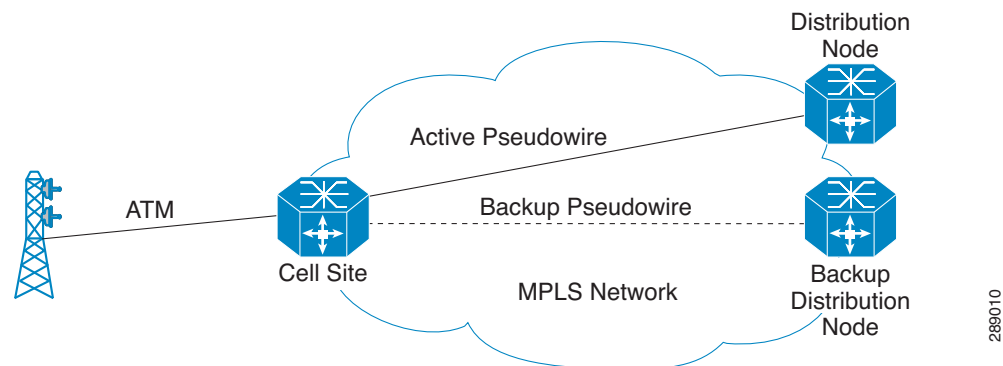
1. Verify prerequisites and preform necessary setup tasks.
2. Create CEM and/or pseudowire classes to be used in RAN backhaul policies and service requests.
3. Create the TDM-CEM or ATM policy.
4. Create template(s) for use in the TDM-CEM or ATM service request.
5. Create the TDM-CEM or ATM service request.
6. Deploy the service request to the device(s) on the network.

In this chapter, the above workflow tasks are documented for the ATM service.

Overview of the ATM Service

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 5-3](#).

Figure 5-3 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.

Prerequisites

To create ATM policies and service requests, you must first define the service-related elements in Prime Provisioning, such as target devices and network links. Normally, you create these elements once. For some coverage of these tasks, see [Setting Up the Prime Provisioning Services, page 3-7](#).

Also see other chapters of this guide for how to perform basic infrastructure set up and discovery tasks. The information in the following chapters assumes you have already performed these preliminary tasks.

Managing 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 3-15](#).

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 Provisioning templates and data files with a policy. See [Using Templates with Policies, page 11-20](#). 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 D, “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 Provisioning 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 Provisioning 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 5-5](#).

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 5-5](#).

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 Provisioning 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 Provisioning allocates a VC ID for pseudowires from the Prime Provisioning-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 5-6](#).
-

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 3-15](#) 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 Provisioning.
- 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 5-6](#).
-

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 D, “Adding Additional Information to Services”](#).

Continue with the steps contained in the next section, [Enabling Template Association, page 5-6](#).

Enabling Template Association

The Prime Provisioning 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 Provisioning.



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 5-7](#), 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 11-20](#).
- 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 5-10](#).

Using Template Variables in ATM Services

This section describes how to access and use the example ATM template in Prime Provisioning.

To create a data file for the example ATM template, perform the following steps.

-
- Step 1** In the Prime Provisioning 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 ATMDData data file in the Data File Name column of the table.
- Step 5** Either click the **Edit** button to edit the ATMDData 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 Provisioning 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 Provisioning. If you require more detailed information on the steps necessary to create templates and data files, see [Chapter 11, “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 Provisioning 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 5-9](#).

**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 Provisioning 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, page 5-10](#)
- [Setting the Service Request Details, page 5-11](#)
- [Selecting Devices, page 5-13](#)
- [Selecting Devices, page 5-13](#)
- [Modifying the ATM Service Request, page 5-15](#)
- [Using Templates and Data Files with an ATM Service Request, page 5-15](#)
- [Saving the ATM Service Request, page 5-16](#)

## 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 5-4](#)). 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 5-11](#).
- 

## 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 Provisioning 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 Provisioning 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 Provisioning allocates a VC ID for pseudowires from the Prime Provisioning-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 Provisioning verifies the VC ID to see if it lies within Prime Provisioning'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 Provisioning

prompts you to allocate a different VC ID. If the VC ID lies outside of the Prime Provisioning VC ID pool, Prime Provisioning does not perform any verification about whether or not the VC ID allocated. The operator must ensure the VC ID is available.

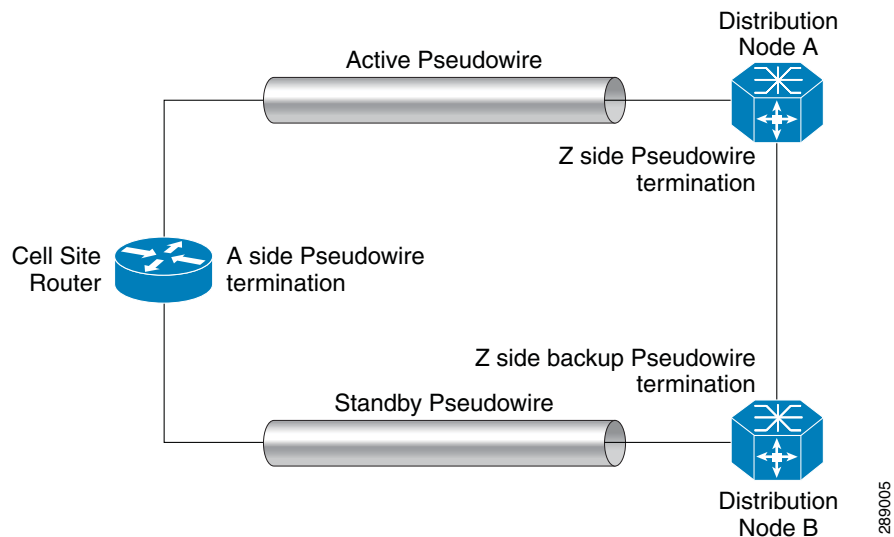
- The VC ID can be entered only while creating a service request. If you are editing the service request, the VC ID field is not editable.

**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 5-4](#) 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 5-4** 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 B, “Terminating an Access Ring on Two N-PEs”](#) and, specifically, the section [Using N-PE Redundancy in FlexUNI/EVC Service Requests, page B-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, [Selecting Devices, page 5-13](#).

## 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 Provisioning, 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 5-4](#) 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.
- 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. A number 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 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 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 Provisioning displays a warning. Make any corrections or updates needed (based on the information provided by Prime Provisioning), and click the **Save** button.

For information on modifying an EVC service request see the section [Modifying the ATM Service Request, page 5-15](#). For additional information about saving an ATM service request, see [Saving the ATM Service Request, page 5-16](#).

---

## 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 Provisioning.
- 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 5-15](#).
- 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 5-16](#).
- 

## Using Templates and Data Files with an ATM Service Request

Prime Provisioning 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 Provisioning 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 11-24](#).

## 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 10-9](#).
- 

For sample configlets for ATM services, see the section [Sample Configlets for ATM Services, page 5-17](#).



# Sample Configlets for ATM Services

This section provides sample configlets for RAN backhaul service provisioning in Prime Provisioning. It contains the following subsections:

- [Overview, page 5-17](#)
- [ATM/IMA PVP Service, page 5-18](#)
- [ATM/IMA VCC Service, page 5-20](#)
- [ATM PVC Service \(IOS-XR device\), page 5-21](#)
- [ATM PVP Service \(IOS-XR device\), page 5-22](#)
- [ATM/PVP Service \(ASR platform, IOS device\), page 5-23](#)
- [ATM/PVP Service \(ASR platform, IOS-XR device\), page 5-24](#)
- [ATM/PVC Service \(ASR platform, IOS device\), page 5-25](#)
- [ATM/PVC Service \(ASR platform, IOS-XR device\), page 5-26](#)

## Overview

The configlets provided in this section show the CLIs generated by Prime Provisioning 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 Provisioning 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.

---

## 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                                                                                                                                                                                                                                                                      |
|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <pre>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</pre> | <pre>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</pre> |

**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 12transport
 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                                                                                                                                                                                                                                                                                                                                                   |
|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <pre>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</pre> | <pre>pseudowire-class c76a3-2  encapsulation mpls  preferred-path interface Tunnel1340 ! 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</pre> |

**Comments**

- None.

## ATM PVC Service (IOS-XR device)

**Configuration**

- Service: RAN Backhaul.
  - Feature: This sections contains sample configlets that would be generated for ATM PVC service on a cell site router with an IOS-XR image.
  - Device configuration:
    - The cell site router could be a device belonging to either MWR 2941-DC, ASR901, ASR903, 7600X, or ME36XX platforms with with an IOS-XR image.
- Interface(s): ATM0

**Configlets****Cell Site Router**

```
interface ATM0/2/1/0
 description Link to ONS 12/2
 atm mctp-timers 1000 2000 4000
!
interface ATM0/2/1/0.21 l2transport
 pvc 2/121
 encapsulation aal0
 cell-packing 17 1
!
!
xconnect group atm21
 p2p atm21
 interface ATM0/2/1/0.21
 neighbor 192.168.0.7 pw-id 21
 pw-class atm
```

---

**Comments**

- None.

## ATM PVP Service (IOS-XR device)

---

**Configuration**

- Service: RAN Backhaul.
- Feature: This sections contains sample configlets that would be generated for ATM PVP service on a cell site router with an IOS-XR image.
- Device configuration:
  - The cell site router could be a device belonging to either MWR 2941-DC, ASR901, ASR903, 7600X, or ME36XX platforms with with an IOS-XR image.

Interface(s): ATM0

---

**Configlets**

---

**Cell Site Router**

---

```
interface ATM0/2/1/0
description ATM MS-PW BU
atm mctp-timers 1000 2000 4000
!
interface ATM0/2/1/0.88 12transport
pvp 200
 cell-packing 200 2
!
!
xconnect group atm88
p2p atm55
 interface ATM0/2/1/0.88
 neighbor 192.168.0.8 pw-id 55
 pw-class atm
!
!
```

---



---

**Comments**

- None.

## ATM/PVP Service (ASR platform, IOS device)

---

### Configuration

- Service: RAN Backhaul.
- Feature: This sections contains sample configlets that would be generated for ATM PVP service on a cell site router and one distribution node.
- Device configuration:
  - The cell site router is an ASR 901 device with an IOS image.  
Interface(s): ATM0/4/IMA0
  - Distribution node A is an ASR 903 device with IOS image.  
Interface(s): ATM 6/0/2

---

### Configlets

---

#### Cell Site Router

```
interface ATM0/4/ima0
 atm pvp 23 l2transport
 cell-packing 12 mcpt-timer 2
 xconnect 20.10.10.100 1503 encapsulation mpls
```

---

---

#### Distribution Node

```
interface ATM6/0/2
 atm pvp 76 l2transport
 no epd
 cell-packing 27 mcpt-timer 3
 xconnect 192.168.0.7 1503 encapsulation
mpls
```

---

---

### Comments

- None.

## ATM/PVP Service (ASR platform, IOS-XR device)

### Configuration

- Service: RAN Backhaul.
- Feature: This sections contains sample configlets that would be generated for ATM PVP service on a cell site router and one distribution node.
- Device configuration:
  - The cell site router is an ASR 901 device with an IOS image.  
Interface(s): ATM0/4/IMA0
  - Distribution node A is an ASR 9K device with IOS-XR image.  
Interface(s): ATM 0/2/1/0.88

### Configlets

#### Cell Site Router

```
interface ATM0/4/ima0
 atm pvp 23 l2transport
 cell-packing 12 mcpt-timer 2
 xconnect 20.10.10.100 1503 encapsulation mpls
```

#### Distribution Node

```
interface ATM0/2/1/0.88 l2transport
 pvp 123
 cell-packing 14 1
l2vpn
 xconnect group ISC
 p2p test_atm_pvp
 interface ATM0/2/1/0.88
 neighbor 70.70.70.1 pw-id 2321
```

### Comments

- None.



## ATM/PVC Service (ASR platform, IOS device)

---

### Configuration

- Service: RAN Backhaul.
- Feature: This sections contains sample configlets that would be generated for ATM PVP service on a cell site router and one distribution node.
- Device configuration:
  - The cell site router is an ASR 901 device with an IOS image.  
Interface(s): ATM0/2.45
  - Distribution node A is an ASR 903 device with IOS image.  
Interface(s): ATM 6/0/3.45

---

### Configlets

---

#### Cell Site Router

---

```
interface ATM6/0/2.45 point-to-point
 pvc 5/66 12transport
 encapsulation aal0
 cell-packing 4 mcpt-timer 1
 xconnect 20.10.10.100 encapsulation mpls
```

---

---

#### Distribution Node

---

```
interface ATM6/0/3.45 point-to-point
 pvc 5/66 12transport
 encapsulation aal0
 cell-packing 4 mcpt-timer 1
 xconnect 45.2.2.2 756 encapsulation
mpls
```

---

---

### Comments

- None.

## ATM/PVC Service (ASR platform, IOS-XR device)

### Configuration

- Service: RAN Backhaul.
- Feature: This sections contains sample configlets that would be generated for ATM PVP service on a cell site router and one distribution node.
- Device configuration:
  - The cell site router is an ASR 901 device with an IOS image.  
Interface(s): ATM6/0/2.45
  - Distribution node A is an ASR 903 device with IOS-XR image.  
Interface(s): ATM 0/2/1/0.7644

### Configlets

#### Cell Site Router

```
interface ATM6/0/2.45 point-to-point
 pvc 5/66 l2transport
 encapsulation aa10
 cell-packing 4 mcpt-timer 1
 xconnect 20.10.10.100 encapsulation mpls
```

#### Distribution Node

```
interface ATM0/2/1/0.7644 l2transport
 pvc 150/3111
 encapsulation aa10
 cell-packing 27 1
l2vpn
 pw-class PW2026
 encapsulation mpls
 protocol ldp
 xconnect group ISC
 p2p ems2941q_442344
 interface ATM0/2/1/0.7644
 neighbor 70.70.70.1 pw-id 442344
 pw-class PW2026
```

### Comments

- None.



## Managing MPLS VPN Services

This chapter describes the tasks required to get started using Cisco Prime Provisioning 7.2, 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 Provisioning services, see [Setting Up the Prime Provisioning Services, page 6-4](#).



**Note**

You can create a service by picking endpoints on a map in Prime Network Vision, when Prime Provisioning and Prime Network are integrated with Prime Central. For MPLS VPNs, only the "no CE" option (no CE device present) is supported by Prime Provisioning.

- 1) On any map select one or more endpoint devices using <Ctrl> Click.
- 2) In the right click menu select Fulfill/Create Service. The same first screen that you see when you create a service in Prime Provisioning, is displayed.
- 3) Pick a policy. Depending on the number of endpoints selected not all policies will work. For example, if you have five endpoints selected, you cannot create a Point to Point service, but you can still create a VPLS or a L3 VPN.
- 4) Once you select the policy, the Service Request page appears with links and with the selected devices prepopulated.

This chapter covers the following topics:

- [Getting Started with MPLS VPN, page 6-2](#)
- [Setting Up the Prime Provisioning Services, page 6-4](#)
- [Independent VRF Management, page 6-14](#)
- [IPv6 and 6VPE Support in MPLS VPN, page 6-31](#)
- [MPLS VPN Service Policies, page 6-40](#)
- [Customizing EVC and MPLS Policies, page 6-82](#)
- [Provisioning Regular PE-CE Links, page 6-109](#)
- [Provisioning Multi-VRFCE PE-CE Links, page 6-121](#)
- [Provisioning Management VPN, page 6-132](#)
- [Provisioning Cable Services, page 6-141](#)
- [Provisioning Carrier Supporting Carrier, page 6-150](#)
- [Provisioning Multiple Devices, page 6-155](#)

- [Spanning Multiple Autonomous Systems, page 6-165](#)
- [Sample Configlets, page 6-177](#)
- [Troubleshooting MPLS VPNs, page 6-214](#)
- [VRFs, page 6-223](#)

## Getting Started with MPLS VPN

This section covers the following topics:

- [Before You Begin, page 6-2](#)
- [Prime Provisioning Service Activation, page 6-2](#)
- [Working with MPLS Policies and Service Requests, page 6-3](#)

### Before You Begin

Before you can use MPLS VPN to provision, perform the following steps:

---

**Step 1** Install Prime Provisioning. See the [Cisco Prime Provisioning 7.2 Installation Guide](#).

**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 Provisioning provisions only PE-CEs, not devices within a given network.

**Step 4** Populate Prime Provisioning.

---

### Prime Provisioning Service Activation

To activate MPLS services you must configure Prime Provisioning so it “knows” about the preconfiguration information, such as devices, providers, customers, and so on, that Prime Provisioning is going to manage and their roles. The major steps to achieve Prime Provisioning 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 Provisioning Services, page 6-4](#)

## Working with MPLS Policies and Service Requests

After you have set up providers, customers, devices, and resources in Prime Provisioning, 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 6-40](#) as well as subsequent chapters in this guide.
- Step 3** Customize the MPLS policy by embedding command line interface (CLI) templates into the MPLS policy. You can also extend policies by adding attributes that you define directly in the policy screen. For more information, see [Customizing EVC and MPLS Policies, page 6-82](#).
- Step 4** Provision the MPLS service request.
- See the appropriate section, depending on the type service request you want to provision:
- [MPLS VPN Service Requests, page 6-82](#)
  - [Provisioning Regular PE-CE Links, page 6-109](#)
  - [Provisioning Multi-VRFCPE PE-CE Links, page 6-121](#)
  - [Provisioning Management VPN, page 6-132](#)
  - [Provisioning Cable Services, page 6-141](#)
  - [Provisioning Carrier Supporting Carrier, page 6-150](#)
  - [Provisioning Multiple Devices, page 6-155](#)
  - [Spanning Multiple Autonomous Systems, page 6-165](#)
- Step 5** Deploy the MPLS service request.
- See [MPLS VPN Service Requests, page 6-82](#)
- Step 6** 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 10-10](#).
  - Audit service requests. See the section [Deploying, Monitoring, and Auditing Service Requests, page 3-52](#).
  - Run MPLS reports. See [Reports, page E-88](#).
- Step 7** Troubleshoot MPLS services.
- See [Troubleshooting MPLS VPNs, page 6-214](#)
-

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 6-31](#).
- For sample configlets generated by Prime Provisioning for MPLS services, see [Sample Configlets, page 6-177](#)
- For information about using templates and data files in Prime Provisioning policies and service requests, see [Chapter 11, “Managing Templates and Data Files.”](#)

## Setting Up the Prime Provisioning Services

This section contains the basic steps to set up the Prime Provisioning services to support MPLS VPN service policies and service requests.



### Note

This section presents high-level information on Prime Provisioning services that are relevant to MPLS VPN. For more detailed information on setting up these and other basic Prime Provisioning services, see the [Chapter 2, “Before Setting Up Prime Provisioning”](#) and [Chapter 10, “Managing Service Requests”](#).

This section covers the following topics:

- [Overview, page 6-4](#)
- [Setting Up Devices for IOS XR Support, page 6-6](#)
- [Migrating PE Devices from IOS to IOS XR, page 6-6](#)
- [Defining VPNs, page 6-6](#)
- [Provisioning MPLS Service Requests Using Unique Route Distinguisher, page 6-12](#)

## Overview

To create an MPLS VPN service request, you must create the following infrastructure data:

- **Devices**

A Device in Prime Provisioning is a logical representation of a physical device in the network. You can import devices (configurations) into Prime Provisioning by using Inventory Manager or the Prime Provisioning 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 Provisioning”](#).
- **Import or add raw devices**

Every network element that Prime Provisioning manages must be defined as a device in the Prime Provisioning repository. An element is any device from which Prime Provisioning can collect information. In most cases, devices are Cisco IOS routers and switches. It is recommended that you discover and import devices via Prime Network. However, you can also set up devices in Prime Provisioning manually or by importing device configuration files.
- **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 Provisioning.

- Sites

A Site is a logical component of Prime Provisioning 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 Provisioning. You can create CPE in Prime Provisioning by associating a device with a Customer Site.

For detailed steps to create customers and sites, see [Setting Up Resources, page 2-39 of Chapter 2, “Before Setting Up Prime Provisioning”](#).

- 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 Provisioning.

- Regions

A Region is a logical component of Prime Provisioning 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 Provisioning. You can create PE in Prime Provisioning by associating a Device with a Provider Region. In Prime Provisioning, 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 2-39 of Chapter 2, “Before Setting Up Prime Provisioning”](#).

- 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 Provisioning as PE-CLE.

To create a provider and a region, see [Setting Up Resources, page 2-39 of Chapter 2, “Before Setting Up Prime Provisioning”](#).

- Resource Pools

- IP Addresses

- Multicast

- Route Distinguisher

- Route Target

- VLANs (for Layer 2 Access)

- BVI

To create a provider and a region, see [Setting Up Resources, page 2-39 of Chapter 2, “Before Setting Up Prime Provisioning”](#).

- VPN

Before creating a Service Policy, a VPN name must be defined within Prime Provisioning.

- Route Target(s)

To create a route target, see [Setting Up Resources, page 2-39 of Chapter 2, “Before Setting Up Prime Provisioning”](#).

## Setting Up Devices for IOS XR Support

Prime Provisioning 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 [Cisco Prime Provisioning 7.2 Release Notes](#).

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 Provisioning to retrieve the committed CLI configuration after an XML configuration has been downloaded. See [Viewing Configlets on IOS XR Devices, page 10-5](#) for more information.
- Step 3** Create the device in Prime Provisioning as an IOS XR device, as follows:
- Create the Cisco device by choosing **Inventory > Physical Inventory > Devices > Create > Cisco Device**.
- The Create Cisco Router window appears.
- 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 G, "Property Settings"](#) or see the [Cisco Prime Provisioning 7.2 Administration Guide](#).

- 
- 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 6-177](#).

## 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 6-106](#).

## Defining VPNs

During service deployment, Prime Provisioning 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 Provisioning.



**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 6-14](#)

This section describes how to define MPLS VPNs and IP Multicast VPNs. It contains the following sections:

- [Creating an MPLS VPN, page 6-7](#)
- [Creating an IP Multicast VPN, page 6-9](#)
- [Enabling a Unique Route Distinguisher for a VPN, page 6-11](#)

## 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 Provisioning, 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** to 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 6-11](#)
  - Step 7** Enter the OSPF domain ID 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 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 Provisioning 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 Provisioning 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 6-60](#).

**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 6-9](#).



**Note**

These attributes are not supported for use with MVRFC 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 6-9](#), 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 Provisioning.



**Note**

You must specify a CERC if multicast is enabled.

- From the CE Routing Communities pane, click **Select**.

The Select CE Routing Communities dialog box appears.

- 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** **Import RT List:** Enter one or more Route Targets (RTs) to be imported in the VPN.

For multiple RTs, use a comma (,) separated list. An example RT list is 100:120,100:130,100:140.

**Step 11** **Export RT List:** Enter one or more Route Targets (RTs) to be exported from the VPN.

For multiple RTs, use a comma (,) separated list.

**Step 12** Check the Enable VPLS check box to enable VPLS.

**Step 13** Choose the VPLS service type from the Service Type drop-down menu: **ERS** (Ethernet Relay Service) or **EWS** (Ethernet Wire Service).

**Step 14** 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 15** 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 2-45](#), 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 6-36](#).

Multicast VRF deployments are supported also. For more information about VRF object support in Prime Provisioning, see [Independent VRF Management, page 6-14](#)

To create an IP Multicast VPN, follow the procedure described in [Creating an MPLS VPN, page 6-7](#) 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 Provisioning 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 Provisioning 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 Prime Provisioning 6.8, 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 Provisioning 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:

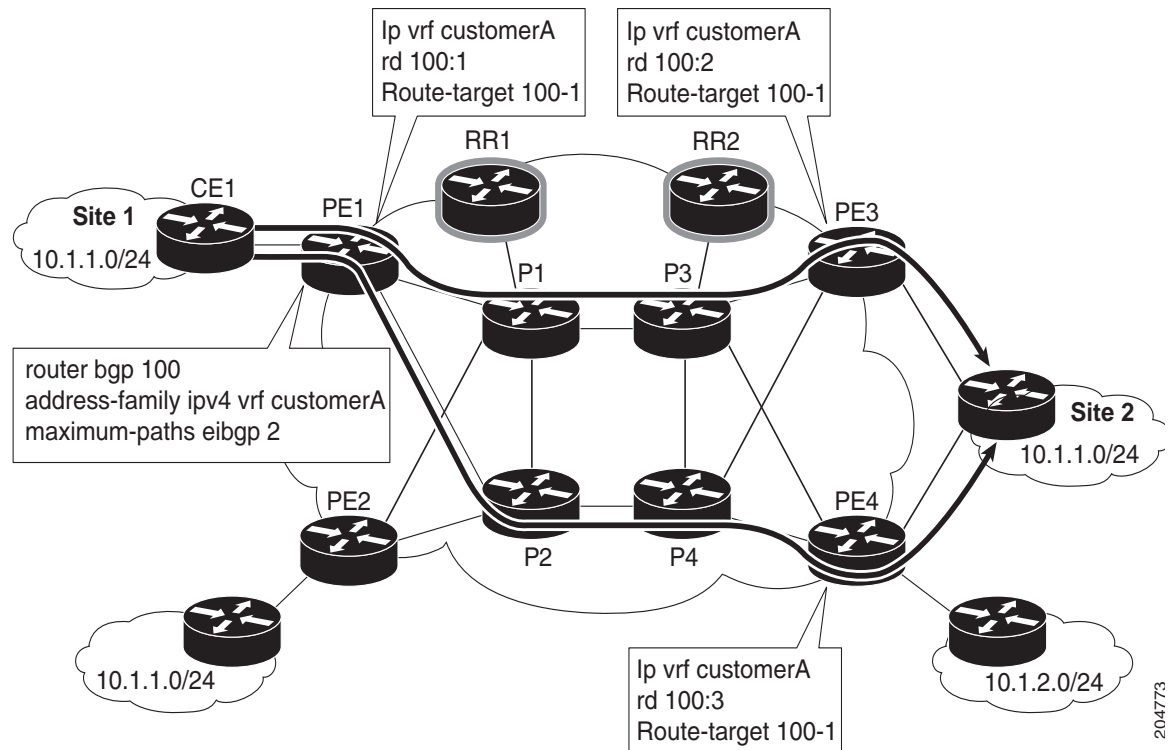
- 
- Step 1** When creating a VPN, check the **Enable Unique Route Distinguisher** check box.
- Step 2** When subsequently creating a service policy and/or service request, select the VPN in the VRF and VPN Membership window.
- The Unique Route Distinguisher **field** appears.
- Step 3** If the unique 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 6-73](#).

## 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 6-1](#) illustrates a load balancing example.


**Figure 6-1** 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 Provisioning supports BGP multipath load sharing through fields and options in the Prime Provisioning 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 6-11](#).
- 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 6-77](#).
- 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 6-1](#).

The outcomes for various cases are described in the Results column of the table.

Table 6-1 Unique RD Use Cases

| SR # | PE  | Unique RD | VRF:RD | Results                                                                                                                                                                                    |
|------|-----|-----------|--------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 1    | pe1 | False     | V24:33 | Prime Provisioning uses the default <i>vrfName:RD</i> , because this is the first time this PE has been configured with this <i>vrfName:RD</i> name.                                       |
| 2    | pe2 | False     | V24:33 | Prime Provisioning uses the default <i>vrfName:RD</i> .                                                                                                                                    |
| 3    | pe3 | True      | V25:34 | Prime Provisioning creates a new <i>vrfName:RD</i> , because Unique RD is true, and it is on a different PE. This PE (pe3) did not have this <i>vrfName:RD</i> configured.                 |
| 4    | pe3 | True      | V25:34 | Prime Provisioning uses the <i>vrfName:RD</i> from SR #3, because the new RD is already present on the PE router.                                                                          |
| 5    | pe2 | True      | V26:35 | Prime Provisioning creates a new <i>vrfName:RD</i> , 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 Provisioning creates a new <i>vrfName:RD</i> , 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. |
| 7    | pe1 | False     | V24:33 | Prime Provisioning uses the default <i>vrfName:RD</i> , as in SR #1.                                                                                                                       |
| 8    | pe3 | False     | V24:33 | Prime Provisioning uses the default <i>vrfName:RD</i> , as in SR #1.                                                                                                                       |
| 9    | pe3 | True      | V25:34 | Prime Provisioning uses the newly created <i>vrfName:RD</i> in SR #4, because it already created a new <i>vrfName:RD</i> for this PE.                                                      |
| 10   | pe2 | True      | V26:35 | Prime Provisioning uses the newly created <i>vrfName:RD</i> in SR #5, because it already create a new <i>vrfName:RD</i> for this PE.                                                       |
| 11   | pe1 | True      | V27:36 | Prime Provisioning uses the newly create <i>vrfName:RD</i> in SR #6, because it already create a new <i>vrfName:RD</i> 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 Provisioning, 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 Provisioning 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 Provisioning 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, page 6-15](#)
- [Working with VRF Objects, page 6-16](#)
- [Working with VRF Service Requests, page 6-22](#)
- [Using VRFs with MPLS VPN Service Requests and Policies, page 6-27](#)
- [Migrating Existing MPLS VPN Service Requests to the VRF Object Model, page 6-30](#)

## Multicast Support for IPv6 on IOS XR Devices

For IOS XR PE devices running release 3.7.0 or later, Prime Provisioning 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 VRFs 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 Provisioning 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 Provisioning will validate this name while provisioning the VRF. If the route map is not defined, Prime Provisioning will generate an error.

**Step 10 Export Route Map:** Enter the name of a route map defined on the device.

Prime Provisioning will validate this name while provisioning the VRF. If the route map is not defined, Prime Provisioning 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 Provisioning 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 Provisioning 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 Provisioning allows the modification of an existing VRF object with the new RD format only if the VRF object is not deployed.
- The following Prime Provisioning 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 Provisioning 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 Provisioning 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 6-60](#).

**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 6-9](#).



**Note** This attribute is not supported for use with MVRFCPE 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 Provisioning 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 VRFs window appears.



**Note** The example assumes that a VRF object has already been created. See [Creating a New VRF Object, page 6-16](#) 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 Provisioning Repository

All VRF objects are stored in the Prime Provisioning repository. You can display these by accessing the VRF Management window at **Inventory > Logical Inventory > VRF** in the Prime Provisioning 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 6-21](#).

### 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 Provisioning 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 Provisioning 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 6-25), 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:
- Click **Save** to save the VRF object and move all of the affected VRF service requests to the **Requested** state.
  - 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.
  - Click **Cancel** to cancel the operation and return to the Edit VRFs window.
- 

## Deleting VRF Objects

To delete VRF objects from the Prime Provisioning 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 Provisioning 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 delete 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 Provisioning service request state transitions, as described in the [Service Enhancements, page 6-82](#).

## 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 VRFs 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 Provisioning 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 Provisioning repository, see [Creating a New VRF Object, page 6-16](#).

**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 Provisioning 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 Provisioning window for selecting a data file and specifying operations such as append and prepend. For information on working with templates in Prime Provisioning, see [Chapter 11, “Managing Templates and Data Files.”](#) For specific information about using the Add/Remove Templates window, see [Using Templates with Service Requests, page 11-24.](#)

**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 Provisioning 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, delete, 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 6-25](#).
- 

## 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 6-106](#) and [Monitoring Service Requests, page 10-10](#).

## 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 Request Manager 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 Provisioning 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 Provisioning 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 Manager 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 Provisioning repository by VRF object name, perform the following steps:

- 
- Step 1** Choose **Operate > Service Requests > Service Request Manager** to access the Service Requests Manager 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.
- 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.



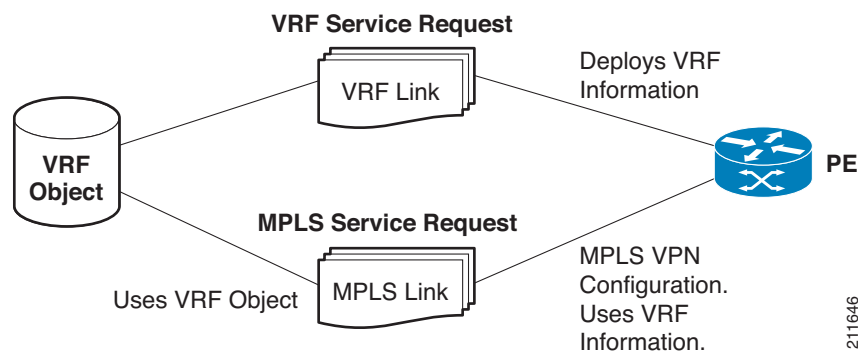
Note

Independent VRF association is not supported for MVRFCPE-based policies and service requests.

## Relationship of VRF Object and Service Requests and PE Device

Figure 6-2 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 6-2 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 Prime Provisioning) 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 Provisioning 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 6-29](#).

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 Provisioning, see [Notes On Using a VRF Object in an MPLS Service Request, page 6-29](#), 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 6-77](#).
-  **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 11, “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 Provisioning 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 MPLS 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 Provisioning 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 Provisioning 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 used 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 Provisioning repository by VRF object name, perform the following steps:

- 
- Step 1** Choose **Operate > Service Requests > Service Request Manager** to access the Service Requests Manager 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 6-27](#). 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 Provisioning 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 6-73](#), for more information.

---

## Migrating Existing MPLS VPN Service Requests to the VRF Object Model

Prime Provisioning 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 Provisioning 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 Provisioning GUI, see the appropriate sections of this guide, as indicated by the references provided.

## Overview of IPv6 and 6VPE

The Prime Provisioning 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 Provisioning Layer 3 VPN services.



Note

For the most current information about IOS and IOS XR versions and hardware platforms supporting IPv6, see [Cisco Prime Provisioning 7.2 Release Notes](#).

This section provides an overview of IPv6 and 6VPE technologies. For an overview of how Prime Provisioning supports IPv6, see [MPLS VPN Support for IPv6 and 6VPE, page 6-33](#).

## 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  $3.4 \times 10^{38}$  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.

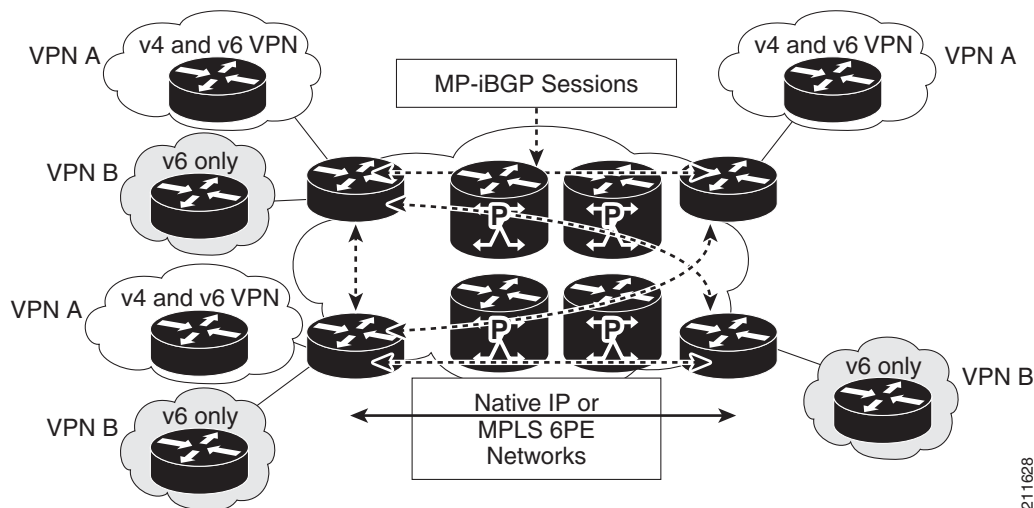
MPLS VPN in Prime Provisioning uses 6VPE to manage Layer 3 VPN services for deployment of IPv6 over a MPLS backbone.

## 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 6-3.

Figure 6-3 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 Provisioning Services, page 6-4](#) for information setting up Prime Provisioning services mentioned in this section.

## IOS and IOS XR Support for IPv6

IPv6 services are available in Prime Provisioning 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 [Cisco Prime Provisioning 7.2 Release Notes](#).

---

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 Provisioning so it “knows” about the preconfiguration information, such as devices, providers, customers, and so on, that Prime Provisioning is going to manage. Prime Provisioning features that support inventory and device management for IPv6 and 6VPE include:

Discovery:

- Prime Provisioning Inventory Manager supports bulk-import of 6VPE devices into the Prime Provisioning 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 Provisioning 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 Provisioning 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 Provisioning 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 Provisioning 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 6-36](#)
- [Multicast Support for IPv6 \(IOS XR Only\), page 6-37](#)

## Independent VRF Object Support

Prime Provisioning 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 Provisioning 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 6-14](#)

## Resource Pools

Prime Provisioning 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 Provisioning MPLS VPN management application supports the provisioning of IPv6 Layer 3 VPNs on an IPv6 Provider Edge router (6VPE). Prime Provisioning 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 Provisioning.

## 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 Provisioning 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 6-36](#).
- 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 6-40](#) 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 to the 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 Provisioning 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 6-82](#) 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 10-3](#), 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 [Cisco Prime Provisioning 7.2 Release Notes](#).

This section describes how Prime Provisioning 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 Provisioning 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 Provisioning 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 Provisioning 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 [Reports, page E-88](#) 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 Provisioning 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 (MVRFC) 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 Provisioning GUI to define MPLS VPN Service Policies. You can also associate Prime Provisioning templates and data files with a policy. See [Chapter 11, “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 D, “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 Provisioning, 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 Provisioning

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 Provisioning 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 Provisioning 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 Provisioning 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 Provisioning asks the service operator, during service activation, only for the PE-CLE or the PE-POP router and customer-facing interface.

**Step 7** Check the **Allow Duplicate IP address** checkbox, if you want this checkbox to appear in MPLS Service Request Editor page.

**Note**

Alternatively, you can set the **AllowDuplicateLinkIPAddress** DCPL property to true for this checkbox to appear in the MPLS Service Request Editor page for all MPLS Service Requests. You can set this DCPL property from the Host Configuration section by choosing repository-> IPAddressPool -> AllowDuplicateLinkIPAddress.

**Step 8** Click **Next**.

To continue with the example, see the following section, [Specifying PE and CE Interface Parameters](#), page 6-43.

## Specifying PE and CE Interface Parameters

To specify the PE, UNI Security, and CE interface information for this MPLS policy follow these steps:

### PE Information

**Step 1** **Interface Type:** From the drop-down list, choose the interface type for the PE. If you select **Any**, the operator creating a service using this policy will be able to select any type of interface. If instead you select a particular interface type, the operator will be restricted to the selected type of interface.

Prime Provisioning 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 6-44.)
- 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 6-2 shows the protocol encapsulations available for each of the supported interface types.

**Table 6-2 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).<br><br><b>Frame-Relay-ietf</b> 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                                                                                                                                                                                                                                                                                                                                       |
| Gigabit 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)                                                                                                                                                                                                                                                                                                              |

Table 6-2 Interface Types and Their Corresponding Encapsulations (continued)

| Interface Type          | Encapsulations (continued)                                                                                                                                                                                                                                                                                                                              |
|-------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Port-Channel            | Default frame, ISL (Inter-Switch Link), dot1q (802.1Q)<br>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)<br><br><br><b>Note</b> In this release, <b>GRE</b> is supported for <b>IOS-XR</b> devices. Earlier while deploying SR on IOS-XR devices, Prime Provisioning was generating invalid configlets and leading to SR deployment failed. |
| VLAN                    | None                                                                                                                                                                                                                                                                                                                                                    |



**Note** MLFR interfaces are supported on IOS and IOS XR devices. Prime Provisioning does not set up the MLFR interface. Prime Provisioning provisions the Layer 3 service on the MLFR interface.

**Step 6 Auto-Pick VLAN ID:** Check this check box to have Prime Provisioning 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 Virtual Interface:** Check this check box to have Prime Provisioning terminate the VRF on a virtual interface. This check box is hidden when you check the **Create virtual interface only** check box. The type of virtual interface created will be chosen appropriately for the device. For example, the 7600 series have a Switched virtual interface (SVI), while the ASR9000 series have a Bridged virtual interface (BVI)

**Step 8 Create virtual interface only:** This option exists if you want to create a layer 2 access service over an MPLS network such as a pseudowire or VPLS, to connect to the L3 VPN. In that case the L3 VPN is not associated with any physical interface, but only the bridge domain from the layer 2 service.

When you check this check box the option to select any physical interface is disabled so that you can directly continue to configuring the Link Attributes. Additionally, the 'Use Virtual interface' option is hidden.

**Step 9 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 6-159.

**Step 10 Standard UNI Port:** Check this check box to access UNI Security Parameters:

#### UNI Security Information

**Step 11 Disable CDP:** Check this check box to disable CDP.

**Step 12 Filter BPDU:** Check this check box to filter BPDU.

- Step 13 Use existing ACL Name:** Check this check box to use existing ACL name.
- Step 14 UNI MAC Addresses:** Click **Edit** to modify or create a MAC address record.
- Step 15 UNI Port Security:** Check this check box to access UNI Port Security parameters:
- Maximum MAC Address:** Enter a valid value.
  - Aging (in minutes):** Enter a valid value.
  - Violation Action:** From the drop-down list, choose one of the following:
    - PROTECT
    - RESTRICT
    - SHUTDOWN
  - Secure MAC Address:** Click **Edit** to modify or create a secure MAC address record.

#### CE Interface Information

- Step 16 Interface Type:** From the drop-down list, choose the interface type for the CE.
- Step 17 Interface Format:** Optionally, you can specify the slot number and port number for the CE interface.
- Step 18 Interface Description:** Optionally, you can enter a description of the CE interface.
- Step 19 Encapsulation:** Choose the encapsulation used for the specified CE interface type.
- Step 20** When satisfied with the interface settings, click **Next**.

To continue with the example, see the following section, [Specifying the IP Address Scheme, page 6-46](#).

## 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 Provisioning: 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 Provisioning uses those addresses (and does not allocate them from the address pool). If the IP addresses are not present in the configuration file, Prime Provisioning picks IPv4 addresses from /30 and /31 subnets 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 Provisioning: MPLS creates a static route for the PE-CE link.



When you choose **IPv4 Unnumbered**, Prime Provisioning: 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 6-48](#).

- **IPv6 Numbered**

This addressing scheme is provided to support a 6VPE router. See [IPv6 and 6VPE Support in MPLS VPN, page 6-31](#) 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 6-31](#) 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 Provisioning software provides the option to specify that 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 Provisioning 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 Provisioning 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 Provisioning uses those addresses (and does not allocate them from the address pool). If the IP addresses are not present in the configuration file, Prime Provisioning picks IP addresses from /30 and /31 subnets point-to-point IP address pool.




---

**Note** This option is not supported for the **IPv6 Numbered** and **IPv4+IPv6 Numbered** address schemes.

---

#### Automatically Assign IPv6 Address

If you choose **IPv6 Numbered** or **IPv4+IPv6 Numbered** and also check the **Automatically Assign IPv6 Address** check box, you will have options to select **IPv6 Pool Mask Type** from /127, /126 or /64 pools. Based on the selection Prime Provisioning picks IPv6 Addresses from the /127, /126 or /64 IPv6 Address Pools.




---

**Note** This option is supported only 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 Provisioning automatically allocate IP addresses from the IP address pool attached to the Region, VPN, or Customer. Prior to defining this aspect of the service policy, the Region, VPN, and Customer must be defined and the appropriate IP address pools assigned to the Region, VPN, and Customer.

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.

---

#### IPv6 Address Pool

The IPv6 Address Pool option gives the service operator the ability to have Prime Provisioning automatically allocate IPv6 addresses from the IPv6 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.




---

**Note** This option is supported only for the IPv6 Numbered and IPv4+IPv6 Numbered address scheme.

---

**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 Provisioning associates the first loopback number with the same VRF name to the interface. If no loopback number already exists, Prime Provisioning creates one.

If a service provider wants Prime Provisioning 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 Provisioning selects loopback interface numbers by sequence. Prime Provisioning 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 Provisioning 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 Provisioning, 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 6-49.

## 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 Provisioning 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 6-31 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 6-51).
- RIP—Routing Information Protocol (see [RIP Protocol Chosen](#), page 6-52).
- BGP—Border Gateway Protocol (see [BGP Protocol Chosen](#), page 6-55).
- OSPF—Open Shortest Path First (see [OSPF Protocol Chosen](#), page 6-60).
- EIGRP—Enhanced Interior Gateway Routing Protocol (see [EIGRP Protocol Chosen](#), page 6-68).
- None—Specifies parameters for cable services (see [None Chosen: Cable Services](#), page 6-72).

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 6-73](#).

---

## 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 6-150](#)

## 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 Provisioning configures **the default-info originate** command on the PE router under the running protocol (for RIP, OSPF, or EIGRP). For Static, Prime Provisioning 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 6-99.

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 6-150
- 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 Provisioning 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 Provisioning configures **the default-info originate** command on the PE router under the running protocol (for RIP, OSPF, or EIGRP). For Static, Prime Provisioning 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 Provisioning issues a **default-information-originate** command under the iBGP address family for the currently specified VRF.



Tip

You must enable the **Redistribute Connected** option when joining the management VPN and you are also using IP numbered addresses.

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 6-73](#).

---

## 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 6-150](#)

**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 Provisioning configures **the default-info originate** command on the PE router under the running protocol (for RIP, OSPF, or EIGRP). For Static, Prime Provisioning 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 Provisioning 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 Provisioning 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 Provisioning 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 Provisioning imports the EIGRP routes into the PE that is running RIP.

**Parameter:** EIGRP autonomous system (AS) number

**Metric:** Any numeral from 1 to 16

- Choose the protocol you want to redistribute into RIP on the PE.
- Enter the appropriate parameter for the protocol selected.
- Click **Add**.
- 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:

- From the **Redistribute Protocols on CE** option, click **Edit**.

The CE Redistributed Protocol dialog box appears.

- Click **Add**.

The CE Redistributed Protocols dialog box appears.

- 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 Provisioning 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 Provisioning 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 Provisioning 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 Provisioning 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 Provisioning 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 Provisioning 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 Provisioning imports the IS-IS routes into the CE that is running RIP.

**Parameter:** IS-IS tag number

- Choose the protocol you want to redistribute into RIP on the CE.
- Enter the appropriate parameter for the selected protocol.
- Click **Add**.
- 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 6-73](#).



**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 Provisioning does not remove all of the RIP CLI commands associated with the interface from the PE configuration file. Specifically, Prime Provisioning 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 Provisioning 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 Provisioning 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 Provisioning 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 6-150](#)

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 Provisioning 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 Provisioning 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 Provisioning 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 6-201.

**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 Provisioning 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 MVRFCCE 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 MVRFCCE 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 Provisioning 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 6-104](#).

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 Provisioning template variables support Prime Provisioning 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 Provisioning 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 Provisioning 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 Provisioning 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, Prime Provisioning 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 Provisioning 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 Provisioning 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 Provisioning 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 6-199.

**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 6-73.

## 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 6-150](#)
- 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 Provisioning 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.
- Check **OSPF Default Information Originate Always** to advertise the default route regardless of whether the routing table has a default route.
  - For **Metric Value**, enter an OSPF metric to be used for generating the default route. Range is 1–16777214.
  - 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**
  - 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 Provisioning 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 6-205.

**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 Provisioning 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 Provisioning 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 Provisioning 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 Provisioning, see [OSPF Process ID for the IGP \(IOS XR Only\)](#), page 6-67.

**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 Provisioning 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 Provisioning ignores the OSPF domain ID configuration.
  - If Use VPN or VRF Domain ID is enabled, at the time of provisioning Prime Provisioning gets the OSPF domain ID from the selected VPN object. If an OSPF domain ID is not configured in the VPN object, Prime Provisioning 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 Provisioning 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 Provisioning 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 Provisioning, see [OSPF Process ID for the IGP \(IOS XR Only\)](#), page 6-67.

**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 Provisioning 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 Provisioning 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 Provisioning 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 Provisioning 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 Provisioning 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 Provisioning 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 Provisioning 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, Prime Provisioning 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 Provisioning 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 Provisioning 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 6-73](#).

## 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 Provisioning 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 Provisioning 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 Provisioning 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 Provisioning 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 Provisioning 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 a 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 6-150](#)  
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 6-207](#).
- 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 Provisioning 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 Provisioning 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 Provisioning 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 Provisioning 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 Provisioning 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 Provisioning 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, Prime Provisioning 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 Provisioning 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 Provisioning 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 Provisioning 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 6-73](#).

## 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 6-4](#).

**Figure 6-4** No Routing Protocol Selected

The screenshot shows a 'Policy Editor' window with 'MPLS' selected in the 'Policy Type' dropdown. Below is a table titled 'PE-CE IPv4 Routing Information' with an 'Editable' column. The 'Routing Protocol' is set to 'NONE'. Other options like 'CsC Support', 'Redistribute Static (BGP only)', and 'Redistribute Connected (BGP only)' are unchecked. At the bottom are 'Back', 'Next', 'Finish', and 'Close' buttons.

| PE-CE IPv4 Routing Information     |                          | Editable                            |
|------------------------------------|--------------------------|-------------------------------------|
| Routing Protocol:                  | NONE                     | <input checked="" type="checkbox"/> |
| CsC Support:                       | <input type="checkbox"/> | <input checked="" type="checkbox"/> |
| Redistribute Static (BGP only):    | <input type="checkbox"/> | <input checked="" type="checkbox"/> |
| Redistribute Connected (BGP only): | <input type="checkbox"/> | <input checked="" type="checkbox"/> |

Buttons: Back, Next, Finish, Close

- 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 6-150](#)
- 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 6-73](#).
- 

## 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 6-5](#).

**Figure 6-5 Specifying the VRF Information**

Policy Editor

Policy Type:

| VRF Information                   |                                     | Editable                            |          |              |        |                                                                                                                                                    |
|-----------------------------------|-------------------------------------|-------------------------------------|----------|--------------|--------|----------------------------------------------------------------------------------------------------------------------------------------------------|
| Use VRF Object:                   | <input type="checkbox"/>            | <input checked="" type="checkbox"/> |          |              |        |                                                                                                                                                    |
| Export Map:                       | <input type="text"/>                | <input checked="" type="checkbox"/> |          |              |        |                                                                                                                                                    |
| Import Map:                       | <input type="text"/>                | <input checked="" type="checkbox"/> |          |              |        |                                                                                                                                                    |
| Maximum Routes (32-5000000):      | <input type="text"/>                | <input checked="" type="checkbox"/> |          |              |        |                                                                                                                                                    |
| Maximum Route Threshold (1-100):  | <input type="text" value="80"/>     | <input checked="" type="checkbox"/> |          |              |        |                                                                                                                                                    |
| VRF Description:                  | <input type="text"/>                | <input checked="" type="checkbox"/> |          |              |        |                                                                                                                                                    |
| BGP Multipath Information         |                                     |                                     |          |              |        |                                                                                                                                                    |
| BGP Multipath Load Sharing:       | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> |          |              |        |                                                                                                                                                    |
| BGP Multipath Action:             | <input type="text" value="eBGP"/>   | <input checked="" type="checkbox"/> |          |              |        |                                                                                                                                                    |
| Maximum Paths (1-32) * :          | <input type="text" value="22"/>     | <input checked="" type="checkbox"/> |          |              |        |                                                                                                                                                    |
| Import Paths (1-32):              | <input type="text" value="22"/>     | <input checked="" type="checkbox"/> |          |              |        |                                                                                                                                                    |
| Allocate New Route Distinguisher: | <input type="checkbox"/>            | <input checked="" type="checkbox"/> |          |              |        |                                                                                                                                                    |
| VRF And RD Overwrite:             | <input type="checkbox"/>            | <input checked="" type="checkbox"/> |          |              |        |                                                                                                                                                    |
| VPN Selection                     |                                     |                                     |          |              |        |                                                                                                                                                    |
| PE VPN Membership:                |                                     | <input checked="" type="checkbox"/> |          |              |        |                                                                                                                                                    |
| #                                 | Customer                            | VPN                                 | Provider | Route Target | Is Hub |                                                                                                                                                    |
|                                   |                                     |                                     |          |              |        | <input type="button" value="Add"/> <input type="button" value="Delete"/>                                                                           |
|                                   |                                     |                                     |          |              |        | <input type="button" value="Back"/> <input type="button" value="Next"/> <input type="button" value="Finish"/> <input type="button" value="Close"/> |

Note: \* - Required Field

2-388802

**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 6-14](#). 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 Provisioning software to define a management VPN, Prime Provisioning 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 Provisioning 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 Provisioning 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 Provisioning 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 6-77](#), 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 Provisioning 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 6-11](#).

**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 6-6](#), 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 6-11](#).

**Figure 6-6 VRF and RD Overwrite Options**

|                       |                                     |                                     |
|-----------------------|-------------------------------------|-------------------------------------|
| VRF And RD Overwrite: | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> |
| VRF Name:             | <input type="text" value="VRF 3"/>  | <input checked="" type="checkbox"/> |
| RD Value:             | <input type="text" value="100:45"/> | <input checked="" type="checkbox"/> |

2388807

- 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**

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, then while attempting to change these values, Prime Provisioning will notify you with an error message since editing these values can alter or disable currently running service requests. If you want to change the values of the VRF Name and RD Value attributes on a deployed service request, you must decommission and purge the service request and create a new service request with the new values. 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.

- c. **(Optional) RD Auto:** Check the **RD Auto** check box to create, modify, and decommission MPLS Service Request with auto RD Value for ASR9k devices.

Figure 6-7 RD Auto

**Note**

If the selected device is ASR9k and RD value is “auto,” you can save MPLS Service Request without an error message. If the selected device is IOS and the specified Rd value is “auto” an error message will be displayed while creating an MPLS Service Request.

- 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 6-81](#) 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 6-82](#)

## BGP Multipath Load Sharing and Maximum Path Configuration

Prime Provisioning 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 6-8](#).

**Figure 6-8** Multipath Configuration Options of the VRF and VPN Membership Window

|                             |                                     |                                     |
|-----------------------------|-------------------------------------|-------------------------------------|
| BGP Multipath Load Sharing: | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> |
| BGP Multipath Action:       | eBGP                                | <input checked="" type="checkbox"/> |
| Maximum Paths (1-32) * :    | 22                                  | <input checked="" type="checkbox"/> |
| Import Paths (1-32) :       | 22                                  | <input checked="" type="checkbox"/> |

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 6-81](#), 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 6-80](#).

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.
- BG\_PIC—Specifies the multipath configuration for BGP additional paths CLI format.
- 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 6-11](#).

## BGP Multipath Support for IOS XR Devices

The following attributes are supported in Prime Provisioning 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.

## BGP Multipath Support for ASR 920, ASR 902 and ASR 903 Devices

To create MPLS services with BGP additional path information support for install or backup CLI format, choose the **BGP\_PIC** option from the **BGP Multipath Action** drop-down list. The **Install** and **Select Backup** fields appear, where you can select any of one or both of the option:

- **Install**: Check this check box to view the Install additional path configuration CLI.
- **Select Backup**: Check this check box to view the backup additional path configuration CLI.

*Figure 6-9 BGP Additional Path Attributes*

This is applicable only for ASR920, ASR 902 and ASR 903 devices. You can view the additional paths configuration in CLI as shown in the example:

```
!
router bgp 31213
 address-family ipv4 vrf $VRF_NAME
 redistribute connected
```

```

 bgp additional-paths select backup
 bgp additional-paths install
 exit-address-family
router bgp 64512
 address-family ipv4 vrf $VRF_NAME
 redistribute static

 bgp additional-paths select backup
 bgp additional-paths install
 exit-address-family

```

## 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 Provisioning 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 Provisioning.

**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 D, “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 11, “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 6, “MPLS VPN Service Requests.”](#)

## Customizing EVC and MPLS Policies

You can embed customized command line interface (CLI) templates into EVC and MPLS policies. You can also extend policies by adding attributes that you define directly in the policy screen. For more information, see [Chapter 8, “Customizing EVC, MPLS and MPLS-TP Policies”](#).

## MPLS VPN Service Requests

This section contains the following sections:

- [Service Enhancements, page 6-82](#)
- [How Prime Provisioning Accesses Network Devices, page 6-83](#)
- [Examples of Creating MPLS VPN Service Requests, page 6-83](#)
- [Migrating PE Devices from IOS to IOS XR, page 6-106](#)
- [Pseudowire access into an L3VPN, page 6-106](#)
- [Pseudowire Headend Interface, page 6-108](#)

To apply MPLS VPN policies to network devices, you must deploy the service request. When you deploy a service request, Prime Provisioning compares the device information in the Repository (the Prime Provisioning database) with the current device configuration and generates a configlet. Additionally, you can perform various monitoring and auditing tasks on service requests. These common tasks that apply to all types of Prime Provisioning service requests are covered in [Chapter 10, “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 Provisioning, 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 Provisioning 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 Provisioning 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
- Allows you to create, modify, and decommission MPLS SR with auto RD value for ASR 9k devices



**Note** If the selected device is ASR9k and Rd values is “auto”, you can save MPLS SR without an error message. If the selected device is IOS and the specified Rd value is “auto” an error message will be displayed while creating an MPLS SR.

## How Prime Provisioning Accesses Network Devices

When Prime Provisioning 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 Provisioning uses the terminal server to access the device.
2. If there is no terminal server, Prime Provisioning looks for the management interface on the device.
3. If there is no management interface, Prime Provisioning 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 Provisioning 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 Provisioning 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 Provisioning 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 6-40](#)

**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 6-109](#) and [Provisioning Multi-VRFCE PE-CE Links, page 6-121](#)

## MPLS VPN Topology Example

[Figure 6-10](#) 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

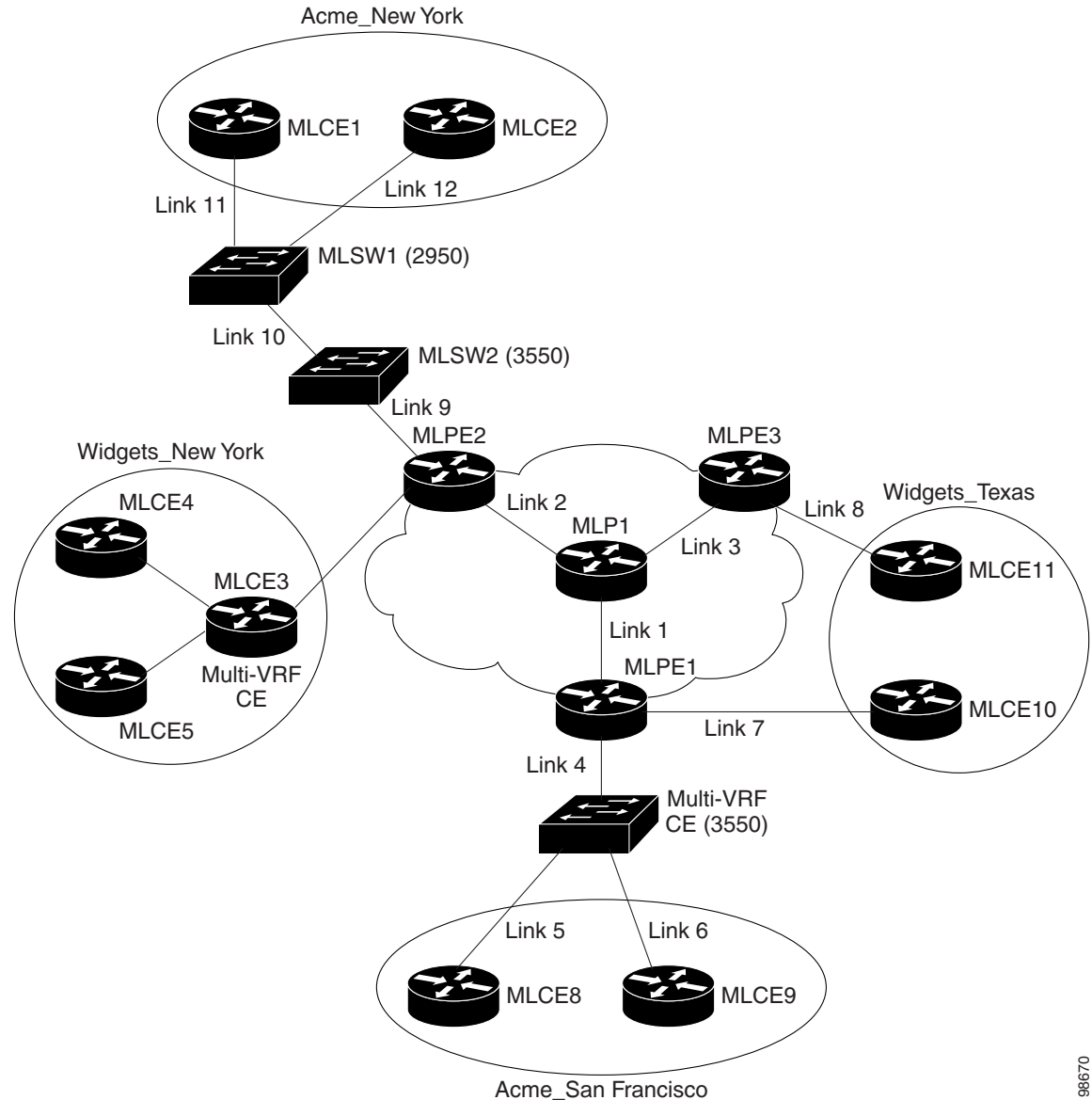
In the Multi-VRF example, the service provider needs to create an MPLS service between a CE (mlce4) in their customer site Widgets\_NY (in New York) and a Multi-VRFCE (mlce3) located in their customer site Widgets\_NY (in New York).

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 6-10 Example Network Topology



98670

## 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 > Service Request Manager > Create**.
  - Step 2 Choose the policy of choice, then click **OK**.  
Or select a policy from the **Service Design > Policy Manager** page and click **Create Service Request**.  
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** Check the **Allow Duplicate IP address** checkbox, if you want to allow duplication of IP address between the Primary links and Standby links within a single MPLS Service Request or between the different Service Requests.

This helps to configure two interfaces (channelized T1/T3, MLPPP) on different routers or in the same router with different interface cards. One interface as the Primary which is active, and the other as a Standby, with the same configuration and IP address.



**Note** This feature is not supported when **Automatically Assign IP Addresses field** is chosen. In such instance, Prime Provisioning fetches the next available IP address from the resource pool, even if Allow Duplicate IP Address is chosen.

- Step 5** **CE:** Click **Select CE**.

The Select CPE Device window appears.

- From the “Show CPEs with” drop-down list, you can display CEs by Customer Name, by Site, or by Device Name.
- You can use the **Find** button to either search for a specific CE, or to refresh the display.
- You can set the “Rows per page” to **5, 10, 20, 30, 40**, or **All**.
- 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 6** 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 7** **CE Interface:** Choose the CE interface from the interface picker.

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/hr381bun.html#wp1410649](http://www.cisco.com/en/US/docs/ios_xr_sw/iosxr_r3.8/interfaces/command/reference/hr381bun.html#wp1410649)

[http://www.cisco.com/en/US/docs/ios\\_xr\\_sw/iosxr\\_r3.8/multicast/configuration/guide/mc38mcast.html#wp1168111](http://www.cisco.com/en/US/docs/ios_xr_sw/iosxr_r3.8/multicast/configuration/guide/mc38mcast.html#wp1168111)



[http://www.cisco.com/en/US/docs/ios\\_xr\\_sw/iosxr\\_r3.8/multicast/configuration/guide/mc38mcst.html#wp1290965](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](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 8 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 9** 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 10 PE Interface:** Choose the PE interface from the interface picker

Note that the Link Attribute **Add** option is now enabled.

See the section [Note on Using Bundle-Ether Interfaces, page 6-86](#), for information on specifying Bundle-Ether interfaces.

**Step 11** 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 6-43](#).




---

**Note** On successful deployment of SR, interfaces and description of interfaces are automatically updated in the device inventory and on successful decommission of SR, any virtual or logical interfaces, created during SR deployment are removed from the device inventory.

---

**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.

- The autopick option for second VLAN ID is also available and it appears only when **Use Virtual Interface** and **Use EVC Service** check boxes are checked. The autopick value is allocated from the EVC Inner VLAN Resource Pool. In case of manual entry, both **Use Virtual Interface** and **Use EVC Service** check boxes are required to be unchecked and a value must be entered in the Second VLAN ID field. 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 Provisioning 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 6-187](#)
  - [CE-PE L3 MPLS VPN \(Q-in-Q/Second VLAN ID, IOS XR\), page 6-189](#)
  - [Frequently Asked Questions, page 6-216](#)

**Step 12** 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 6-46](#).

**Step 13** 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 6-49](#).

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 6-99](#).

**Step 14** 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 6-73](#).



**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 6, “Independent VRF Management.”](#) That section describes how to use independent VRF objects in MPLS VPN service policies and service requests.

**Step 15** 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 16** 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 6-73](#). 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 6-91](#) for information on defining VRF and VPN attributes during service request creation.

**Step 17** The next 2 screens of the policy editor are to define additional attributes and associate the policy with templates. See [Chapter 11, “Managing Templates and Data Files”](#). If you need to add attributes or templates click **Next**, else you can click **Finish**.

**Step 18** 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 11, “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 19** 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 20** 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. The link you have just defined can be activated in the network by deploying the service request as described in [Migrating PE Devices from IOS to IOS XR, page 6-106](#).

### Creating MPLS Service with IP Helper Address for ASR920, ASR 902 and ASR 903 Devices

You can provide DHCP Helper IP parameters for ASR920, ASR 902, ASR 903 devices, while creating MPLS service requests for the selected PE interface type.

In the MPLS Service Request Editor-Interface area, check the **Use EVC Service** check box to enable the following DHCP Helper IP fields:

- **DHCP Helper Server IP (a.b.c.d)**
- **DHCP Helper GRT IP (a.b.c.d)**
- **DHCP Helper VRF IP (a.b.c.d)**
- **MTU (68 - 9216)**

**Figure 6-11** DHCP Helper IP Addresses

The screenshot shows the configuration interface for an MPLS service request. The following fields are visible:

- VLAN ID\* (1-4094): 1332
- Use Virtual Interface:
- Use EVC Service:
- Service Instance ID:
- EVC Name:
- DHCP Helper Server IP (a.b.c.d): 13.2.4.5
- DHCP Helper GRT IP (a.b.c.d): 11.8.3.2
- DHCP Helper VRF IP (a.b.c.d): 14.1.1.9
- MTU (68 - 9216): 72
- Outer VLAN ID (1-4096): 112

The DHCP Helper IP fields (Server IP, GRT IP, VRF IP, and MTU) are highlighted with a red box.

You can view the IP helper address details of a server, VRF, GRT, MTU in CLI under the interface node as shown in the example:

```
!
interface BDI512
 description By VPNSC: Job Id# = 5
 vrf forwarding V1:MplsVpn1
 ip mtu 501
 ip address 10.12.15.17.255.255.224.0
 no shutdown
 ip helper-address 1011.14.18
 ip mtu 1504
 ip helper-address global 11.13.15.17
```

```
ip helper-address vrf V1:mplsvpn 12.14.18.22
```

## 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 6-73](#) in [MPLS VPN Service Policies, page 6-40](#). 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 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 6-94](#).

**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 Provisioning 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 only 16 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 Provisioning 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 6-92](#).

### 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, then while attempting to change these values, Prime Provisioning will notify you with an error message since editing these values can alter or disable currently running service requests. If you want to change the values of the VRF Name and RD Value attributes on a deployed service request, you must decommission and purge the service request and create a new service request with the new values. 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 Provisioning 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 Provisioning 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 Provisioning 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 Provisioning automatically changes the Route Targets tagging to the selected address family.
- Prime Provisioning 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 6-214.

### 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 6-14

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 Provisioning, see [Enabling a Unique Route Distinguisher for a VPN](#), page 6-11.

---

- 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 Provisioning 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 deleted.
  - The VRF service request is decommissioned, deleted, and redeployed.
- Unique RD can be enabled for the VRF.

**Step 6** Click **Next** to continue setting the MPLS link attributes.

---

## Extending VLAN-ID Range to Support BVI Interfaces with Values Range

You can create MPLS SR with VLAN-ID that ranges between 1-65535 for L3VPN/MPLS services on ASR9K series devices, only when the **EVC Service** is enabled.



**Note** This functionality supports only PE devices of IOS-XR type.

---

When you enable the **EVC Service** check box in the **MPLS Service Request Editor** page, the VLAN-ID value is automatically updated/changed to support VLAN-ID ranges of 1- 65535.

It is recommended to enter the VLAN-ID ranges within the specified range otherwise, an exemption occurs in the following instances:

- By clicking **Next**, an invalid VLAN-ID error message is displayed when the value entered is greater than 65535.
- If the selected device is IOS and the given VLAN-ID value is more than 4094 an error message is displayed.



**Note**

Enabling EVC service on L3VPN creates BVI virtual interfaces to allow routing through BVI interfaces. BVI interfaces can be configured in the range of 1-65535. This is because Prime Provisioning does not have an option to provide BVI interface and thus uses the VLAN- ID value as BVI interface value. The extended value of 1-65535 specified in the **VLAN-ID** range field allows you to create BVI interface with BVI interface number range 1-65535.

---

## Supporting EVPN-VPWS Attributes through a Customization XML File

From this release, EVPN-VPWS configurations are supported by importing a EVPN-VPWS customization.xar file as part of Prime Provisioning policies and service requests to manage various Ethernet Virtual Circuit services.

**Before you Begin**

To view or edit the EVPN-VPWS attributes in the **Policy Editor** window or the **EVC Service Request Editor** window, make sure that you have completed the following prerequisite.

1. Import the **EVPNVPWSCustomization.xar** file under \$PRIMEP\_HOME/packages/std, which is already copied as part of Prime Provisioning product (PP6.8.2 or greater). The customization XML uses merge mode as “Combine”. For more information about how to Import customization files, see [Importing and Exporting Customizations](#) section in the Cisco Prime Provisioning 7.1 User Guide.

**Note**

The EVPN-VPWS attributes are visible in the **Policy Editor** or the **EVC Service Request Editor** window only when you import the **EVPNVPWSCustomization.xar** file.

2. Make sure that the supported platform includes ASR9K with IOS-XR version 6.0 or later.

**Enter or Modify EVPN-VPWS Attributes**

To enter the EVPN-VPWS Attributes:

1. Create an EVC PW policy and import the “EVPNVPWSCustomization.xml”. Below is a configuration sample:

```
interface GigabitEthernet0/1/0/13.315 l2transport
 description EVC-JOBID:53
 encapsulation dot1q 315
 no shutdown
l2vpn
 xconnect group ISC
 p2p ELINE888
 interface GigabitEthernet0/1/0/13.315
 neighbor evpn evi 4341 target 64123 source 64222
```

2. In the **Policy Editor** window, in the **Source ac-id** field, enter the range between 1 and 16777215.
3. In the **Remote ac-id** field, enter the range between 1 and 16777215.
4. Click **Finish** to save the EVC Policy.

**Note**

You can also create an SR (VPWS) using this policy with XR devices and enter the values for the attributes.

**Supporting EVPN-VPLS Attributes through a Customization XML File**

From this release, EVPN-VPLS configurations are supported by importing a EVPN-VPLS customization.xar file as part of Prime Provisioning policies and service requests to manage various Ethernet Virtual Circuit services.

**Before you Begin**

To view or edit the EVPN-VPLS attributes in the **Policy Editor** window or the **EVC Service Request Editor** window, make sure that you have completed the following prerequisite.

1. Import the **EVPNVPLSCustomization.xar** file under \$PRIMEP\_HOME/packages/std, which is already copied as part of Prime Provisioning product (PP6.8.2 or greater). The customization XML uses merge mode as “Combine”. For more information about how to Import customization files, see [Importing and Exporting Customizations](#) section in the Cisco Prime Provisioning 7.1 User Guide.

**Note**

The EVPN-VPLS attributes are visible in the **Policy Editor** or the **EVC Service Request Editor** window only when you import the **EVPNVPLSCustomization.xar** file.

2. Make sure that the supported platform includes ASR9K with IOS-XR version 6.2.1 or later and supports EVPN VPLS configurations.
3. Before configuring the EVPN-VPLS, configure BGP with new EVPN Address family as in the sample configuration below:

```
router bgp 64
 bgp router-id 1.100.100.100
 address-family l2vpn evpn
 !
 neighbor 2.100.100.100
 remote-as 64
 update-source Loopback0
 address-family l2vpn evpn
```

**Enable or Modify EVPN-VPLS Attributes**

To enter the EVPN-VPLS Attributes:

1. Create an EVC VPLS policy and import the “EVPNVPLSCustomization.xml”. Below is a configuration sample:

```
interface GigabitEthernet0/1/0/17.2017 l2transport
 description EVC-JOBID:216
 encapsulation dot1q 2017
 no shutdown
l2vpn
 bridge group BGNevpn1
 bridge-domain BDNevpn1
 mtu 345
 interface GigabitEthernet0/1/0/17.2017
 vfi wppq
 evi 2001
evpn
 evi 2001
 load-balancing flow-label static

evpn
 evi 2001
 advertise-mac

evpn
 evi 2001
 unknown-unicast-suppress

evpn
 evi 2001
 control-word-disable

evpn
 evi 2001
 bgp
 route-target import 200:101
 route-target export 200:101
```

2. In the **Policy Editor** window, the following EVPN-EVI attributes are added.

| Field                   | Description                                                                                                                                                        |
|-------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Load Balancing          | To provision the EVPN load balancing command and to view the load balancing policy name in the Service Request Manager, check the <b>Load Balancing</b> check box. |
| Enable Advertise Mac    | To provision advertise Mac command, check the <b>Enable Advertise Mac</b> check box.                                                                               |
| Unknown Unicast Supress | To provision the Unknown unicast supress command, check the <b>Unknown Unicast Supress</b> check box.                                                              |
| Control Word Disable    | To provision the control word disable command, check the Control Word Disable check box.                                                                           |
| Enable BGP              | To provision the BGP command, check the Enable BGP check box.                                                                                                      |
| Route Target Import     | Enter the format for Route Target Import. For example: 1234:5678([0-9]:[0-9]).                                                                                     |
| Route Target Export     | Provide the format for Route Target Import. For example: 1234:5678([0-9]:[0-9]).                                                                                   |



**Note**

You can also create an SR (VPLS) using this policy with XR devices and enter the values for the required 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 10-5](#). For sample configlets, see [Sample Configlets, page 6-177](#)

## 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 13](#) in the section [Creating an MPLS VPN PE-CE Service Request, page 6-85](#) 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 6-100](#).
- 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 13](#) in the section [Creating an MPLS VPN PE-CE Service Request, page 6-85](#) 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 6-100](#).
- 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 6-35](#).
- 

### Outgoing Interface Name + Next Hop IP Address Support for Static Route Configuration

Prime Provisioning 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 Provisioning 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 6-214.

## 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 > Create**.
- 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 interface picker.
- 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 interface picker.  
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 interface picker.
- Step 13** Click **Add** in the **Link Attribute** cell.  
The MPLS Link Attribute Editor - Interface window appears.




---

**Note** On successful deployment of SR, interfaces and description of interfaces are automatically updated in the device inventory and on successful decommission of SR, any virtual or logical interfaces, created during SR deployment are removed from the device inventory.

---

- 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 6-91.

**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 11, “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 > Create**.

**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 6-106.

**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 interface picker.

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 6-43](#).




---

**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 6-87](#).

---




---

**Note** On successful deployment of SR, interfaces and description of interfaces are automatically updated in the device inventory and on successful decommission of SR, any virtual or logical interfaces, created during SR deployment are removed from the device inventory.

---

**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 6-46](#).

**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 6-49](#).

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 Provisioning 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 6-35](#).

**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 6-73](#).



**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 6-14](#). 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 6-91](#).

**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 11, “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 6-106](#).
- 

## 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 6-103](#), perform the following steps:

- 
- Step 1** Follow [Step 1](#) through [Step 5](#) of [Creating a PE-Only Service Request, page 6-103](#).
- Step 2** Click **Select CLE**. The Select PE Device dialog box appears.
- From the “Show PEs with” drop-down list, you can display PEs by Provider Name, by Region, or by Device Name.
  - You can use the **Find** button to either search for a specific PE, or to refresh the display.
  - You can set the “Rows per page” to **5, 10, 20, 30, 40**, or **All**.
  - 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 interface picker.
- Step 5** Continue following [Step 4](#) through [Step 16](#) of “[Creating a PE-Only Service Request](#)” section on [page 6-103](#).
- 

## 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.

## Pseudowire access into an L3VPN

To enable the service deployment of pseudowire access into an L3VPN by selecting a bridge virtual interface (BVI), perform the following steps:

- 
- Step 1** Create and deploy an MPLS service that you can use to provision a BVI interface on an ASR9K device. See [“Working with MPLS Policies and Service Requests”](#) section on page 6-3 for more details.
- Step 2** Create an EVC Pseudowire service with the **Configure Bridge Domain** check box enabled (available under Pseudowire Core Connectivity attributes). See the [“Creating an EVC Service Request”](#) section on page 3-23 for more details.
- Step 3** Add a link using the ASR9K device and selecting an appropriate UNI interface depending on the service topology. See the [“Setting up Links to the N-PE”](#) section on page 3-24 for more details.
- Step 4** Click **Edit** in the Link Attributes column to specify the UNI attributes for the ASR9K device. The Standard UNI Details window is displayed.
- Step 5** Enable the **Use BVI** check box (only for IOS-XR) and select an appropriate BVI interface created from the Configuration Collection for the device in Step-2.
- Step 6** Enter other required link attributes and deploy the service.
- The service deployment of the Pseudowire into L3VPN is now enabled. The configlet that is pushed into the device is highlighted below:

#### Configlet deployed on the L3 service:

```
vrf V8:vpnX2
 address-family ipv4 unicast
 import route-target
 64512:10002
 64512:10003
 export route-target
 64512:10002
interface BVI780
 description By VPNSC: Job Id# = 24
 vrf V8:vpnX2
 ipv4 address 40.10.10.141 255.255.255.252
 no shutdown
 router bgp 64512
 vrf V8:vpnX2
 rd 64512:10006
 label-allocation-mode per-vrf
 address-family ipv4 unicast
 redistribute static
```

#### Configlet deployed when the BVI interface is used in the L2 service:

```
l2vpn
 bridge group cisco
 bridge-domain domain50
 Interface GigabitEthernet0/1/0/0.50
```

```

routed interface bvi 780
neighbor 1.2.3.4 pw-id 55

```

---

## Pseudowire Headend Interface

Using Prime Provisioning, you can now configure an L3 VPN attachment circuit, with a Pseudowire access. Using the Pseudowire Headend feature on the ASR9000, this can be achieved without terminating the Pseudowire on an Ethernet interface, or allocating bridge domain for this purpose. This enables the creation of an end to endMPLS network where access is provided by a small switch that does not support many L3VPN instances. On that switch you configure a pseudowire which terminates on the ASR9000. There the pseudowire is directly connected to L3VPN.

To prepare to use this feature you need both an L3 VPN policy and a EVC policy:

- 
- Step 1** Navigate to the Policy Editor page.
  - Step 2** In the PE Interface details section, check the **Create virtual interface only** check box. This displays the **Configure Pseudowire Headend** check box.
  - Step 3** Check the **Configure Pseudowire Headend** check box to enable the pseudowire headend feature for the PE interface.
  - Step 4** Make other required changes and save the policy.  
Note that the **Configure Pseudowire Headend** check box is hidden until you select the **Create virtual interface only** check box. When you use this policy to create a service request, Prime Provisioning disables the PE Interface column in the Service Request Editor. When this service is deployed, Prime Provisioning creates a pseudowire-ether interface configured in the device.
  - Step 5** Create an EVC policy, this should have:
    - Core type- PSUEDOWIRE
    - For end to end MPLS which is the typical case, enable **CE directly connected to N-PE**.
    - Ensure that the **Configure Bridge Domain** check box is disabled.
- 

Then to create services, follow these steps:

- 
- Step 1** Navigate to **Operate > Service Request Manager**.  
The Service Request Manager window appears.
  - Step 2** Click **Create**.  
The Service Request Editor window appears.
  - Step 3** From the policy picker, choose the L3 policy that you created in steps 1-4.  
The L3 VPN Service Request editor window appears. This window enables you to specify options for the service request, as well as configure links.
  - Step 4** Create an EVC service request using the EVC policy created in step 5 above
  - Step 5** Set the pseudowire core connectivity attributes. See [Table 3-7Pseudowire Core Connectivity Attributes](#), for more details about the attributes.
  - Step 6** Set up links to the N-PE as described in section [Setting up Links to the N-PE](#).

- Step 7** When you have completed setting the attributes in the EVC Service Request Editor window, click the Save button to save the settings and create the EVC service request.
- Step 8** Now you are ready to deploy both service requests, see [Deploying Service Requests](#).

## Provisioning Regular PE-CE Links

This section describes how to configure MPLS VPN PE-CE links in the Prime Provisioning provisioning process.

### MPLS VPN PE-CE Link Overview

To provision an MPLS VPN service in Prime Provisioning, you must first create an MPLS VPN Service Policy. In Prime Provisioning, a Service Policy is a set of default configurations for creating and deploying a service request.

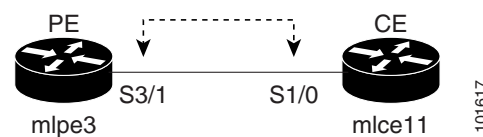
Prime Provisioning supports two MPLS VPN Service Policy Types: Regular PE-CE and MVRFCE PE-CE. The following scenarios focus on the Regular PE-CE Policy Type.

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 6-12](#) shows an example of a normal PE to CE link between two devices.

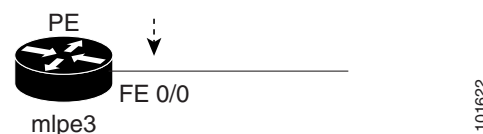
**Figure 6-12** PE to CE link with CE Present



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 6-13](#) shows an example of a PE Only link with no CE.

**Figure 6-13** PE to CE link with No CE

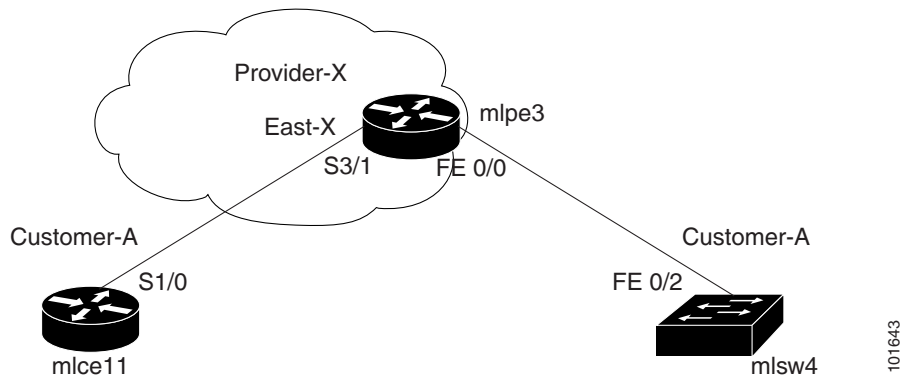


In a PE to CE link with CE Present disabled, interface FE0/0 is configured as an MPLS VPN link in the service request process.

## Network Topology

Figure 6-14 shows an overview of the network topology in which the MPLS VPN PE-CE links are created.

Figure 6-14 Network Topology for MPLS VPN PE-CE Scenarios.



The network topology in Figure 6-14 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 Provisioning, you must complete the following Service Inventory tasks:

- 
- Step 1 Set up a Customer with a Site (see [Managing Customer Premise Devices, page 2-34](#)).
  - Step 2 Set up a Provider with a Region (see [Providers, page 2-14](#)).
  - Step 3 Import, create, or discover Devices (see [Devices, page 2-1](#)).
  - Step 4 Create CPE and PE (see [Providers, page 2-14](#)).
  - Step 5 Collect Configurations (see [Tasks, page 12-1](#)).
  - Step 6 Create Resource Pools (see [Resource Pools, page 2-43](#)).
  - Step 7 Create Route Target(s) (see [Route Targets, page 2-52](#)).
  - Step 8 Define a MPLS VPN (see [Creating an MPLS VPN, page 6-7](#)).
- 

## Defining a VPN for the PE-CE Link

During service deployment, Prime Provisioning 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 Provisioning.

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 6-14](#)



**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 6-91](#).

## Creating MPLS VPN PE-CE Service Policies

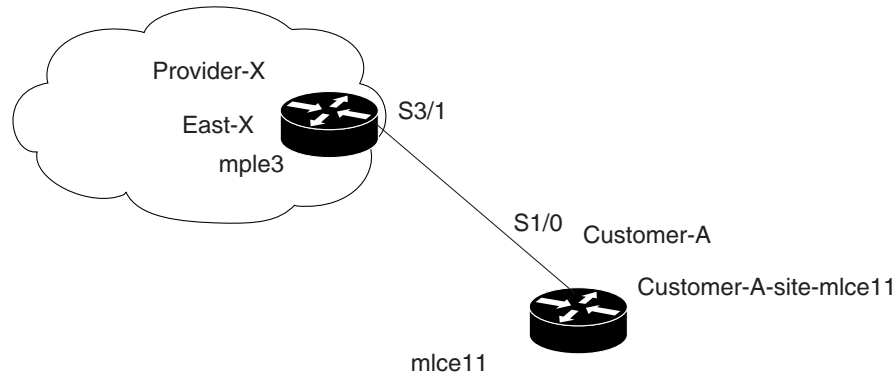
This section contains the following sections:

- [PE-CE Service Policy Overview, page 6-111](#)
- [Creating MVRFCPE PE-CE Service Policies, page 6-123](#)
- [Creating PE-NoCE Service Policies, page 6-125](#)

### PE-CE Service Policy Overview

[Figure 6-15](#) shows an example of the PE-CE link that is defined in the PE-CE Service Policy scenario.

Figure 6-15 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 > Policy Manager > Create**.  
The Policy Editor window appears.
- Step 2** Choose MPLS as the policy type.  
The Policy Editor window appears.
- Step 3** 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 4** **CE Present:** Check to set CE as present.
- Step 5** Click **Next**.  
The MPLS Policy Editor - Interface window appears.
- Step 6** 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 7** Edit all applicable attributes.




---

**Note** If you check **Automatically Assign IP Address**, the screen refreshes and adds a fourth attribute: **IP Address Pool**.

---

**Step 8** Click **Next**.

The MPLS Policy Editor - Routing Information window appears.

**Step 9** 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 6-49.

---




---

**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 6-14. That section describes how to use independent VRF objects in MPLS VPN service policies and service requests.

---

**Step 10** 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 D, “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 11, “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 11** 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 > Policy Manager > Create**.  
The Policy Editor window appears.

- Step 2** Choose MPLS as the policy type.  
The Policy Editor window appears.
- Step 3** 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 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.

---

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 6-46](#).

- Step 6** Edit all applicable attributes.




---

**Note** If you check **Automatically Assign IP Address**, the screen refreshes and adds a fourth 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 6-49](#).

---




---

**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 6-14](#). 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 D, “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 11, “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-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 6-126](#)
- [Creating MVRFCE PE-NoCE Service Requests, page 6-129](#)

## 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 > Create**.
- The Service Request Editor 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 interface picker.

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 interface picker.

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.




---

**Note** On successful deployment of SR, interfaces and description of interfaces are automatically updated in the device inventory and on successful decommission of SR, any virtual or logical interfaces, created during SR deployment are removed from the device inventory.

---

**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 Provisioning 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:** Enter the Second VLAN ID, the value must be an integer from 1 to 4094.

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.

**Auto-Pick Second VLAN ID:** The autopick option for second VLAN ID is also available and it appears only when **Use Virtual Interface** and **Use EVC Service** check boxes are checked. The autopick value is allocated from the EVC Inner VLAN Resource Pool.

If these check boxes are checked, the Second VLAN ID field is not visible in the GUI.

For usage details about this attribute, see [Notes on the VLAN ID and Second VLAN ID Attributes, page 6-87](#).

**Bridge Group Name:** If you check this auto pick **Bridge Group Name** check box and if the DCPL property is set to 'true', auto pick bridge group takes the VRF name instead of the provider name. For example, bridge group VRFtest1. It is recommended to use “\_” special character instead of special characters (. , : , <Space>) in the VRF name.

**Step 20** **Use SVI:** Check this box to have Prime Provisioning 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 6-49](#).

---

- Step 26** Choose a Next Hop Option:
- USE\_OUT\_GOING\_INTF\_NAME
  - USE\_NEXT\_HOP\_IPADDR (enables the BFD attribute)
  - OUTGOING\_INTF\_NAME+NEXT\_HOP\_IPADDR (enables the BFD attribute)




---

**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 6-99](#).

---

- Step 27** Specify the BFD values (enabled only when the Next Hop option is set to USE\_NEXT\_HOP\_IPADDR or OUTGOING\_INTF\_NAME+NEXT\_HOP\_IPADDR):
- BFD Minimum interval,
  - BFD Multiplier.

During service provisioning, Prime Provisioning ensures that configlets with the BDF values are generated only for IOS-XR devices. BFD configlets are generated only if you provide the value for the **Advertised Routes for CE** attribute. Without this value configlets will not be generated, even if BFD check box is enabled and values for BFD Minimum interval and Multiplier are specified. In the generated configlet, the BFD command is generated along with the route command and it is appended with advertised routes for CE. The new attributes that appear in the configlet are BFD Required, BFD Minimum Interval, and BFD Multiplier. These value are applicable to IPV4 and IPV6 devices.

- Step 28** 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 6-14](#). 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 6-91](#).

---

**Step 29** Click **Add** to join a VPN.

The Select CERCs window appears.

**Step 30** Choose a Customer from the drop-down list.

**Step 31** Choose a VPN from the drop-down list.

**Step 32** Check to choose a VPN from the list.

**Step 33** Click **Join As Hub** or **Join As Spoke**.

**Step 34** Click **Done**.

The MPLS Link Attribute Editor - VRF and VPN window reappears.

**Step 35** 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 11, “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 36** 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 37** 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 > Create**.

**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 interface picker.  
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.
- 
- Note** On successful deployment of SR, interfaces and description of interfaces are automatically updated in the device inventory and on successful decommission of SR, any virtual or logical interfaces, created during SR deployment are removed from the device inventory.
- 
- 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 Provisioning 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:** Enter the Second VLAN ID, the value must be an integer from 1 to 4094.  
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.
- Step 16** **Auto-Pick Second VLAN ID:** The autopick option for second VLAN ID is also available and it appears only when **Use Virtual Interface** and **Use EVC Service** check boxes are checked. The autopick value is allocated from the EVC Inner VLAN Resource Pool.  
If these check boxes are checked, the Second VLAN ID field is not visible in the GUI.
- Step 17** For usage details about this attribute, see [Notes on the VLAN ID and Second VLAN ID Attributes, page 6-87](#).
- Step 18** **Use SVI:** Check this box to have Prime Provisioning terminate VRF on SVI.
- Step 19** **Standard UNI Port:** Check this box to access additional UNI security parameters.
- Step 20** Click **Next**.  
The MPLS Link Attribute Editor - IP Address Scheme window appears.
- Step 21** 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 22** 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 6-49.

**Step 23** 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 6-14. 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 6-91.

**Step 24** Click **Add** to join the VPN.

The Join VPN dialog box appears.

**Step 25** Check to choose the VPN.

**Step 26** Click Join as Hub or Join as Spoke.

**Step 27** Click **Done**.

The MPLS Service Request Editor window reappears.

**Step 28** 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 11, “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 29** 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 30** 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 Provisioning provisioning process.

## MPLS VPN MVRFCE PE-CE Link Overview

This section contains the following sections:

- [Network Topology, page 6-122](#)
- [Prerequisite Tasks, page 6-122](#)

To provision an MPLS VPN service in Prime Provisioning, you must first create an MPLS VPN Service Policy. In Prime Provisioning, a Service Policy is a set of default configurations for creating and deploying a service request. Prime Provisioning supports two MPLS VPN Service Policy Types: Regular PE-CE and 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:

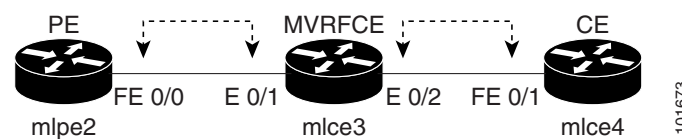
- PE
- Multi-VRF CE
- CE

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 6-16](#) shows an example of an MVRFCE PE-CE link with three devices.

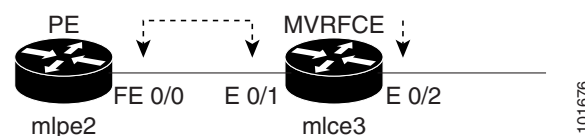
**Figure 6-16 MVRFCE PE-CE Link**



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 6-17](#) shows an example of a PE to MVRFCE link with no CE.

**Figure 6-17 MVRFCE PE-CE Link with No CE**

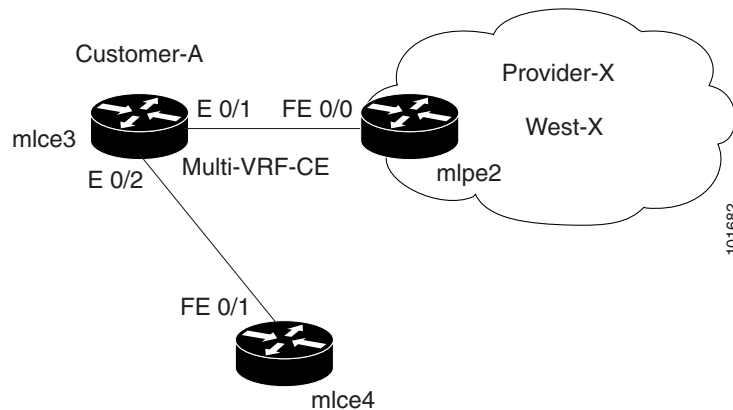


In an MVRFCE PE-CE link with CE Present disabled, interfaces FE 0/0, E 0/1, and E 0/2 are configured as an MPLS VPN link in the service request process.

## Network Topology

Figure 6-18 shows an overview of the network topology in which the MPLS VPN MVRFCE PE-CE links are created.

Figure 6-18 Network Topology for MPLS VPN MVRFCE PE-CE Scenarios



The network topology in Figure 6-18 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 Provisioning, you must complete the following Inventory Management tasks:

- 
- Step 1 Set up a Customer with a Site (see [Managing Customer Premise Devices, page 2-34](#)).
  - Step 2 Setup a Provider with a Region (see [Providers, page 2-14](#)).
  - Step 3 Import, create, or discover Devices (see [Chapter 2, “Devices”](#)).
  - Step 4 Create CPE and PE (see [Providers, page 2-14](#)).
  - Step 5 Collect Configurations (see [Tasks, page 12-1](#)).
  - Step 6 Create Resource Pools (see [Resource Pools, page 2-43](#)).
  - Step 7 Create CE routing communities (CERC) (see [Route Targets, page 2-52](#)).
  - Step 8 Define a MPLS VPN (see [Creating an MPLS VPN, page 6-7](#)).
- 

## Defining VPN for MVRFCE PE-CE Links

During service deployment, Prime Provisioning 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 Provisioning. 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 6-123](#)
- [Creating PE-NoCE Service Policies, page 6-125](#)

## 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 > Policy Manager**.  
The Policy Manager window appears.  
Choose the policy that you want to edit and click **Edit**.
- 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 6-49](#).

---

**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 D, "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 11, "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 > Policy Manager**.
- The Policy Manager 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 Provisioning 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 Provisioning 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 6-49.

**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 D, “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 11, “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, page 6-126](#)
- [Creating MVRFCE PE-NoCE Service Requests, page 6-129](#)

## 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 interface picker.
- 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 interface picker.
- 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.



---

**Note** On successful deployment of SR, interfaces and description of interfaces are automatically updated in the device inventory and on successful decommission of SR, any virtual or logical interfaces, created during SR deployment are removed from the device inventory.

---

#### 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 6-49](#).

---

**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 6-91](#).

---

**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 11, "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 interface picker.

**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.



**Note** On successful deployment of SR, interfaces and description of interfaces are automatically updated in the device inventory and on successful decommission of SR, any virtual or logical interfaces, created during SR deployment are removed from the device inventory.

---

### 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 6-49](#).

---

**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 6-91](#).

---

**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 6-91](#).

---

**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 11, "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.

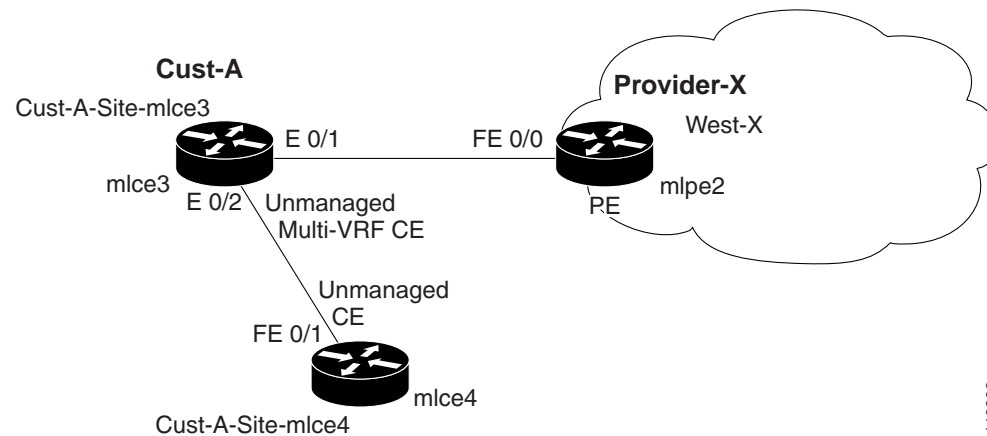
## Creating an Unmanaged MVRFCE

The unmanaged MVRFCE feature is similar to the unmanaged CE feature in so far as the service provider does not use Prime Provisioning to upload or download configurations to the CPE. This feature is similar to the managed MVRFCE feature in so far as Prime Provisioning 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 Provisioning to generate the configuration and then send it to the customer for manual implementation.

Figure 6-19 shows an overview of a network topology with MPLS VPN MVRFCE PE-CE links.

**Figure 6-19** Unmanaged MVRFCE PE-CE Network Topology



The network topology in Figure 6-19 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 Provisioning management subnet. Before Prime Provisioning 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.

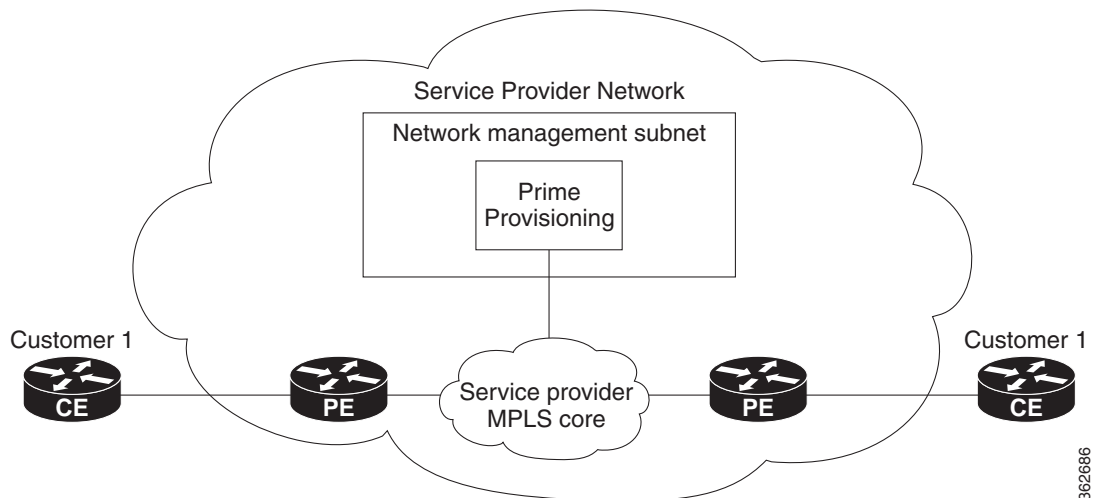
## Unmanaged Customer Edge Routers

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 Provisioning is not employed for provisioning or managing unmanaged CEs.

Figure 6-20 shows a basic topology with unmanaged CEs. The network management subnet has a direct link to the Service Provider MPLS core network.

Figure 6-20 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.

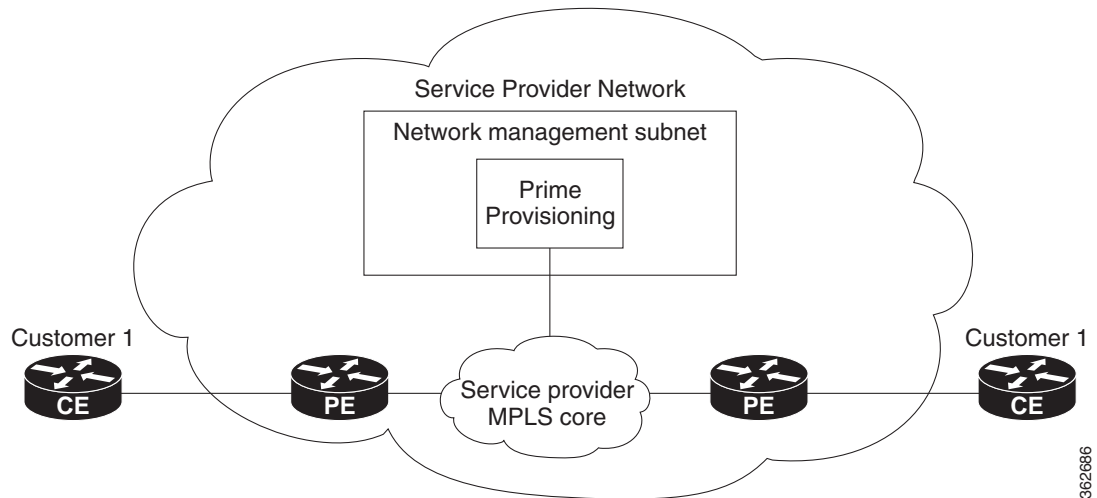
The following sections discuss the concepts and issues required for administering a managed CE environment.

## 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 6-21 shows the Prime Provisioning network management subnet and the devices that might be required to connect to it:

Figure 6-21 The Prime Provisioning 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 Provisioning 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 Provisioning.

Before you provision a CE in the Prime Provisioning, 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 Provisioning. You configure the MCE by identifying the CE as part of the management LAN in Prime Provisioning.

### 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 Provisioning recommends two main network management subnet implementation techniques:

- Management VPN Technique

The MPE-MCE link uses a Management VPN (see [Management VPN, page 6-136](#)) to connect to managed CEs. To connect to the PEs, the MPE-MCE link uses a parallel IPv4 link.

- Out-of-Band Technique

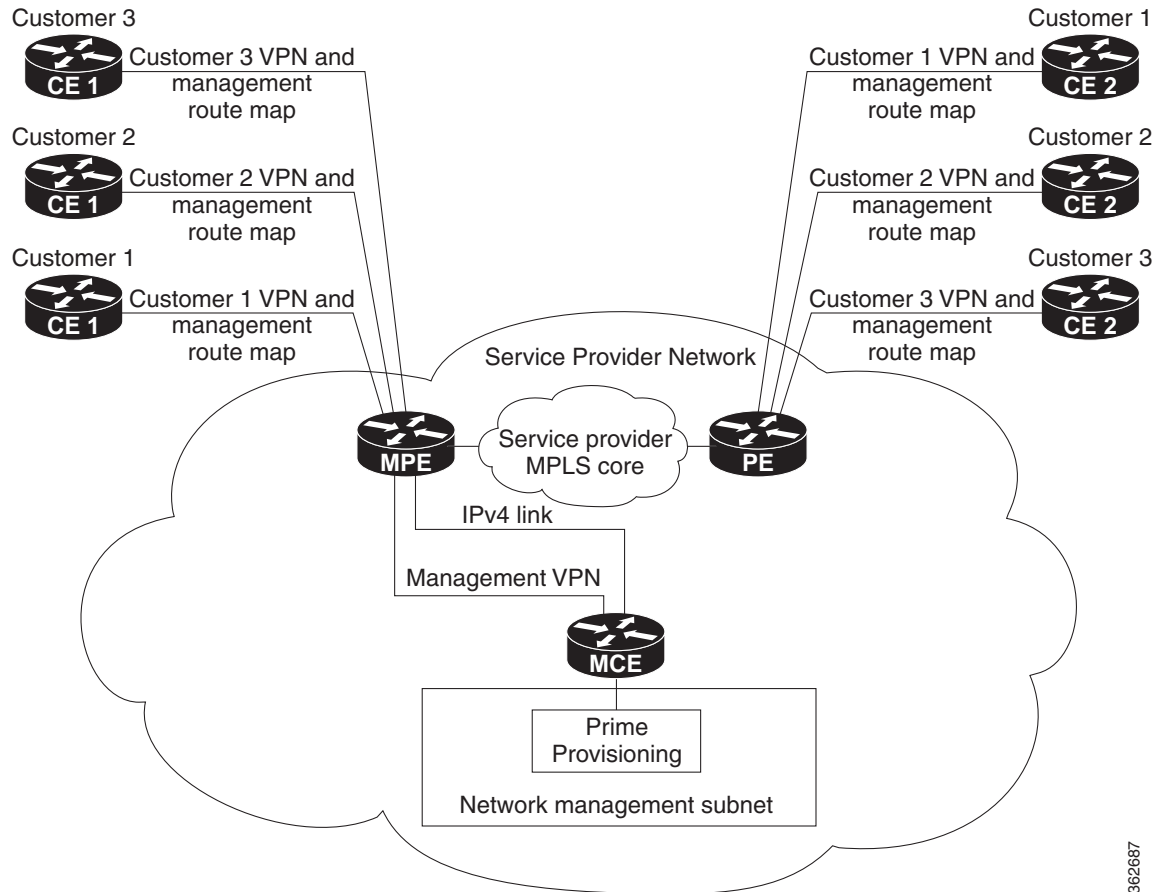
In the Out-of-Band technique, the MCE has IPv4 connectivity (that is, not MPLS VPN connectivity) to all the CEs and PEs in the network (see [Out-of-Band Technique, page 6-137](#)). In this context, *out-of-band* signifies a separate link between PEs that carries the provider's management traffic.

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 Provisioning. 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 6-22](#) shows a typical topology for the Management VPN technique.

**Figure 6-22 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 6-22](#), 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.

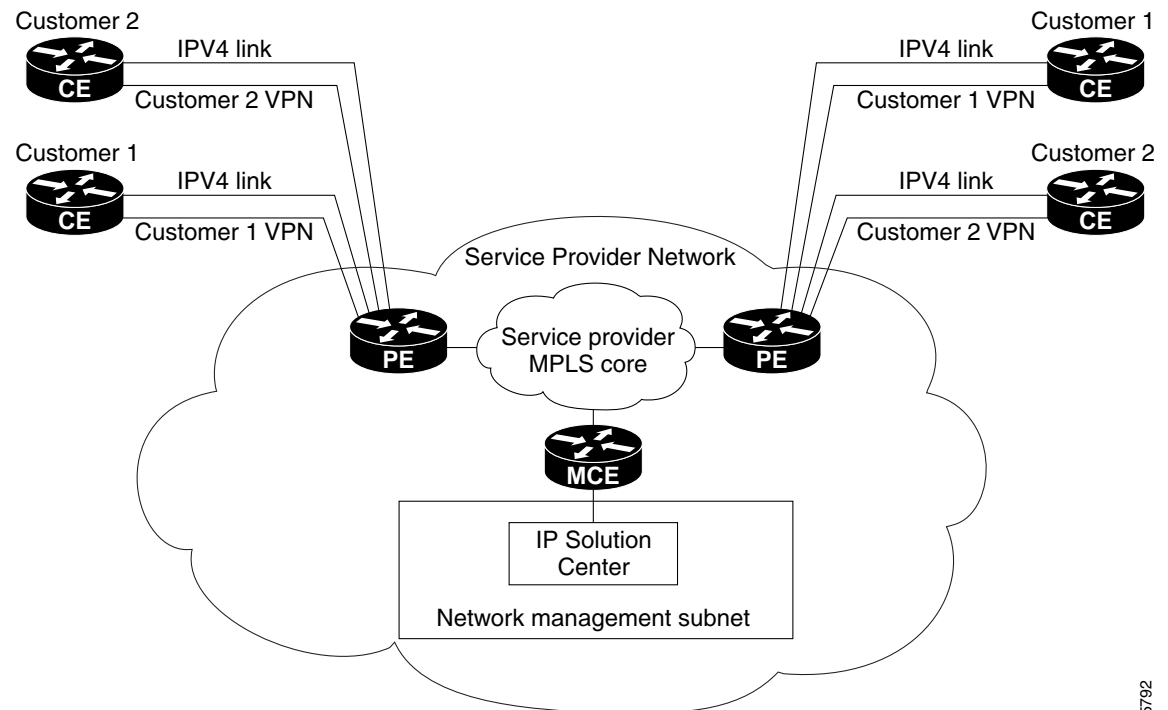
The advantages involved in implementing the Management VPN technique are as follows:

- 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 6-23](#), the MCE provides separation between the provider's routes and the customer's routes.

**Figure 6-23** 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 Provisioning

The Prime Provisioning 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 Provisioning.

### Defining CE as MCE

You configure the MCE by identifying the CE as part of the management LAN in Prime Provisioning 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 > Service Request Manager**.
- The Service Request Manager window appears.
- This window displays the list of all the MPLS service policies that have been defined in Prime Provisioning.
- 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.
- From the “Show CPEs with” drop-down list, you can display CEs by Customer Name, by Site, or by Device Name.
  - You can use the **Find** button to either search for a specific CE, or to refresh the display.
  - 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 interface picker.  
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 interface picker.  
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 6-43](#).



**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 6-87](#).

- 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 6-46](#).
- 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 6-49](#).  
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 6-73](#).




---

**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 6-91](#).

---

**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 11, “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 Provisioning 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 6-24](#), 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.

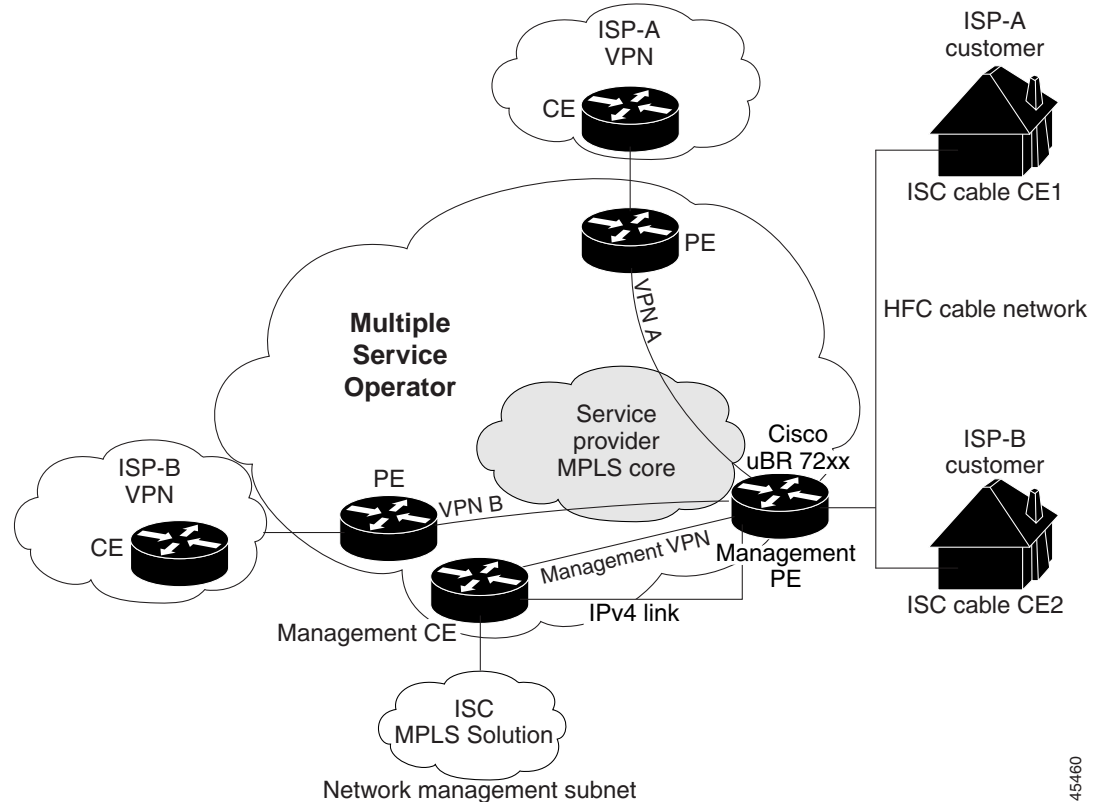
The routers in the cable network are as follows:

- 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 Provisioning.
- 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 6-24 Example of an MPLS VPN Cable Network



45460

## 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 Provisioning 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 6-132](#)

As shown in [Figure 6-24](#), the management VPN is comprised of the network management subnet (where the Prime Provisioning 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 Provisioning), 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 Provisioning, 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 Provisioning automatically selects the subinterface number based on the VRF. If a subinterface that is associated with the current VRF does not yet exist, Prime Provisioning 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 Provisioning) 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 Provisioning

The tasks you must complete to provision cable services in Prime Provisioning 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 Provisioning 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 an MPLS VPN PE-CE Service Request, page 6-85](#)
- [Creating Cable Link Service Requests, page 6-148](#)

## 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 > Service Request Manager**.
- 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 Provisioning.
- 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 interface picker.
- Only the major interface names are available for you to select. Prime Provisioning 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 6-49](#).

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 6-14](#). 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 6-91](#).

**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 11, “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 > Service Request Manager**.
- 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 Provisioning.
- 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 interface picker.
- 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 6-49](#).

---

**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 6-14](#). 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 6-91](#).

- 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 11, “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 Provisioning provisioning process. It contains the following sections:

- [Carrier Supporting Carrier Overview, page 6-151](#)



- [Defining CSC Service Policies, page 6-155](#)
- [Provisioning CSC Service Requests, page 6-155](#)

## 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 Provisioning, 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

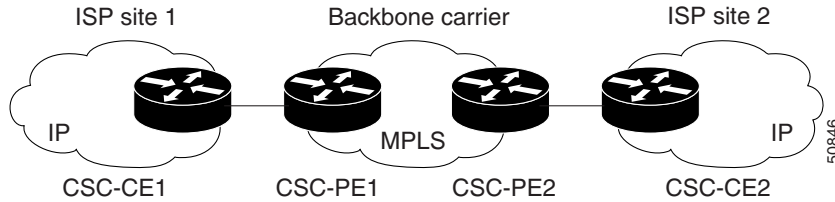
The CSC service is applicable for the BGP PE-CE routing protocol on IOS XR devices.

## 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 6-25](#) 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 6-25** 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:

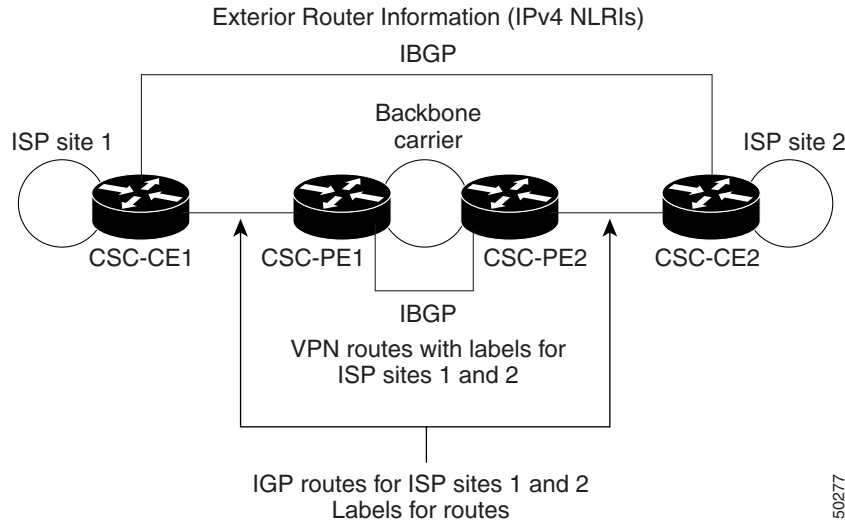
- Internal routes go to any of the routers within the ISP.
- External routes go to the Internet.

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 6-26 shows how information is exchanged when the network is configured in this manner.

**Figure 6-26 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 6-27 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 6-27 Carrier Supporting Carrier Network with a Customer Carrier Who Is an MPLS VPN Provider**

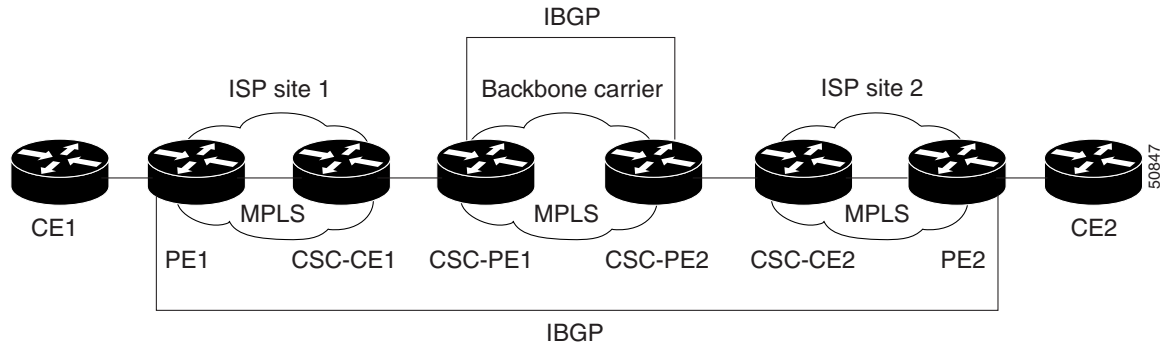
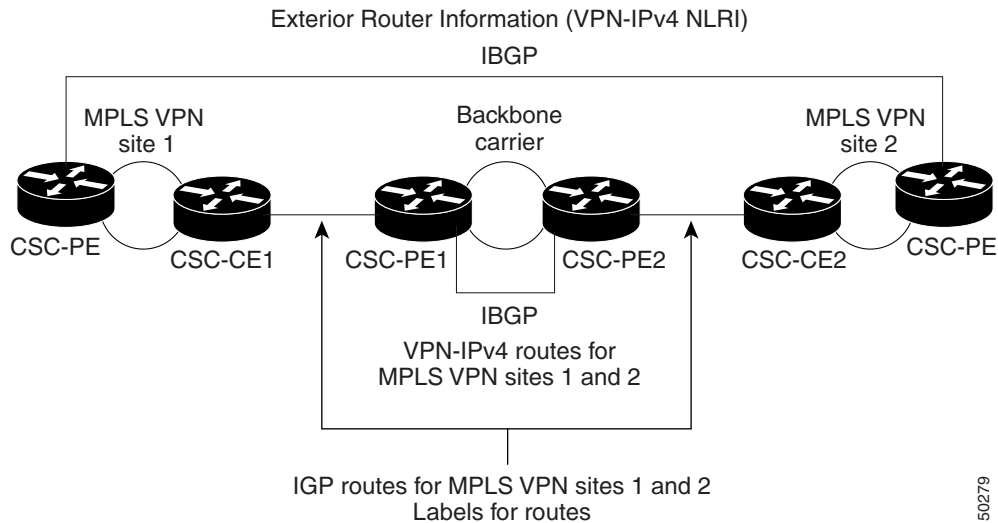


Figure 6-28 figure shows backbone carrier exchanging information with a customer carrier who is an MPLS VPN service provider.

**Figure 6-28** Backbone Carrier Exchanging Information with a Customer Carrier Who Is an MPLS VPN Service Provider



## Prime Provisioning 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 Provisioning provisioning process. It contains the following sections:

- [NPC Ring Topology, page 6-155](#)
- [Ethernet-To-The-Home \(ETTH\), page 6-159](#)

### 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 Provisioning provisioning process.

This section contains the following sections:

- [Ring Topology Overview, page 6-155](#)
- [Creating Ring of Three PE-CLEs, page 6-156](#)
- [Configuring NPC Ring Topology, page 6-157](#)

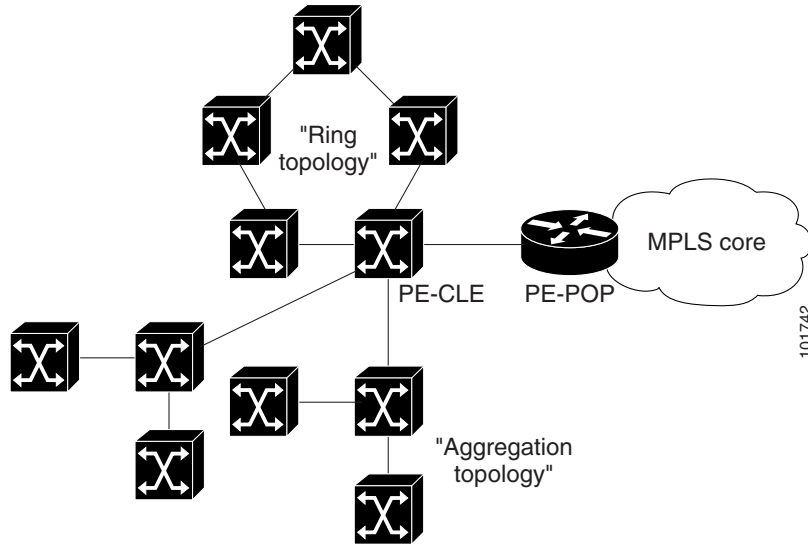
### Ring Topology Overview

Service providers are now looking to offer L2 and L3 services that must integrate with a common MPLS infrastructure. Prime Provisioning supports two basic L2 topologies to access L3 MPLS networks:

- Ring Topology
- Aggregation Topology (“Hub and Spoke”)

[Figure 6-29](#) shows an example of these two basic L2 access topologies.

Figure 6-29 L2 Access Topologies

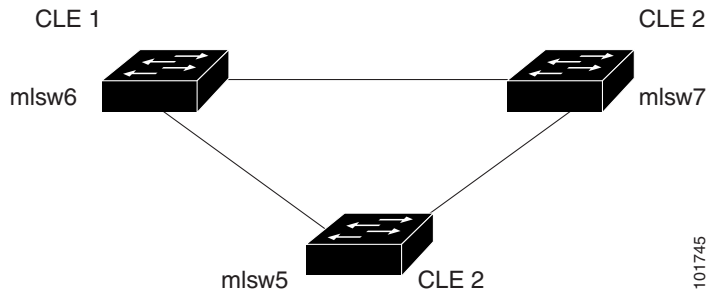


## Creating Ring of Three PE-CLEs

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 6-30 shows an example ring of three Catalyst 3550 switches: mlsw5, mlsw6, and mlsw7.

Figure 6-30 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 6-31](#) 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.

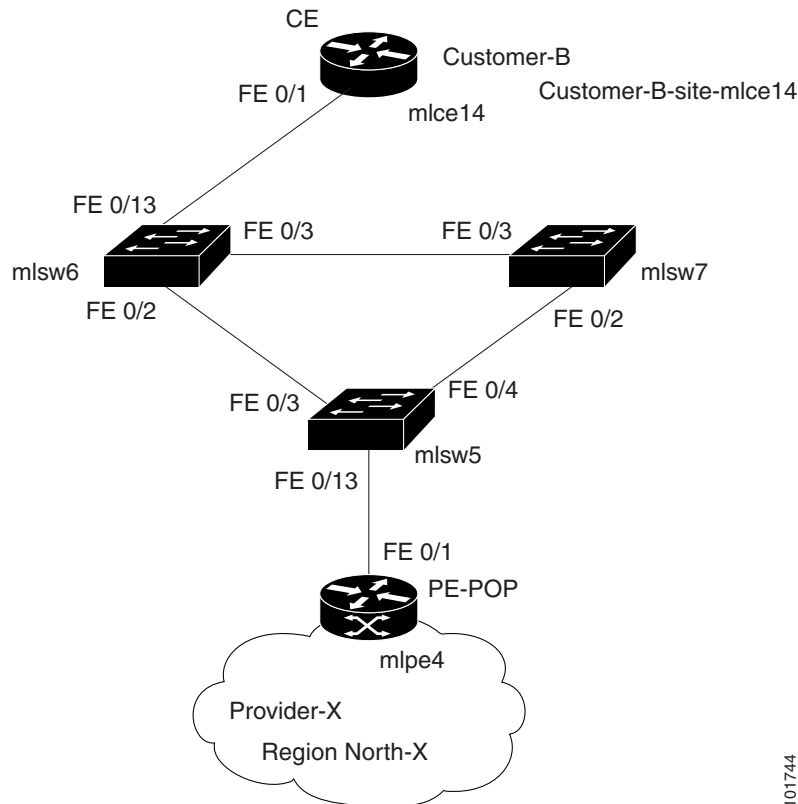
Proceed to [Configuring NPC Ring Topology, page 6-157](#).

---

## Configuring NPC Ring Topology

[Figure 6-31](#) shows an example of the Ring Topology (three CLE) inserted between a CE (**mlce14**) and a PE-POP (**mlpe4**).

Figure 6-31 The Ring Topology



101744

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.
- 

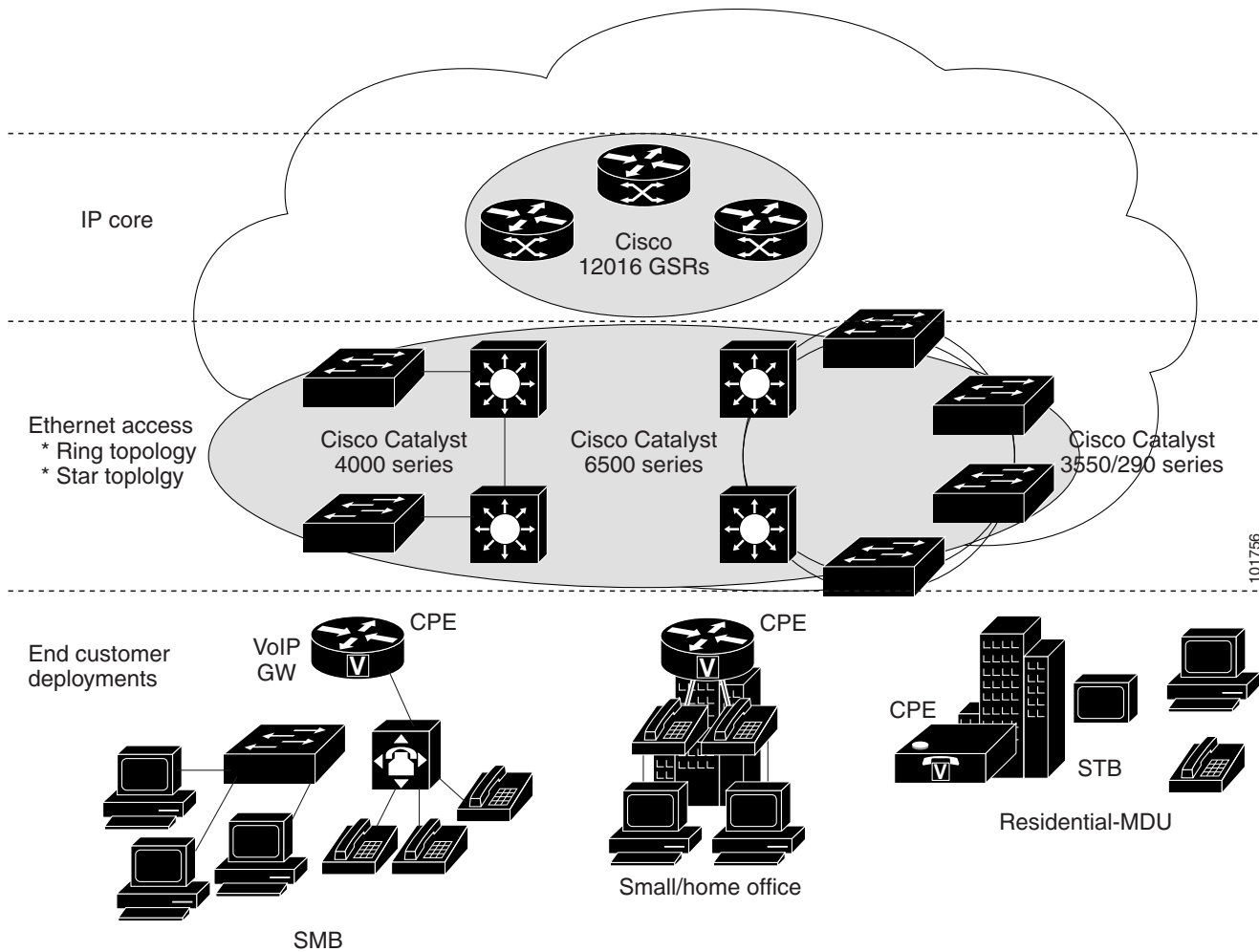
## Ethernet-To-The-Home (ETTH)

This section describes how to configure Ethernet-To-The-Home (ETTH) using the Prime Provisioning 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 Provisioning with the L2VPN Metro Ethernet service feature. Unlike ETTB, whose customers are mainly business customers, ETTH is targeted at residential customers.

[Figure 6-32](#) shows an overview of the Cisco ETTx solution.

Figure 6-32 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 Provisioning, 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 Provisioning provides the ability to create MAC access lists by specifying the MAC addresses to be allowed or blocked.

## Prime Provisioning ETTH Implementation

The Prime Provisioning MPLS VPN implementation of ETTH consists of the following three subfeatures:

- [PVLAN or Protected Port, page 6-161](#)
- [Access Port, page 6-161](#)
- [IGMP with MVR, page 6-161](#)

### 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 Provisioning, 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 Provisioning to decide when it is used. Prime Provisioning only makes it available and the network application running above Prime Provisioning 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**.  
The Policy Manager window appears.

- 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**.



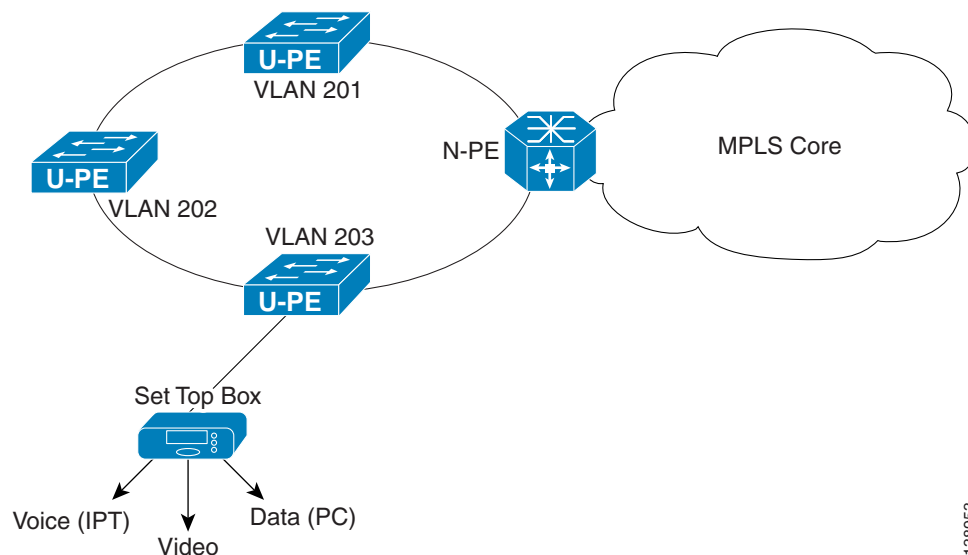
**Note** For more information on setting the VRF and VPN attributes in MPLS VPN service requests, see [Defining VRF and VPN Attributes in an MPLS Service Request](#), page 6-91.

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 6-33](#).

**Figure 6-33 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 > Service Request Manager**.  
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 6-49](#).

---

- 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 6-91](#).

---

## 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 > Policy Manager > 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 interface picker, 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 6-49](#).

---

**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 6-14](#). 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 6-91](#).

---

**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 11, “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 Provisioning 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 6-167](#) 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.

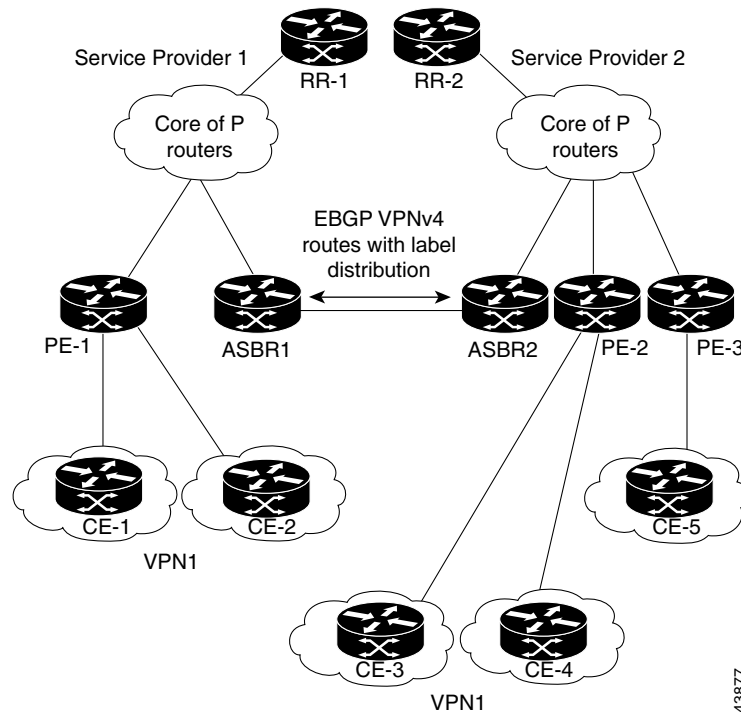
- Allows confederations to optimize iBGP meshing

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 6-34 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 6-34 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 6-35 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 *RDI: 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 6-35 Exchanging Routes and Labels Between Two Autonomous Systems

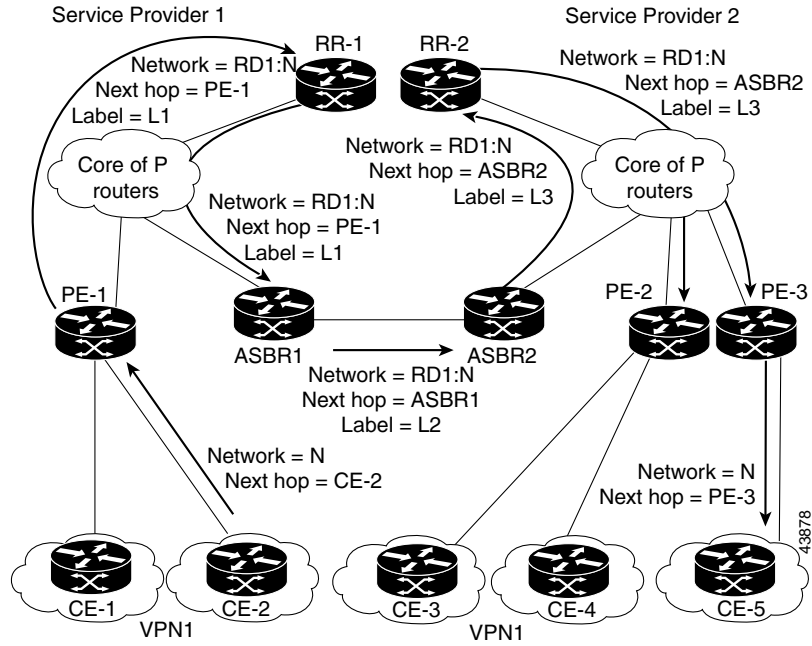


Figure 6-36 illustrates the exchange of VPN route and label information between autonomous systems. The only difference is that ASBR2 is configured with the **redistribute connected** command, which propagates the host routes to all PEs. The **redistribute connected** command is necessary because ASBR2 is not the configured to change the next hop address.

Figure 6-36 Host Routes Propagated to All PEs Between Two Autonomous Systems

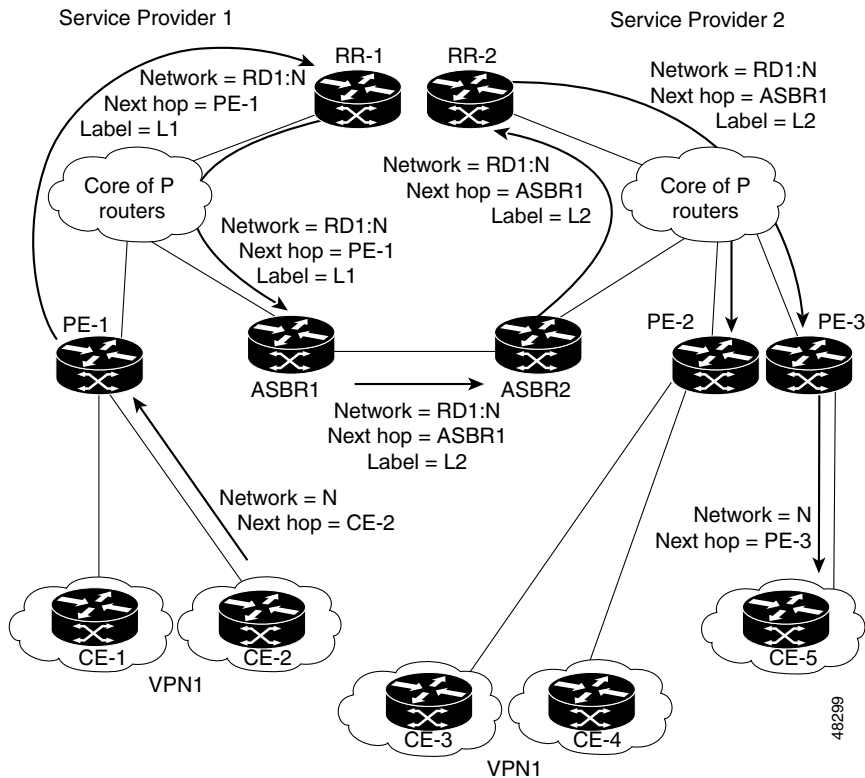


Figure 6-37 illustrates how packets are forwarded between autonomous systems in an interprovider network using the following packet forwarding method:

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 6-37 Forwarding Packets Between Two Autonomous Systems

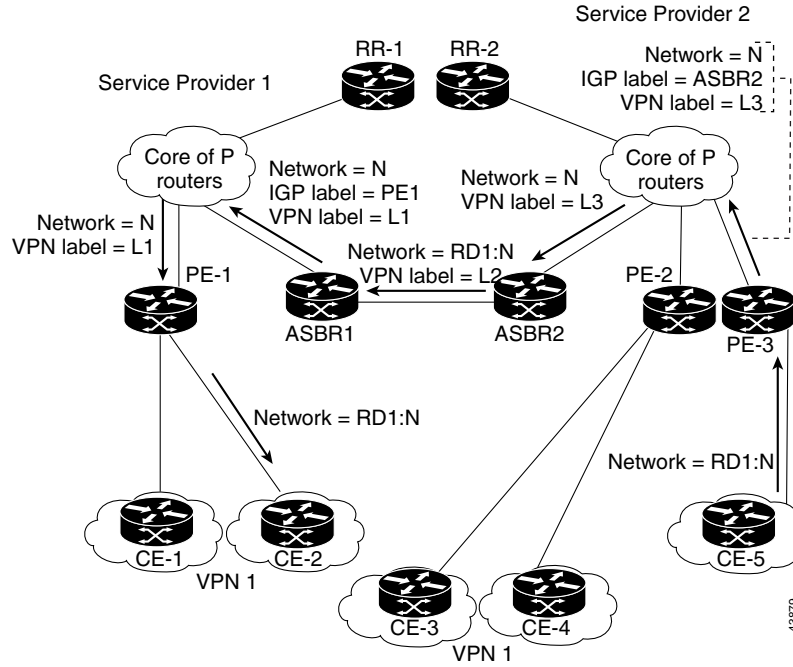
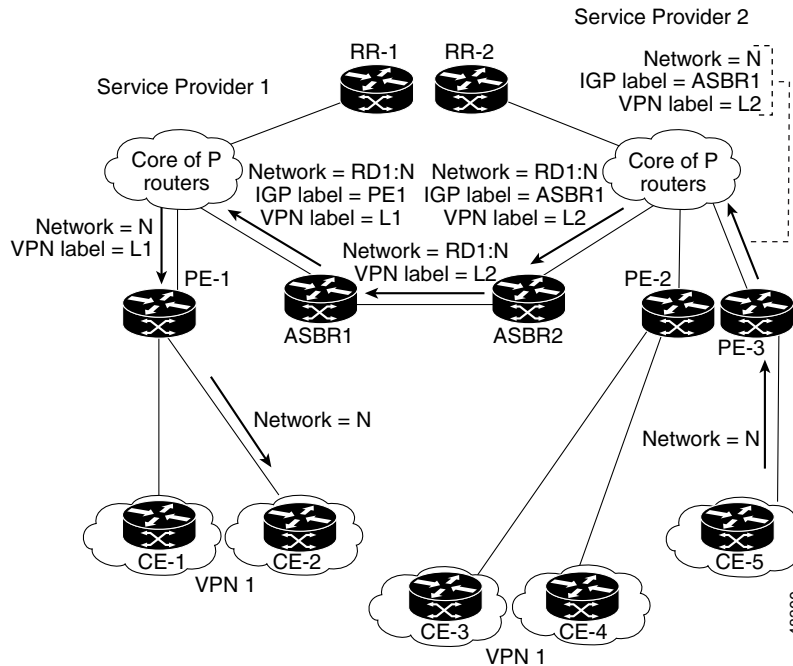


Figure 6-38 illustrates shows the same packet forwarding method, except the eBGP router (ASBR1) forwards the packet without reassigning it a new label.

Figure 6-38 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.

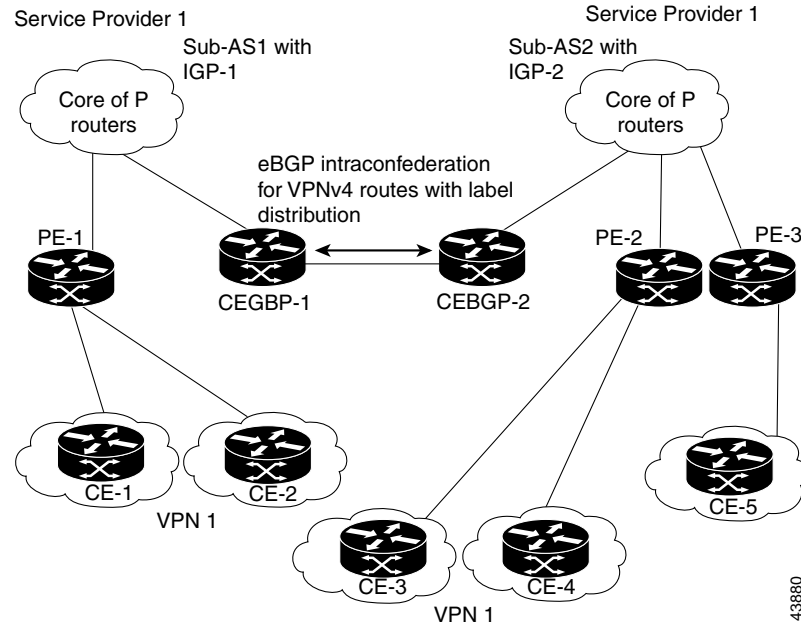
In a confederation, each subautonomous system is fully meshed with other subautonomous systems. The subautonomous systems communicate using an IGP, such as Open Shortest Path First (OSPF) or Intermediate System-to-Intermediate System (IS-IS). Each subautonomous system also has an eBGP connection to the 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 6-39 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 6-39** 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.

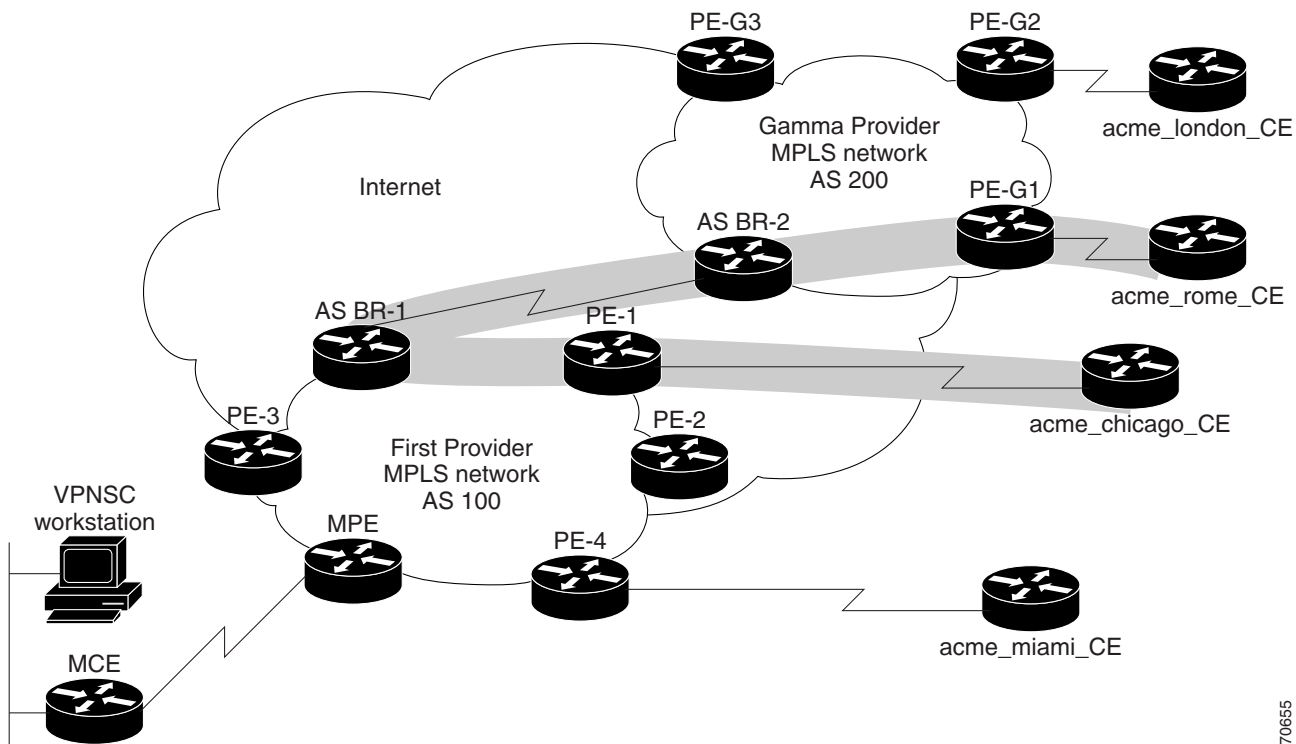
## Using Prime Provisioning to Span Multiple Autonomous Systems

As described in [Exchanging VPN Routing Information, page 6-168](#), 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 6-40](#) shows the example Prime Provisioning network used in this section.

Figure 6-40 Example VPN Network with Two Autonomous Systems



70655

In order for traffic from Acme\_Chicago in AS 100 to reach Acme\_Rome in AS 200, Prime Provisioning 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 6-40](#), Prime Provisioning 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 Provisioning 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 Provisioning supports inter-autonomous system (inter-AS) and inter-provider VPNs through Prime Provisioning templates.



**Note** Prime Provisioning 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 Provisioning 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 vpv4** 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 Provisioning 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 Provisioning, see [Chapter 11, “Managing Templates and Data Files.”](#)

The default inter-AS templates are provided in the Examples templates directory in Prime Provisioning. 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 Provisioning 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 Provisioning 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 Provisioning 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 Provisioning. It contains the following sections:

- [Overview, page 6-177](#)
- [L2 Access into L3 MPLS VPN, page 6-179](#)
- [CE-PE L3 MPLS VPN \(BGP with full-mesh\), page 6-181](#)
- [CE-PE L3 MPLS VPN \(BGP with SOO\), page 6-182](#)
- [CE-PE L3 MPLS VPN, page 6-184](#)
- [PE L3 MPLS VPN \(Dual-stack, Static \[IPv4\], BGP \[IPv6\], IOS\), page 6-185](#)
- [CE-PE L3 MPLS VPN \(Q-in-Q/Second VLAN ID, IOS\), page 6-187](#)
- [CE-PE L3 MPLS VPN \(Q-in-Q/Second VLAN ID, IOS XR\), page 6-189](#)
- [PE L3 MPLS VPN \(with Multicast, IPv4 and IPv6 Enabled VPN, IOS XR\), page 6-191](#)
- [PE L3 MPLS VPN \(Static, IOS, IPv6\), page 6-193](#)
- [PE L3 MPLS VPN \(BGP, IOS\), page 6-195](#)
- [PE L3 MPLS VPN \(BGP, IOS, IPv6\), page 6-196](#)
- [PE L3 MPLS VPN \(BGP, IOS XR\), page 6-197](#)
- [PE L3 MPLS VPN \(BGP, RD Format, IOS XR\), page 6-198](#)
- [PE L3 MPLS VPN \(BGP, Maximum Prefix/Restart, IOS XR\), page 6-199](#)
- [PE L3 MPLS VPN \(BGP, Default Information Originate, IOS XR\), page 6-201](#)
- [PE L3 MPLS VPN \(OSPF, IOS\), page 6-203](#)
- [PE L3 MPLS VPN \(OSPF, IOS XR\), page 6-204](#)
- [L3 MPLS VPN \(OSPF, Default Information Originate, IOS XR\), page 6-205](#)
- [PE L3 MPLS VPN \(EIGRP, Authentication Keychain Name, IOS XR\), page 6-207](#)
- [PE L3 MPLS VPN \(Independent VRF, IOS XR\), page 6-209](#)
- [PE L3 MPLS VPN \(Independent RTs for IPv4 and IPv6, IOS XR\), page 6-211](#)
- [PE L3 MPLS VPN \(Bundle-Ether Interface, IOS XR\), page 6-213](#)
- [PE L3 MPLS VPN \(Outgoing Interface + Next Hop IP Address, Static Route Configuration, IOS XR and IOS\), page 6-214](#)

## Overview

The configlets provided in this section show the CLIs generated by Prime Provisioning 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 Provisioning 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**

---

If multiple SRs are deployed on same device comprised of shared VRF commands, during decommission of such SRs, only the last SR will remove the configuration related to shared VRF commands from the same device.

---

**Note**

---

All examples in this appendix assume an MPLS core.

---

For information on how to view configlets, see [Viewing Service Request Configlets, page 10-5](#).

## 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                                                                                                                                                                                                                                                                                             | U-PE                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                        | N-PE                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                               |
|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <pre> ! 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 </pre> | <pre> ! 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 </pre> | <pre> ! 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 </pre> |

---

**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                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                            |
|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <pre> ! vlan 62 exit ! interface FastEthernet0/13 no ip address switchport switchport trunk encapsulation dot1q switchport mode trunk switchport trunk allowed vlan 62 ! interface Vlan62 description By VPNSC: Job Id# = 29 ip address 10.19.19.42 255.255.255.252 no shutdown ! router bgp 10 neighbor 10.19.19.41 remote-as 100           </pre> | <pre> ! ip vrf V9:mpls_vpn1 rd 100:1506 route-target import 99:3204 route-target export 99:3204 maximum routes 100 80 ! interface FastEthernet2/5.62 description FastEthernet2/5.62 dot1q vlan id=62. By VPNSC: Job Id# = 29 encapsulation dot1Q 62 ip vrf forwarding V9:mpls_vpn1 ip address 10.19.19.41 255.255.255.252 no shutdown ! router bgp 100 address-family ipv4 vrf V9:mpls_vpn1 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           </pre> |

### 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 S00)

### 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 Provisioning.  
Interface(s): FE1/0/14.
  - Routing protocol = BGP.
  - VPN = hub.

### Configlets

| CE                                                                                                                                                                                                                                                                                                                                                                   | PE                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                            |
|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <pre> ! 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 </pre> | <pre> ! 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 </pre> |

### 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                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                |
|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <pre> ! 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           </pre> | <pre> ! 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           </pre> |

### 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 Provisioning).
- 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**.

## 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                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                |
|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <pre>! interface FastEthernet0/0.158 description FastEthernet0/0.158 dot1q vlan id=158. By VPNSC: Job Id# = 239 encapsulation dot1q 158 ip address 10.1.1.98 255.255.255.252 no shutdown ! ip route 0.0.0.0 0.0.0.0 FastEthernet0/0.158</pre> | <pre>! ip vrf V15:MPLS-1 rd 100:6812 route-target import 100:7000 route-target import 100:7001 route-target export 100:7000 ! 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</pre> |

### 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 Provisioning 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 Provisioning, see the coverage of the Second VLAN ID attribute in the section [Creating an MPLS VPN PE-CE Service Request, page 6-85](#).

# 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
```



# 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 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_VerVel 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_VerVel 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_VerVel address-family ipv4
 rp-address 115.101.110.122 list1
 !
router pim vrf V18:VPN_VerVel address-family ipv6
 rp-address 1114::122 list2
 !
end
```

## 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.

## CE 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 CE is running IOS.
  - Interface(s): GigabitEthernet1/0/4.2894.
  - Routing protocol = STATIC.

### Configlets

#### PE

```
vrf definition V4:Oct10_Vpn333
rd 64512:36861
address-family ipv6
export map grey_mgmt_vpn_Prio_64512_V4:Oct10_Vpn333
route-target import 64512:26245
route-target import 64512:26246
route-target export 64512:26245
route-target import 64512:26251
interface GigabitEthernet1/0/4.2894
description GigabitEthernet1/0/4.2894 dot1q vlan id=2894. By VPNSC: Job Id# = 9
encapsulation dot1Q 2894
vrf forwarding V4:Oct10_Vpn333
ipv6 address 4518::758/64
no shutdown
router bgp 64512
address-family ipv6 vrf V4:Oct10_Vpn333
redistribute static
exit-address-family
route-map grey_mgmt_vpn_Prio_64512_V4:Oct10_Vpn333 permit 20
match ipv6 address V4:Oct10_Vpn333_V6_ACL
set extcommunity rt 64512:26252 additive
ipv6 access-list V4:Oct10_Vpn333_V6_ACL
permit ipv6 4518::/64 any
```

### 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 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
```

## 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 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
 !
!
```



# 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 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
```

# 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 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
 !
!
!
```

## 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 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
```

# 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 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
```



# 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 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
```

# 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 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
```

#### 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
```

# 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 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 string **-etc** appended, as shown below.

```
vrf V872:vpn2-etc
address-family ipv4 unicast
```

```
import route-target
64512:1005
!
export route-target
64512:1005
!
!
```

## 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 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
!
```

# 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 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 GigabitEthernet0/3/0/0 10.12.16.19
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 GigabitEthernet0/3/0/2 10.14.54.18 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
!
```

## 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 6-216](#) 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 6-214](#). 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/mps`.

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 Prime Provisioning data. The Prime Provisioning 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 Provisioning servers being stopped or disabled.

To check the status of all Prime Provisioning servers, perform the following steps:

- 
- Step 1** Open the Host Configuration dialog by going to **Administration > Control Center > Hosts**.  
The 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 6-41](#).

**Figure 6-41** Prime Provisioning Server Status

The screenshot shows the 'Servers' page in the Prime Provisioning interface. It features a table with 7 columns: #, Name, State, Generation, Start Time, Successful Heartbeats, and Missed Heartbeats. There are 5 rows of data. Below the table is a pagination control showing 'Page 1 of 1' and a 'Rows per page' dropdown set to 10. At the bottom right, there are buttons for 'Start', 'Stop', 'Restart', 'Logs', and 'OK'. A vertical ID number '238851' is visible on the right edge of the screenshot.

| # | Name                              | State    | Generation | Start Time             | Successful Heartbeats | Missed Heartbeats |
|---|-----------------------------------|----------|------------|------------------------|-----------------------|-------------------|
| 1 | <input type="checkbox"/> nspoller | started  | 1          | Nov 21 07:37:07 AM EST | 690                   | 0                 |
| 2 | <input type="checkbox"/> dbpoller | started  | 1          | Nov 21 07:37:07 AM EST | 682                   | 0                 |
| 3 | <input type="checkbox"/> httpd    | started  | 1          | Nov 21 07:37:12 AM EST | 685                   | 0                 |
| 4 | <input type="checkbox"/> rgserver | disabled | 11         | Nov 21 08:00:25 AM EST | 0                     | 0                 |
| 5 | <input type="checkbox"/> cnserver | started  | 1          | Nov 21 07:37:12 AM EST | 690                   | 0                 |

- Step 4** On the Prime Provisioning 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 Provisioning 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 6-214](#) 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 > Hosts**.  
The 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 Provisioning 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 Provisioning 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?

Prime Provisioning 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 Provisioning.

## What is the intent of the Allocate New Route Distinguisher check box in the MPLS policy?

There were some behavior changes implemented in Prime Provisioning 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 Provisioning 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 Provisioning. 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 Provisioning.



### Note

While saving a MPLS service request, you can specify new values to the overwrite attributes VRF name and RD value. When you deploy the SR, the VRF and RD overwrite values gets correlated. So, if you want to modify or use the existing attribute values both VRF name and the corresponding RD value has to be modified or copied accordingly. For example, consider that you have deployed a service request SR1 with the overwrite attribute values as VRF1 and RD1. For the modification to happen successfully, you have to modify both VRF1 and RD1 as they are correlated.

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 Provisioning 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 Provisioning 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 Provisioning finds the first matching VRF configured across the network. If the PE does not have this VRF, Prime Provisioning 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.
6. While creating an MPLS service request, if the selected device is IOS and the RD value is “auto”, then an error message is displayed. If the selected device is ASR9k and the RD value is “auto”, then you can save the MPLS service request without an error.

## 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 FastEthernet0/15
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 Provisioning? The only way is to do it manually, without using auto pick. Prime Provisioning 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 Provisioning 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 Provisioning does not have any support for different pools to be used within the same PE.

Prime Provisioning allows you to create multiple pools, but you can only use one based on the provider region. Prime Provisioning 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 Provisioning 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 deleted.

## Why doesn't Prime Provisioning remove some of the router BGP/EIGRP commands when a service request is decommissioned?

Prime Provisioning 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 Provisioning. This is particularly applicable when the network statement was added outside of Prime Provisioning. Prime Provisioning does not remove the redistribution from other routing protocols under EIGRP because the redistribute command might not be created specifically for the link.

Prime Provisioning 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 Provisioning 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 Provisioning 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 6-224](#)
- [Editing VRFs, page 6-226](#)

## Creating a VRF

After you create a VRF object, you can provision it using a VRF service request, as explained in the [Cisco Prime Provisioning 7.2 User Guide](#).

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:
- Name** (required)—Enter the name of the VRF, any name of your choice. This name is directly deployed on the PE device.
  - Provider** (required)—To select the provider associated with this VRF, choose **Select**.
  - From the list of providers, select the appropriate provider, and then click **Select**.
  - Description** (optional)—Enter a description, if you choose.
  - Route Targets** (required)—Click the **Select** button.
  - From the list of Route Targets, choose only one appropriate Route Target, and then click **Select**.
  - 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.
  - Export RT List**—Enter one or more Route Targets (RTs) to be exported from the VRF. For multiple RTs, separate the RTs by commas.
  - Import Route Map**—Enter the name of a route map defined on the device. Prime Provisioning 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 Provisioning 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 Provisioning 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 6-224](#). See the [“Creating a VRF” section on page 6-224](#) 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 6-224](#) 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.
-





# Managing MPLS Transport Profile Services

---

This chapter describes the tasks required to get started using Prime Provisioning, Multiprotocol Label Switching (MPLS) Transport Profile (TP) services.

This section covers the following topics:

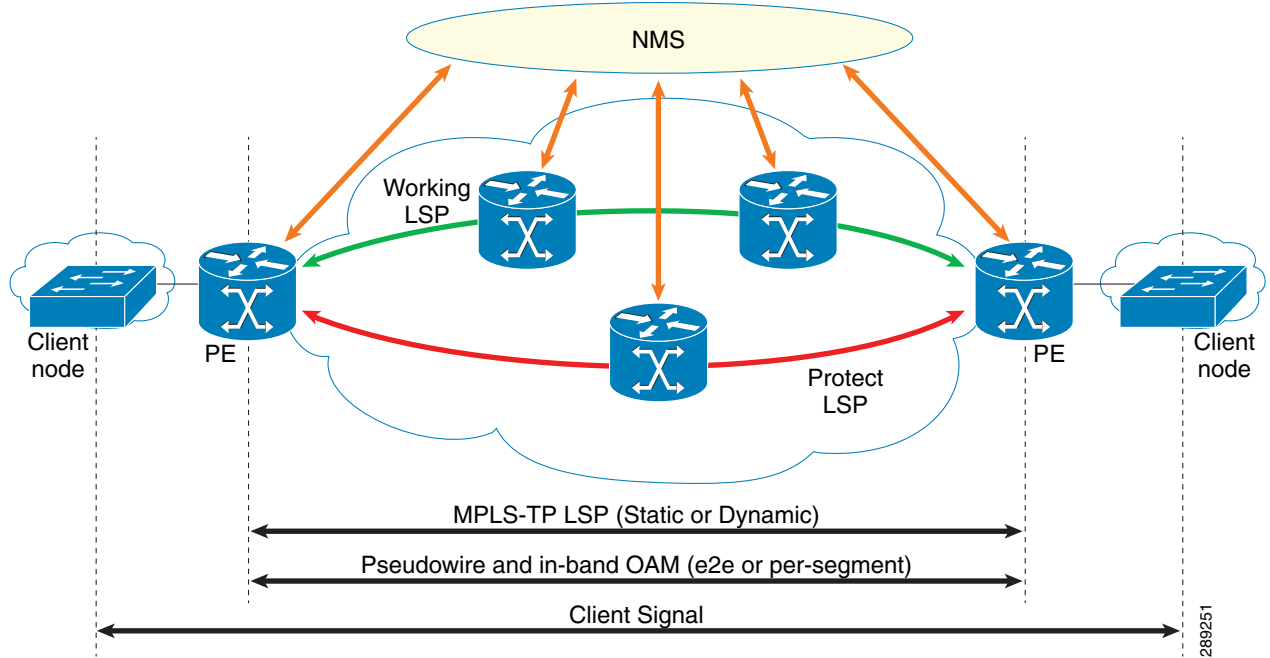
- [Introduction, page 7-1](#)
- [Prerequisites and Limitations, page 7-2](#)
- [Preconfiguration Process, page 7-2](#)
- [Running MPLS-TP Discovery, page 7-5](#)
- [Creating an MPLS-TP Policy, page 7-7](#)
- [Creating an MPLS-TP Service Request, page 7-9](#)
- [Deploying an MPLS-TP Tunnel, page 7-14](#)
- [Sample Configlets, page 7-15](#)

## Introduction

MPLS-TP is a transport service (managed by Prime Provisioning) for a dynamic MPLS core.

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 Provisioning 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 7-1 An MPLS-TP Enabled Network



## Prerequisites and Limitations

The current release of Prime Provisioning involves certain prerequisites and limitations, which are described in the [Cisco Prime Provisioning 7.2 Installation Guide](#), 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 7-9](#)) 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. IE9 (and other Prime Provisioning supported browsers) shows the calculated path graphically.

Changes performed to an operational device sometimes take time to reflect on Prime Network.

Polling is performed by Prime Network every 15 minutes (at least). In the duration of 1 to 15 minutes, polling is performed many times. Each poll collects different data (tunnels, labels, links, etc). Since all the information is not collected in a single poll, the time taken to reflect tunnel update, label update, links update varies in Prime Network.

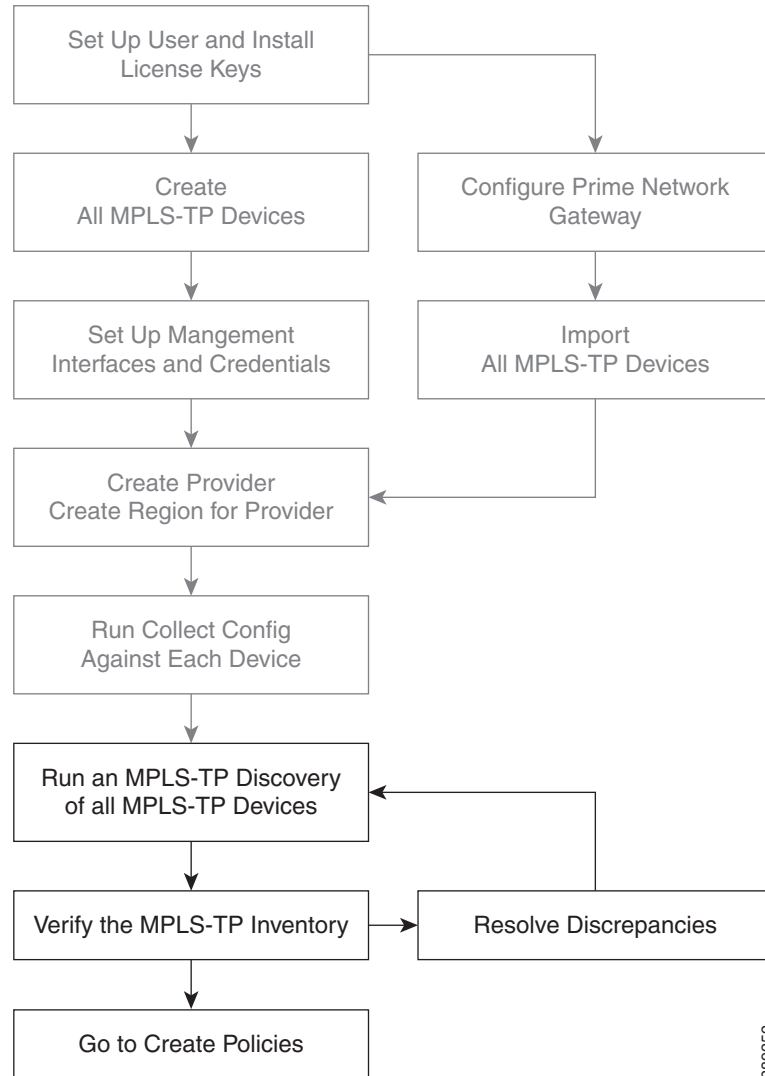
For supported device and OS information, refer to [Cisco Prime Provisioning 7.2 Supported Devices](#).

## 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 7-2](#).



**Figure 7-2** Preconfiguration Process

289/252

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 7-4](#).

Setting up new user and installing license keys is described in [Cisco Prime Provisioning Administrator's Guide 7.2](#) and the other steps are covered in [Setting Up Devices and Device Groups, page 2-1](#).

As a result, the Prime Provisioning user will need to wait some before running MPLS-TP Discovery after a device change in Prime Network.

**Note**

When Prime Provisioning is integrated with Prime Network, it is required to import Prime Network certificate into Prime Provisioning Trust Store, which is described in [Device Import Prerequisite, page 13-12](#).

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 7-5](#).)
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 7-7](#).)

## MPLS-TP Setup and Installation

Before setting up Prime Provisioning, the Prime Provisioning software must be installed. To do so, see the [Cisco Prime Provisioning 7.2 Installation Guide](#).

To set up a new Prime Provisioning user, one or more users with a MPLS-TP role must be created. MPLS-TP roles are described in [MPLS-TP User Roles, page 7-4](#) and for step by step instructions for creating user roles, refer to [Cisco Prime Provisioning 7.2 Administration Guide](#).

For more information about Prime Provisioning licenses and for the procedure required to install licenses, refer to [Cisco Prime Provisioning 7.2 Administration Guide](#).

## MPLS-TP User Roles

Prime Provisioning currently supports two MPLS-TP roles, the MPLS-TPRole and MPLS-TPServiceOpRole. These two user roles behave similarly to the other roles in Prime Provisioning, 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, refer to [Cisco Prime Provisioning 7.2 Administration Guide](#).

## 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 Router 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.
- Step 4** Create a BFD class to be used to monitor your MPLS-TP tunnels.
- 

## Running MPLS-TP Discovery

Prime Provisioning supports MPLS-TP discovery from IOS & IOS-XR devices when deployed together with Prime Network(s) (or) in a Prime suite.

As a prerequisite for running MPLS-TP discovery, all devices must be present and a Collect Config task must be run (see [Collect Config from Files, page 12-4](#)).

Prime Provisioning should be 'paired' with the Prime Network(s) by setting the gateway details of the Prime Network(s) in the Prime Provisioning DCPL properties **Inventory Import**. Multiple Gateways can be configured by separating the values with a comma. The order in which the Prime Networks are configured has an impact on the MPLS-TP discovery process. The instance mentioned first in the DCPL has the highest priority whereas the last has the least priority. For further details on setting the DCPL properties, refer to [Cisco Prime Provisioning 7.2 Administration Guide](#), or see [Appendix G, "Property Settings"](#)



**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

When MPLS-TP discovery process runs, Prime Provisioning checks if the chosen device(s) are present in the Prime Network instances. If the device chosen for MPLS-TP discovery is present in multiple Prime Network instances, MPLS-TP discovery is performed based on the Prime Network instance priority. MPLS-TP information is collected from the highest priority, whereas the TP enabled links alone are collected from all the Prime Network instances.



**Caution**

If an MPLS-TP network spans devices that are in multiple gateways, the link will not be discovered between devices that are in different gateways. To ensure a full network discovery, add border devices to both the Prime Network gateways, so that all TP links are visible at least in one Prime Network instance.

---

If the device chosen for MPLS-TP discovery is not present in the highest priority Prime Network, Prime Provisioning checks the next priority Prime Network instance 'paired' in the DCPL to discover the MPLS-TP information.

If the device chosen for MPLS-TP discovery is not present in any of the Prime Network instance, an error message is logged, "MPLS-TP Discovery for device "DeviceName" failed. Device not found in any of the gateways."

If the device chosen for MPLS-TP discovery has any directly connected neighbors and available in the Prime Provisioning inventory, MPLS-TP discovery is performed for the neighbors from the same Prime Network instance.

Prime Provisioning, in standalone mode (without Prime Network integration) supports CDP-based MPLS-TP discovery from IOS devices but this is deprecated.

MPLS-TP enabled devices should be added or created on Prime Provisioning Inventory by:

- Directly creating the devices on Prime Provisioning (or)
- Using the “Import” functionality available in the Prime Provisioning device creation page - where the device can be imported from Prime Network.

The MPLS-TP network is discovered using the **MPLS-TP Discovery** task. This populates the repository with the network topology in an automated way. 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.

The necessary steps are described in the below sections:

- [Creating an MPLS-TP Discovery Task, page 7-6](#)
- [Verifying the MPLS-TP Discovery Results, page 7-7](#)

## 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**.  
To run the MPLS-TP Discovery task immediately after the device creation navigate to:  
**Inventory > Devices > Create > Cisco Device**.  
Check the MPLS-TP check box in Create Cisco Router window.

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 occurred in the MPLS-TP network. Discovery updates the logs with affected SR's in cases where the links in working or protect LSP no longer exist or have been changed. This could be as a result of node insertion/removal or simply changing a link number.

To view the log, select the relevant task in Task Manager and click **Logs**.

### Verifying Links, Pools, and MPLS-TP Global and Router IDs

You can verify the status of links and pools by navigating 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**.

## MPLS Label Sync

MPLS Label Sync task is to update the labels information. MPLS Labels can be out of sync due to manual provisioning. Hence, it is recommended to update the label information alone rather than the entire MPLS topology information often.

Similar to MPLS Discovery, MPLS Label Sync task can be performed from:

- Task Manager window
- Device Inventory window
- Device Creation window

MPLS-TP Labels sync task can also be done using this process.

## 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** picker 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, ensure that the **Protection** box is selected so that Prime Provisioning auto generates 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 the available list of BFD templates on the source and destination devices respectively or you can enter the BFD template name in the field irrespective of device or device type. A valid BFD template name is max. 31 characters long.



**Note**

On IOS devices BFD timers are specified via a template, while on IOS-XR they are either specified globally and can be overridden in each individual TP tunnel. You can create a policy that works on IOS and IOS-XR, by defining the template and optionally by specifying timer values that would override the global BFD values in IOS-XR. During policy creation you are just providing default values, validation will be performed once you have actual values, when creating the individual tunnel service requests.

For an explanation of global ID and router ID, see [Global ID and Router ID, page 7-9](#).

**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 Provisioning 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 MPLS-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 enabled, 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 7-7](#).

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 policy picker.

**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 7-7](#) 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.

If the source device type is IOS, the BFD template details are mandatory or if the source device type is IOS-XR and the global template was not defined in the device, then you have to provide Source BFD min-interval and Source BFD Multiplier details.

Based on the chosen device type, the BFD template or the BFD attributes will be disabled accordingly as follows:

- If the device type is IOS, BFD attribute fields, such as BFD min-interval, BFD min-interval Standby, BFD Multiplier will be disabled.
- If the device type is IOS-XR, BFD template picker will be disabled.

If you have selected IOS/ IOS-XR device and entered the BFD template or the BFD attributes respectively, and then re-select IOS-XR/ IOS device respectively, the BFD template/ BFD attributes will be disabled without losing the values entered in the respective fields. So, if you try to re-select the device again, the BFD template/ BFD attributes will be enabled with the already entered values.




---

**Note** Validation of the BFD template against the device will be performed only for the IOS device as BFD template will be disabled for IOS-XR.

---

**Step 6** In the Review Routing accordion, a default path is calculated and displayed automatically between source and destination.

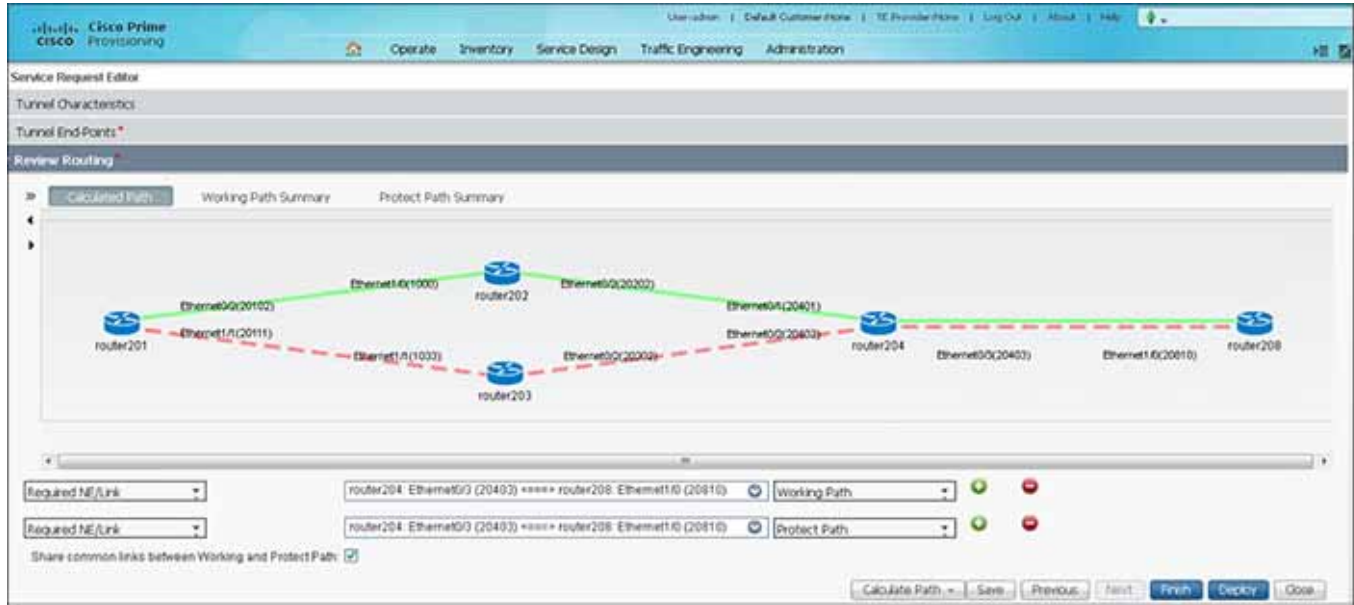
Working path—Green solid line

Protect Link—Red dotted line

For an example of an MPLS-TP routing diagram, see [Figure 7-3](#).



Figure 7-3 MPLS-TP Routing Diagram



289253

- **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.
- **Share common links between Working and Protect path**—Check this box to enable the common links in both Working and Protect Path.
  - To make use of common links or node in both Working and Protect Path add the common link or node as a **Required NE/Link** constraint in both working path and protect path.

For more information about path constraints, see [Working with Path Constraints, page 7-12](#).

**Step 7** Click **Calculate Path** to calculate the path.



**Note**

In the case of Service Request modification the **Calculate Path** button will have the following options:

- **Working LSP**—Select this option to calculate and view the working path in the path diagram.
- **Protect LSP**—Select this option to calculate and view the protected path in the path diagram.
- **Both LSPs**—Select this option to calculate and view both the working and protected paths in the path diagram.

- Step 8** Go back over the various accordions to check and edit as necessary.
- Step 9** Click **Finish** on the last accordion to complete the create service request operation.  
The Service Request Manager window opens.
- Step 10** Click Deploy to deploy the service request at the time of creation itself.

**Note**

During path calculation, the available and reserved bandwidth are not considered. So when a tunnel makes bandwidth reservation, the path with insufficient bandwidth could be considered. Sufficient bandwidth check occurs during 'Finish' or 'Deploy' operation. When there is insufficient bandwidth, the SR displays an error message with the corrective measures to be taken by the user. To make the path calculation pick a different path, use path constraints.

For information about the Service Request Manager elements and operations, see [Chapter 10, “Managing Service Requests.”](#)

Guidelines for working with path constraints are provided in [Working with Path Constraints, page 7-12](#).

A service request in the DRAFT state has not passed all validation and cannot be deployed. 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 10, “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**.

## Managing MPLS-TP Topology Changes

When a topology changes due to node insertion/removal, the MPLS-TP discovery has the ability to:

- Manage MPLS-TP topology change due to node insertion/removal.
- Identify MPLS-TP Tunnel SRs that has been impacted by node insertion/removal.
- Modify the impacted SRs to repair the MPLS-TP tunnels.
- Detect the MPLS-TP tunnel SRs that are affected by node insertion/removal.
- Re-calculate the path for affected SRs. During the recalculation:
  - Affected LSP is locked by Prime Network for uninterrupted traffic.

- All affected SRs in Prime Provisioning are re-routed except those that are in Closed, Pending or In-Progress state or DELETE Op Type.
- Transition the affected SR into appropriate state.
  - Transition occurs only for deployed SRs.
  - If a new route is found for the broken tunnel for deployed SRs, the SR moves to Requested state.
  - A deployed tunnel SR moves into invalid state when no new route is found.
  - For all other SRs, except Closed, Pending or In-Progress State, and Op Type DELETE, the path is re-calculated without any state change.
- Report the affected SRs and update the SR logs. Discovery updates the logs with affected SRs in cases where the links in working or protect LSP no longer exist or have been changed.
  - For all the affected SRs, discovery updates the discovery logs and SR history report.
  - Discovery updates the SR history with:
    - Affected path, working/protect LSP.
    - State change details, previous/current state.
    - Messages related to path change/failure.
- Re-provision only the affected LSP. When the SR in Requested Modify state is selected to be deployed, only the LSP which has changed is re-provisioned by Prime Provisioning. This ensures that the traffic on the active LSP is uninterrupted.

## 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. Once the SR is successfully deployed, the bandwidth allocated by the SR is subtracted from the available bandwidth of each TP link used by the TP tunnel.



### Note

During bulk deployment, the bandwidth of the SR is reduced from the available bandwidth of the TP link for each successful deployment. Eventually, the available bandwidth is reduced to zero or less than the bandwidth requested by the consecutive SR. Such service request must be transitioned to invalid state due to insufficient bandwidth. Also, when the device does not have the enough bandwidth as requested by the SR, an insufficient bandwidth error message is displayed.



### Tip

A service request in **DRAFT** state cannot be deployed.

The deploy functionality is the same as for other Prime Provisioning services. For instructions on how to deploy an MPLS-TP service request, see [Deploying Service Requests, page 10-9](#).

## 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.



Note

---

Once the SR is successfully decommissioned and moved to CLOSED state, the bandwidth allocated to TP tunnel is added back to the available bandwidth of each TP link used by the tunnel.

---

To decommission one or more service requests, see [Chapter 10, “Managing Service Requests.”](#)

## Sample Configlets

The configlets included in this section show the CLIs generated by Prime Provisioning 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 Provisioning 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 Prime Provisioning.

It includes the following section:

- [MPLS-TP Working Tunnel Configlet \(IOS-XR\), page 7-17](#)
- [MPLS-TP Working Tunnel Configlet \(IOS-XR\), page 7-17](#)

## 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                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  |
|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <pre> <b>Endpoint Config</b> 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  <b>Midpoint Config</b> 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  <b>EndPoint Config</b> 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 </pre> | <p>Create an MPLS-TP working tunnel with endpoint and midpoint nodes. This involves configuring the settings on each node in the tunnel.</p> <p>Create an MPLS-TP working tunnel with the following attributes:</p> <p>Endpoint 1:</p> <ul style="list-style-type: none"> <li>- tp tunnel name: test</li> <li>- Source: 3.3.3.3</li> <li>- Destination 1.1.1.1</li> <li>- Bandwidth 100 kbps</li> <li>- bfd BFDTemplate-SingleHopMicrosec-1</li> <li>- Working LSP configuration</li> <li>- Protect LSP configuration</li> </ul> <p>Midpoint:</p> <ul style="list-style-type: none"> <li>- Source: 3.3.3.3</li> <li>- Destination 1.1.1.1</li> <li>- Bandwidth 100 kbps</li> <li>- Forward LSP configuration</li> <li>- Reverse LSP configuration</li> </ul> <p>Endpoint 2:</p> <ul style="list-style-type: none"> <li>- tp tunnel name: test</li> <li>- Source: 1.1.1.1</li> <li>- Destination 3.3.3.3</li> <li>- Bandwidth 100 kbps</li> <li>- bfd BFDTemplate-SingleHopMicrosec-1</li> <li>- Working LSP configuration</li> <li>- Protect LSP configuration</li> </ul> |

## MPLS-TP Working Tunnel Configlet (IOS-XR)

### Configuration

- Service: MPLS-TP Working Tunnel
- Feature: MPLS-TP configlet (IOS-XR) for configuring MPLS-TP enabled nodes.

### Configlets

| IOS-XR Device Configuration                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                              | Comments                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                    |
|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <pre> <b>Endpoint Config</b> interface tunnel-tp0   description PrimeF:JobID:2 (testTunnel)   source 3.3.3.3   destination 1.1.1.1 global-id 8 tunnel-id 1   working-lsp     in-label 36     out-label 23 out-link 12     lsp-number 0   protect-lsp     in-label 37     out-label 33 out-link 100     lsp-number 1   bfd     min-interval 50     min-interval standby 50     multiplier 3  <b>Midpoint Config</b> mpls traffic-eng tp   mid 3.3.3.3_1_protect_3.3.3.4_0     source 3.3.3.3 tunnel-id 1 global-id 8     destination 1.1.1.1 tunnel-id 0   global-id 80     forward-lsp       in-label 32 out-label 37 out-link 100     reverse-lsp       in-label 33 out-label 24 out-link 10  <b>EndPoint Config</b> interface tunnel-tp1   description PrimeF:JobID:2(testTunnel)   source 1.1.1.1   destination 3.3.3.3 global-id 80 tunnel-id 0   working-lsp     in-label 23     out-label 36 out-link 4     lsp-number 0   protect-lsp     in-label 24     out-label 32 out-link 10     lsp-number 1   bfd     min-interval 50     min-interval standby 50     multiplier 3 </pre> | <p>Create an MPLS-TP working tunnel with endpoint and midpoint nodes. This involves configuring the settings on each node in the tunnel.</p> <p>Create an MPLS-TP working tunnel with the following attributes:</p> <p>Endpoint 1:</p> <ul style="list-style-type: none"> <li>- Source: 3.3.3.3</li> <li>- Destination 1.1.1.1</li> <li>- Bandwidth 100 kbps</li> <li>- Working LSP configuration</li> <li>- Protect LSP configuration</li> </ul> <p>Midpoint:</p> <ul style="list-style-type: none"> <li>- Source: 3.3.3.3</li> <li>- Destination 1.1.1.1</li> <li>- Bandwidth 100 kbps</li> <li>- Forward LSP configuration</li> <li>- Reverse LSP configuration</li> </ul> <p>Endpoint 2:</p> <ul style="list-style-type: none"> <li>- Source: 1.1.1.1</li> <li>- Destination 3.3.3.3</li> <li>- Bandwidth 100 kbps</li> <li>- Working LSP configuration</li> <li>- Protect LSP configuration</li> </ul> |







# Customizing EVC, MPLS and MPLS-TP Policies

---

This chapter describes how to add custom data fields to the user interface (UI) for a specific policy and how to embed customized command line interface (CLI) templates to EVC, MPLS and MPLS-TP policies. It contains the following sections:

- [Customizing EVC and MPLS Policies, page 8-1](#)
  - [User Interface Customizations, page 8-1](#)
  - [Command Line Interface Customizations, page 8-2](#)
  - [Importing and Exporting Customizations, page 8-4](#)
  - [Changing Customizations When a Policy is in Use, page 8-5](#)
- [Customizing MPLS-TP Policies, page 8-6](#)

## Customizing EVC and MPLS Policies

### User Interface Customizations

This section provides a detailed explanation of UI customization features. You can extend policies by adding attributes that you define directly in the policy screen. It also helps you to define new UI attributes in a separate XML file. The new attributes defined in the policy behave in a manner similar to the existing feature, but allow you to define the templates in-line.

This section contains the following topics.

- [Adding User Interface Groups to Pages, page 8-1](#)
- [Adding User Interface Attributes to Groups, page 8-2](#)

### Adding User Interface Groups to Pages

While creating EVC and MPLS policies, a new **Create UI Group** button on every page of the policy enables you to create any number of UI groups on any number of pages on the policy. The name you provide for the UI Group appears as the title of the new section. Groups are used to keep related custom fields together. You can add attributes only to UI groups you create, and not to existing groups in the policy. You can further edit, delete, and reorder attributes within the UI group.

For example, edit an L3VPN policy and navigate to the VRF/VPN screen. You can create a new UI group called **Custom VRF Data** using the **Create UI Group** button.

A new **Create Global UI** button allows you to specify global attributes which are identical across all links of a service request. These global attributes appear on the first page of the service request. Like other attributes, global attributes can also be set as editable or non-editable, and have default values assigned to them.

## Adding User Interface Attributes to Groups

Once you create a UI group, the **Create** button and the Settings icon is displayed in the title bar, enabling you to create attributes. Using the **Create** button, you can add custom fields to UI groups as attributes and specify their type. The types that you can specify are:

- String – regular expression and length bounds for validation
- Password – similar to the string attribute but masked in UI
- Integer – requires you to enter numbers and defines a range
- Hexadecimal – requires you to enter hexadecimal values
- Enumeration - drop-down list
- Check box – provides a check box
- IPv4 – IP v4 address, may define range
- IPv6 - IP v6 address, may define range
- Device – pick devices from the inventory – filter by device role
- Device Interface – pick device interfaces from the inventory

The list below contains the common fields that appear for all the UI attributes:

**Name** – refers to the value of that attribute from a CLI template. For example, if you create an attribute called *cbr*, you can refer to this new attribute in the CLI template using the variable *\$cbr*.

**Display Name** – used as the label for the attribute in the Prime Provisioning UI.

**Display Description** – displayed when you hover over the tool tip icon for that attribute.

Attributes can be marked Required or Optional. To verify whether optional values are provided, you can use *#if ( \$my\_optional\_attribute )* within a CLI template. Attributes marked Required are displayed on the policy and Service Request pages.

## Command Line Interface Customizations

You can extend the provisioning logic using **Provisioning CLI customizations section** of the Policy Editor. This uses data fields from both standard service model and UI Customization as input. The product of a standard service model is a custom configlet, which is merged with the standard configlet and sent to the device.

The new CLI templates in the policy are simpler to use, and allow you to create and use CLI customizations without the need for data files. However, when you upgrade using an existing database, it is not possible to convert existing templates into the new form of CLI templates automatically.

This section contains the following topics:

- [Creating Templates, page 8-3](#)
- [Variable Completions for Specifying CLIs, page 8-3](#)
- [Creating Rules for CLI Templates, page 8-4](#)

## Creating Templates

You can now create and customize templates that consist of the CLIs, which you want to deploy on the devices using **Create CLI Template** button. Templates can refer to the data that you enter in UI groups. When you create a template, in a policy, you can specify:

- **CLI Merging Mode:**
  - **External-** This mode acts in a manner similar to the Template Manager customizations. It is suitable for adding the configuration that you want Prime Provisioning to generate without modifying any lines in the configuration. Extra configuration is simply sent to the device as is.
  - **Combine-** This mode acts in a manner similar to the XDE/PAL customizations. It is suitable for changing the configuration that Prime Provisioning generates. The content in the template is merged with the existing configuration, and is also sent to the device only when the current device configuration does not contain the required configuration. In addition to this, the output of the template is audited so that Prime Provisioning can verify the final device configuration and check that the configuration specified in the template is present on the device. Combining depends on the ability of Prime Provisioning's config parser (NOM) to parse the configuration generated by the template. To determine whether this Combine mode can be used with a given template, you need to merely preview the configuration generated for a service request. If NOM does not recognize a line from the template, you will see an error and the line is not included in the final configuration.
  - **ExternalWithModify -** To modify the customized template attribute value, this CLI merging mode has to be selected.
- **Commission Sequence:** Determines whether the commission CLI is added before or after the configuration that Prime Provisioning generates. To ensure that Prime Provisioning sets up the basic service before it adds the features in the template, select **After**. If the merge mode you select is **Combine** and the commission sequence you select is **After**, the template can overwrite or remove the configuration that Prime Provisioning generated. Instead, if the commission sequence you select is **Before**, it will be Prime Provisioning's configuration that can overwrite that of the template.
- **Commission CLI:** The CLI generated during the commission sequence specified in the Velocity Template Language.
- **Decommission Sequence:** Determines whether the decommission CLI configuration is removed before or after the configuration that Prime Provisioning generates by default. This is the opposite of the Commission Sequence. To control the decommissioning sequence individually, you can create a separate template solely for the purpose of decommissioning.
- **Decommission CLI:** The CLI created during the decommission sequence.
- **Verify:** Click the **Verify** button after entering into CLI. It lists the missed out variable name, which is defined in the policy page but wrongly declared in the CLI section.

## Variable Completions for Specifying CLIs

Variable completions are now available while specifying CLIs in templates. This means that you can use Ctrl-Space for completion of variables that you want to enter.

For example, when you type \$ and then type Ctrl-Space, the list of all possible variables is displayed and you can select variables directly from this list without having to know them beforehand. Similarly, if you type a prefix to a variable e.g. \$SR, then a filtered list of all \$SR variables is listed. Further typing while the variable list is visible will further narrow the available options. When only a single option is available, it is selected automatically.

The displayed list of variables consists of:

- customized attributes that you define in the UI groups.
- `$$SR`. standard attributes from the service request section for template attributes. These are the same attributes (names and values) as are defined for the template manager.
- the configuration of the device in the form of an XML document as parsed by NOM is present in the variable `$DeviceConfig`.
- the definition of the service to be configured as represented in the Database is also available as an XML document in the variable `$$ServiceIntent`. This can be used if you need to get some aspect of the service which is not available in the `$$SR` prefixed variables.
- `$system.xpath (<XML>, <XPath query>)`.
- `$list.xpath (<XML>, <XPath query>)`.
- `$system.xpathreference (<XML>, <XPath query>)`.
- `$list.xpathreference (<XML>, <XPath query>)`.
- variables that return sections of XML documents queried using XPath (The `$list` variants will return a list of matches while the `$system` variants return the first matched element if any. The reference variants do not create a copy of the parts of the XML document that are returned.)
- `$system.log()`– logs a message in the http log.
- `$system.print()`– prints a message in the http.out log.
- `$system.throwException()` exception name, message (For example, “MPLS.customization”, “MPLS service cannot be provisioned because of ..”)– This is useful to throw a validation error, No configuration will be deployed. A deployed Service Request deploy that throws an exception transitions to the Invalid state and the exception message is shown in the task log and in the configuration preview.
- `$DeviceCredentials`. A set of device inventory related attributes for testing properties of the device.

## Creating Rules for CLI Templates

While creating CLI templates, you can also determine the type of devices on which the template can be deployed by specifying a set of rules in **Device Support section**. The rules are mainly categorized based on the device type, role type, and operating system. Prime Provisioning deploys the template only when the criteria specified in these rules are fulfilled. When no rules are specified, the template is deployable on all devices.

You can create multiple rules for a given template. For example, you could have one rule for a template to be deployed on only IOS-XR devices of type N-PE; while another rule for the template to be deployed on IOS devices of type U-PE.

## Importing and Exporting Customizations

You can export customizations in an XML format and save it using a text editor to create a backup of your customization. It is recommended that you create a backup of your customizations or copy the policy and modify the copy before you modify a policy with existing service requests (see [Changing Customizations When a Policy is in Use](#)), so that you can revert back to these customizations by merely importing the same XML document that you saved. To do this, an **Import/Export** button has been provided on the policy creation page. The customizations that you export are displayed in a new browser window from which you can copy the customizations onto a text editor for further use.

By exporting the customization data in an XML text format, you can:

- Apply the same customization to different policies by simply exporting the XML text and importing the same over to a new policy. This is useful when you cannot copy the whole policy for example copying a customization to a policy that is already in use with service requests.
- Edit the order in which the UI groups are placed and also edit the order in which the attributes are displayed within the UI groups.

## Changing Customizations When a Policy is in Use

The new UI attributes that you define in policies can be edited even after service requests are defined based on those policies.

To introduce a new capability for only newly created services, it is recommended that you create a new policy with this capability. This can be done by copying an existing policy to create a new one and making the current policy inactive. You can also rename the policies that you copy so that operators can use the same name for the new policy. While creating new services requests, Prime Provisioning only lists the active policies so that you do not select the inactive policies used for existing service requests. This ensures that you do not face any errors while modifying in use policies.

Changes to the attributes in the policy will cause no change to the data in the associated service requests. The changes can only be noticed in the user interface and the way the service request is configured.

To create a backup of the previous version of a customization, refer to [Importing and Exporting Customizations](#). This enables you to revert back to the previously saved version after modifying a policy that has existing service requests.

Some types of changes that you make to a policy can result in undesired changes to a service and hence it's recommended that you review existing service requests before you make these changes to the policy. The types of changes that requires you to review existing service requests are:

- **Removing an attribute:**  
When an attribute is removed from the policy page with its declaration existing still in the CLI template, an appropriate error with link "**Has errors**" is enabled in the "**Provisioning CLI Customizations**" page. On rolling over the mouse over the "Question mark" icon, the necessary details are shown. Although the removed attribute is no longer displayed and referenced from the provisioning logic and templates, it continues to exist in the service request. The saved value reappears only if you add an attribute with the same name. This behavior is to ensure that the removal of attributes is reversible step. When you remove attributes that continue to be referenced from the provisioning logic or from templates, the templates fail because they are referring to undefined attributes. Thus it is recommended that you first remove all references to the attributes, before you proceed with the removal of these attributes.
- **Removing values from the valid range of an attribute:**  
This can be done by changing a string validation regular expression, restricting an integer range, and removing values for an enumeration. After you remove these values and then edit the service request, while retaining the invalid values, you will not be able to save the service request. You will need to either change the value of the attribute or cancel your edit. Thus it is recommended that you edit service requests and not use the invalid values before you change the policy.
- **Making an attribute non-editable:**  
The attribute can not be modified and will be hidden from the service request page create using the policy with the non-editable attribute. Attribute values modified during service request creation, are not visible to the operated modifying services. To ensure that different service requests do not have different values for the same attribute, it is recommended that all service requests created with the policy contain the same default values before they are marked non-editable. Thus new service requests can only be created with the default values.

Different changes that you make to existing service requests can have varied results. The results are:

- Adding an attribute: The next time you create or edit the service request, this attribute will be added with its default value and can be referred from the templates and provisioning logic.
- Expanding the valid range of an attribute: No changes to the existing service request, however, you can edit the service request to select the new value.
- Editing the default value for an attribute: No change to the existing service requests. Only newly created service requests will take the default value.
- Make an attribute editable: The value can be modified while creating or editing existing services. The attribute will contain its former value.

After you add new attributes to a service, which translate to more lines of configuration, and re-deploy the service, managing the transition is easy since the template will be activated automatically.

However, if you remove template configurations and replace them with new configurations, you need to ensure that you maintain the decommissioning sequence of the old features before you add the new features. Once the service is migrated, you must no longer use the old features. To do this, you can introduce an additional attribute that represents whether the service is migrated or not. This can be used as a condition with an ‘if’ statement in the template to decide whether an old extension has to be decommissioned or not. For advanced help in migrating from template solutions to customizing policies, you can contact the Advanced Services team.

## Customizing MPLS-TP Policies

In MPLS-TP policy editor page, a new accordion “Additional Attributes” has been introduced. This accordion consists of a CLI Customizations text box where you can paste the customization in XML format. The customization data can be applied to either a new policy or to an existing policy. The customization is applied to the policy only when you click **Finish** or else the policy will ignore the changes introduced by the customization.

When you open the MPLS-TP policy again, you can view the customization changes listed as additional attributes in Tunnel Characteristics and Tunnel End-Points accordions. You can change the value of these additional attributes anytime by modifying the customization XML of the policy. These values get reflected on the service request associated with the policy once you click the Finish button. But you cannot change the value of the additional attributes through UI of a policy. It will not update the associated service request.

The different types of attributes supported in MPLS-TP customization are mentioned below:

- String – regular expression and length bounds for validation
- Integer – requires you to enter numbers and defines a range
- Enumeration - drop-down list
- Check box – provides a check box
- IPv4 – IP v4 address, may define range

You can also mention the different CLI merging modes in the customization XML. The supported modes are: External, Combine, ExternalWithModify. For detailed information on this, refer to the CLI Merging Mode section of [Creating Templates](#).

The Additional Attributes added through customization can be edited or removed from the UI even after the Service Request is defined for the policy but this can be done only by pasting the modified customization XML in the Additional Attributes accordion.



# Managing MPLS Traffic Engineering Services

---

This chapter contains a detailed description of the Cisco Prime Provisioning 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.

This chapter includes the following sections:

- [Getting Started, page 9-1](#)
- [TE Network Discovery, page 9-11](#)
- [TE Resource Management, page 9-21](#)
- [Basic Tunnel Management, page 9-28](#)
- [Advanced Primary Tunnel Management, page 9-45](#)
- [Protection Planning, page 9-59](#)
- [TE Traffic Admission, page 9-68](#)
- [Administration, page 9-71](#)
- [TE Topology, page 9-81](#)
- [Sample Configlets, page 9-89](#)
- [Warnings and Violations, page 9-99](#)
- [Document Type Definition \(DTD\) File, page 9-109](#)
- [Traffic Engineering Management Concepts, page 9-112](#)

## Getting Started

This section describes the installation procedure for Prime Provisioning. The general installation procedure for Cisco Prime Provisioning (Prime Provisioning) is described in the [Cisco Prime Provisioning 7.2 Installation Guide](#).

It includes the following sections:

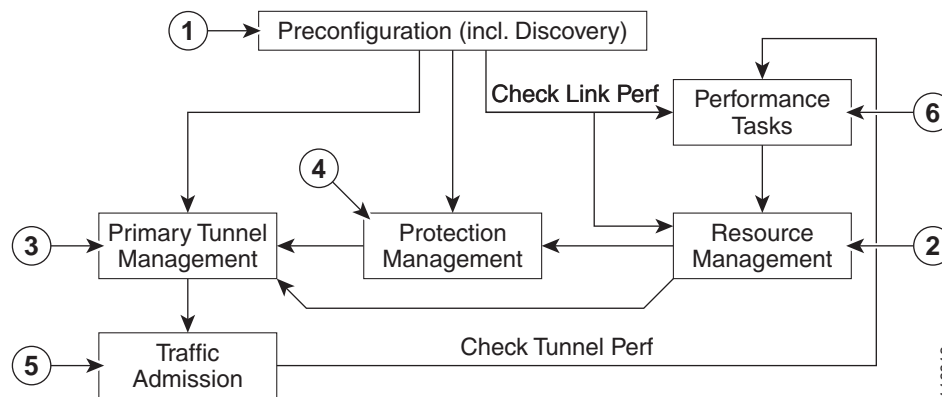
- [Prerequisites and Limitations, page 9-3](#)
  - [General Limitations, page 9-3](#)
  - [Feature-Specific Prerequisites and Limitations, page 9-3](#)
  - [Non-Cisco Devices and TEM, page 9-4](#)
  - [Supported Platforms, page 9-4](#)

- [Error Messages](#), page 9-4
- [Preconfiguration Process Overview](#), page 9-4
- [TEM Setup and Installation](#), page 9-7
  - [Editing DCPL Properties \(Optional\)](#), page 9-7
- [Creating a TE Provider](#), page 9-8

## Process Overview

The main components and flows in TEM are shown in [Figure 9-1](#).

**Figure 9-1** Main Process Flows in TEM



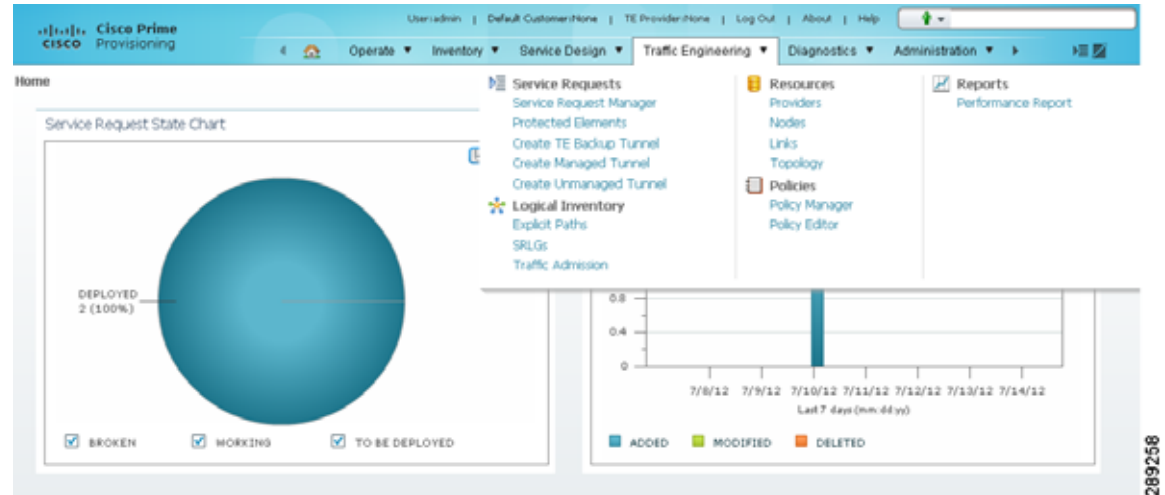
The illustration includes the following components:

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 9-1)
2. **Resource Management**—Tuning of certain properties on the TE interfaces to optimize the tunnel placement. (See [TE Resource Management](#), page 9-21)
3. **Primary Tunnel Management**—Create and manage primary tunnels, either unmanaged (see [Basic Tunnel Management](#), page 9-28) or managed. (see [Basic Tunnel Management](#), page 9-28 or [Advanced Primary Tunnel Management](#), page 9-45)
4. **Protection Management**—Protect selected elements in the network (links, routers, or SRLGs) against failure. (See [Advanced Primary Tunnel Management](#), page 9-45)
5. **Traffic admission**—Assign traffic to traffic-engineered tunnels. (See [TE Traffic Admission](#), page 9-68)
6. **Performance Tasks**—Calculates interface/tunnel bandwidth utilization using the Simple Network Management Protocol (SNMP). (See [Administration](#), page 9-71)

The Traffic Engineering menu options in the Prime Provisioning user interface are shown in [Figure 9-2](#).



Figure 9-2 Traffic Engineering Menu Options



## Prerequisites and Limitations

The current release of Prime Provisioning involves certain prerequisites and limitations, which are described in this section.

See the [Cisco Prime Provisioning 7.2 Installation Guide](#) for general system recommendations.

### General Limitations

The present release of Prime Provisioning has the following limitations:

- Although concurrent use of Prime Provisioning is supported in the Planning portion of the current implementation (see the section [Multiple Concurrent Users](#), page 9-115), 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 Provisioning.
- 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 Provisioning 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 [Traffic Engineering Management Concepts](#), page 9-112.

- A number of specific requirements are associated with the TE Discovery task. These are described in [TE Discovery Prerequisites and Limitations, page 9-13](#).
- Prime Provisioning manages a single OSPF area or IS-IS level. Prime Provisioning 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 Provisioning only supports MPLS-TE topology with point-to-point links.
- Prime Provisioning supports JRE version 7 (update 21) through to JRE version 7 (update 45) without issue. If using JRE version 7 (update 51) or later, the JRE security level must be reduced from High to Medium in order to launch the TE Topology Tool.
- For the topology viewer and the deprecated L3 topology viewer to function, Java Runtime Environment (JRE) version 7 (update 21) or later must be configured on the system running the browser.

## Non-Cisco Devices and TEM

Prime Provisioning does not manage non-Cisco devices and Prime Provisioning cannot be used to provision them.

Prime Provisioning 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 Provisioning. TE tunnels originating from non-Cisco devices will not be discovered.

Sorting can be performed on different attributes in various parts of the Prime Provisioning 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 Provisioning 7.2 Installation Guide](#).

## Error Messages

Warnings and violations that are invoked when using the TE planning tools in Prime Provisioning are documented in [Warnings and Violations, page 9-99](#)

Elixir warning messages might appear when performing deployments in Prime Provisioning:

```
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.
```

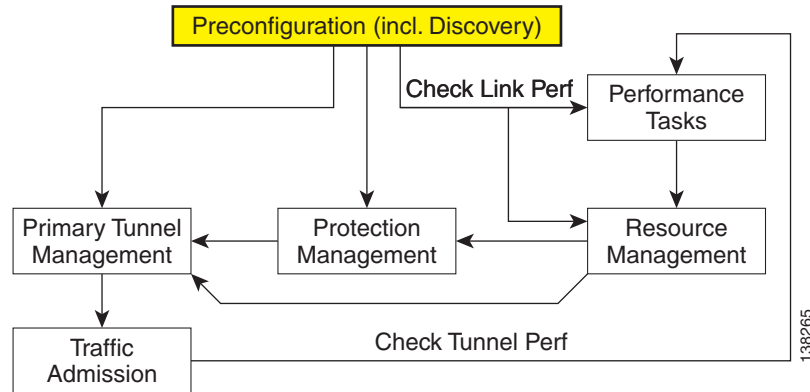
The deployments will, however, be successful and these messages can be safely ignored.

## 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.

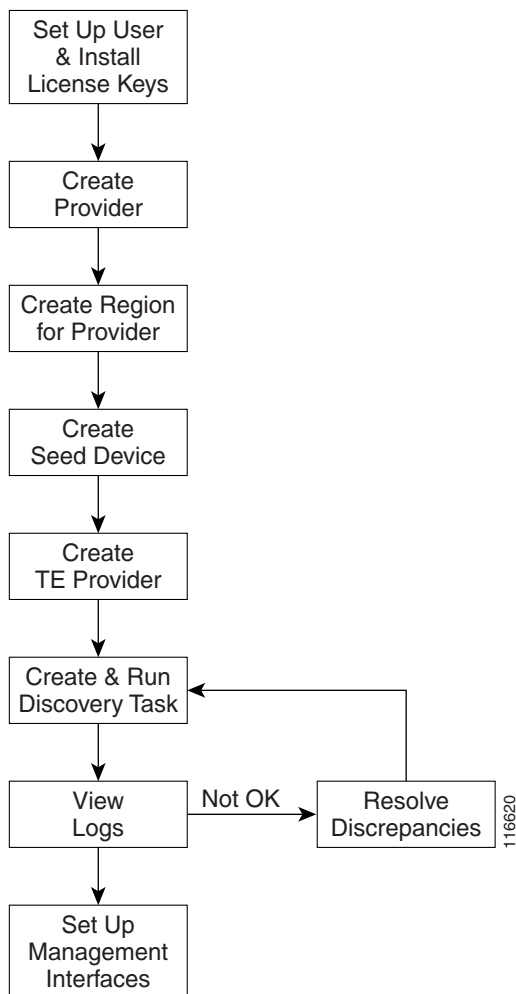
The highlighted box in [Figure 9-3](#) shows where in Prime Provisioning the preconfiguration steps take place.

**Figure 9-3** *Prime Provisioning Process Diagram - Preconfiguration*



The different steps in the preconfiguration process are provided in [Figure 9-4](#).

Figure 9-4 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 Provisioning, it is necessary to create a new user and install license keys. These keys allow you to view and manage the TE tunnels and resources using Prime Provisioning. (See [TEM Setup and Installation](#), page 9-7)
2. **Create a provider**—The provider is a concept designed to allow many different operators to work on Prime Provisioning 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 2-14.)
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 9-8.](#))
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 9-11.](#))
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 9-20.](#))

**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 Provisioning, the Prime Provisioning software must be installed. To do so, see the [Cisco Prime Provisioning 7.2 Installation Guide](#).

To set up a new Prime Provisioning user, one or more users with a TE role must be created. For step by step instructions, see [Cisco Prime Provisioning 7.2 Administration Guide](#).

Licensing information, including the Prime Provisioning licensing options and the procedure needed to install licenses is described in [Cisco Prime Provisioning 7.2 Administration Guide](#).

## Editing DCPL Properties (Optional)

The Prime Provisioning Dynamic Component Properties Library (DCPL) includes a wide variety of properties that are accessible from the GUI, some of which can be modified.

**Warning**

**Do not attempt to modify the DCPL properties unless you fully understand the implications.**

In the Prime Provisioning 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 9-4](#)).

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 [Cisco Prime Provisioning 7.2 Administration Guide](#).

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 9-5](#) appears.

Figure 9-5 Create/Edit TE Provider

Create/Edit TE Provider

| TE Provider Info:                    |                                                                          |
|--------------------------------------|--------------------------------------------------------------------------|
| TE Provider * :                      | te_provider2                                                             |
| Provider * :                         | Select Provider1                                                         |
| TE Provider Area:                    |                                                                          |
| TE Area                              | 100                                                                      |
| Primary Route Generation Parameters: |                                                                          |
| Default Primary RG Timeout (sec) * : | 100                                                                      |
| Backup Route Generation Parameters:  |                                                                          |
| Backup RG Timeout (sec) * :          | 1000                                                                     |
| FRR Protection Type * :              | <input checked="" type="radio"/> Sub Pool <input type="radio"/> Any Pool |
| Default Link Speed Factor * :        | 1.00                                                                     |
| Minimum Bandwidth Limit (Kbps) * :   | 10                                                                       |
| Max. Load Balancing Tunnel Count * : | 1                                                                        |
| Discovery Default Parameters:        |                                                                          |
| Default Region for TE Devices * :    | Select Region4                                                           |
| Customer for Primary Tunnels:        | Select                                                                   |
| Select as default TE provider:       | <input type="checkbox"/>                                                 |
| Save Cancel                          |                                                                          |
| Note: * - Required Field             |                                                                          |

238201

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 Provisioning 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 [Traffic Engineering Management Concepts](#), page 9-112.

- **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 [Traffic Engineering Management Concepts, page 9-112](#).

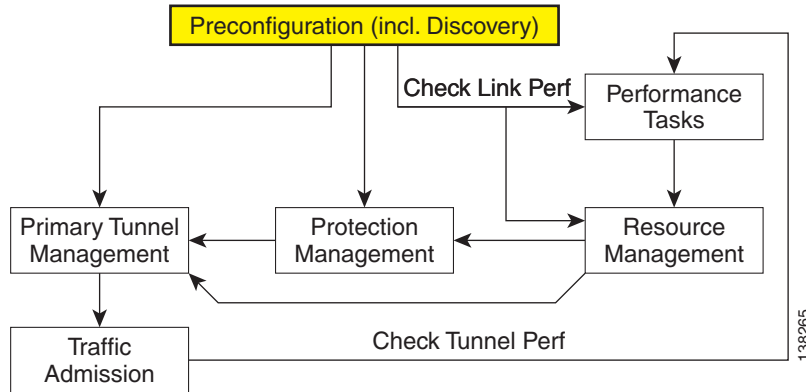
- Step 9** Fill in the remaining required fields (marked ‘\*’) and any optional fields as desired.
- Step 10** For the required **Default Region for TE Devices** field, click the corresponding **Select** button.  
The Region for Create TE Provider window appears.
- Step 11** Select the desired region 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 top of the Prime Provisioning window above the menu toolbar ([Figure 9-2](#)) 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 9-6](#) shows where in Prime Provisioning the preconfiguration steps takes place.

Figure 9-6 Prime Provisioning 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, page 9-13](#)
  - [Accessing TE Routers for TE Discovery, page 9-13](#)
  - [Memory Shortage on Large Networks, page 9-13](#)
  - [IOS XR and Enable Passwords, page 9-14](#)
- [Creating a TE Discovery Task, page 9-14](#)
  - [TE Incremental Discovery, page 9-14](#)
  - [TE Full Discovery, page 9-15](#)
- [Managing Per Area Discovery, page 9-16](#)
  - [Performing a Per Area TE Discovery, page 9-16](#)
  - [Running a Per Area TE Discovery Through an ABR, page 9-17](#)
- [Verifying a TE Discovery Task, page 9-17](#)
  - [Task Logs, page 9-17](#)
  - [View Network Element Types, page 9-20](#)
- [Setting Up Management Interfaces, page 9-20](#)
  - [MPLS-TE Management Process, page 9-20](#)
  - [Configuring Ethernet Links, page 9-20](#)

## TE Discovery Prerequisites and Limitations

The following prerequisites apply mainly to TE discovery.

For an overview of the general Prime Provisioning prerequisites and limitations, see [Prerequisites and Limitations, page 9-3](#).

### 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 Provisioning 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 Provisioning Traffic Engineering Management (TEM) management station to each device.

See [Preconfiguration Process Overview, page 9-4](#) 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 9-14](#)
- [TE Full Discovery, page 9-15](#)

## 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 Provisioning.

For background information on this topic, see the section Multiple OSPF Areas in [Traffic Engineering Management Concepts, page 9-112](#).

This section describes the following:

- [Performing a Per Area TE Discovery, page 9-16](#)
- [Running a Per Area TE Discovery Through an ABR, page 9-17](#).

## 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 Provisioning 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 9-7](#) informing you to either specify an area identifier for the TE provider or use a non-ABR device as the seed.

**Figure 9-7** TE Discovery Through an ABR with no TE Area Identifier Specified

| Date                | Level   | Component       | Message                                                                                                                                                                                                                                                                                           |
|---------------------|---------|-----------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 2011-03-08 07:49:42 | WARNING | repository/rbac | Thread RBAC enabled flag is set to false.                                                                                                                                                                                                                                                         |
| 2011-03-08 07:49:55 | SEVERE  | DiscoveryTask   | Seed device 192.168.1.128 has TE enabled in multiple IOP areas. This configuration is unsupported with the specified TE Provider, aborting discovery. Retry discovery from a seed device with TE enabled in one IOP area or specify the area you wish to be discovered by adding the TE Provider. |
| 2011-03-08 07:49:55 | WARNING | DiscoveryTask   | Fatal Error Encountered, aborting Discovery.                                                                                                                                                                                                                                                      |
| 2011-03-08 07:49:55 | SEVERE  | DiscoveryTask   | Discovery FAILURE.                                                                                                                                                                                                                                                                                |
| 2011-03-08 07:49:55 | WARNING | repository/rbac | Thread RBAC enabled flag is set to true.                                                                                                                                                                                                                                                          |

## 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 occurred in the network.
- **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 9-8](#). 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 9-8** TE Discovery Task Log - Example 1

| 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: SOLKTXESBAW                                                         |
| 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 9-9](#) and [Figure 9-10](#) show a sample device debug and information section.



Figure 9-9 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 9-10 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 bandwidth.
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 no 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-
WJRDUT307AW-CLSPCOYK8AW-2 WJRDUT307AW-CLSPCOYK8BW-1 WJRDUT307AW-
HCHILMLT7AW-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-SOLKTXE8BAW-2 WJRDUT307AW-SOLKTXE8BW-1
DEBUG: MplsTeShowRunningCallback: Device: WJRDUT307AW, has tunnels: 1003 1004 1001 10500

```

Step 3 Click **Return to Logs** to quit the current log with the option to open another log.

## 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:

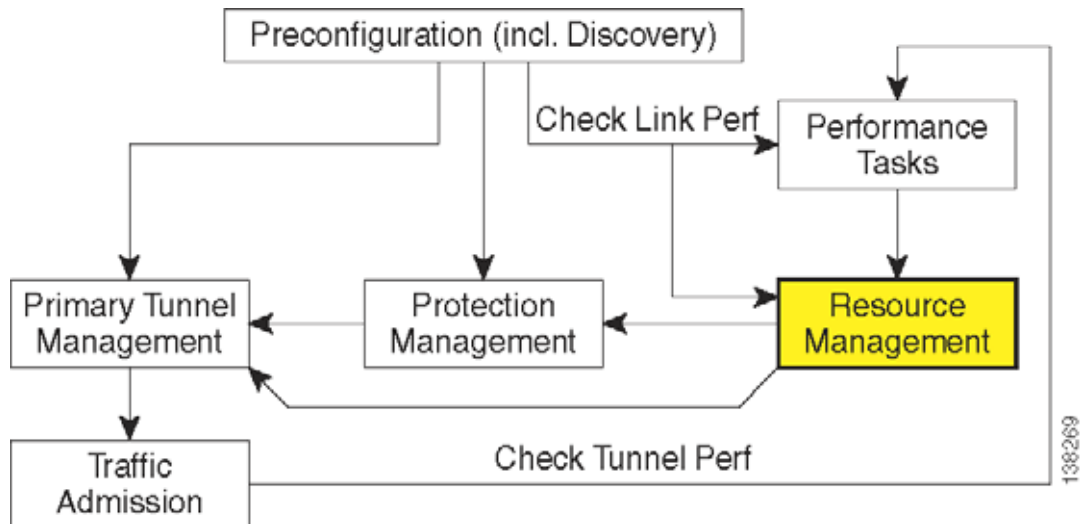
```
router ospf <id> area <area identifier> interface <name> network point-to-point
```

# 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 9-11](#) shows where in Prime Provisioning resource management occurs.

**Figure 9-11** Prime Provisioning Process Diagram - Resource Management



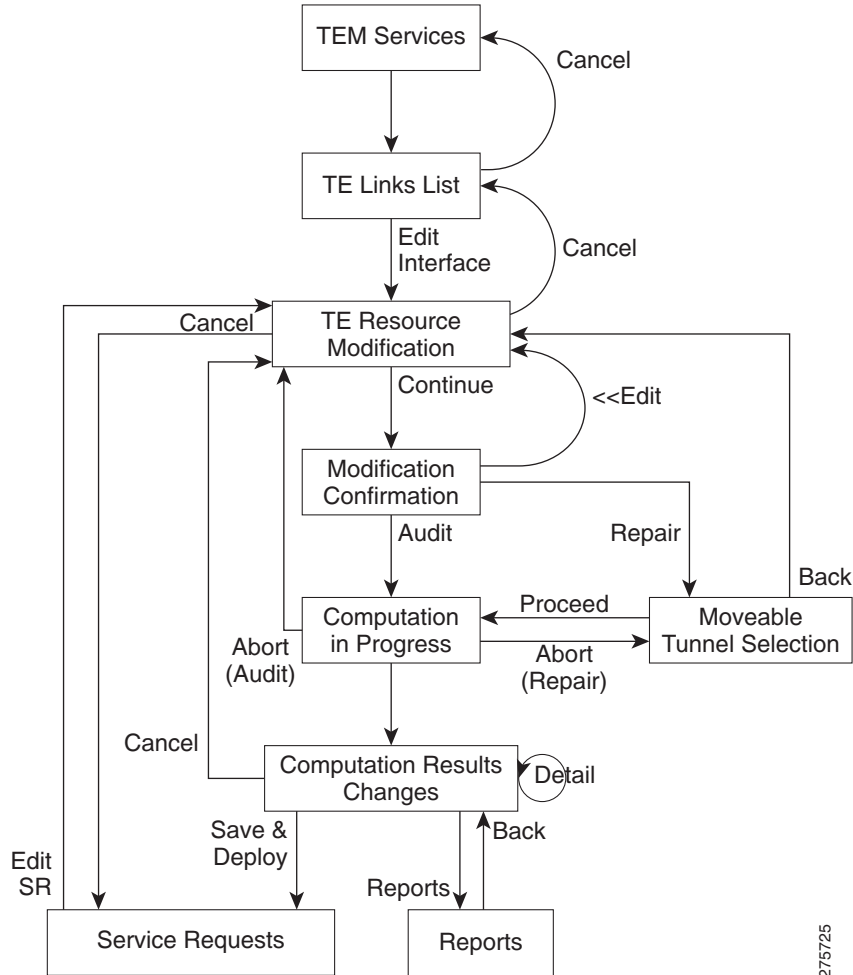
When a tunnel placement is attempted and there is insufficient bandwidth, sometimes the resources on the TE links can be changed and the tunnel placement retried.

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 Provisioning 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 9-12](#).

Figure 9-12 Resource Management Processes



This section includes the following:

- [Modifying Network Resources, page 9-22](#)
- [Changing Link Status, page 9-24](#)
- [Deleting TE Links, page 9-25](#)
- [Deleting TE Tunnels, page 9-26](#)
- [Deleting TE Nodes, page 9-27.](#)

## 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 Provisioning. 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 Prime Provisioning 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 9-45](#)  
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 Provisioning, 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 9-17](#).

## 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 Provisioning 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 9-45](#).
-

## 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 Provisioning 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 9-39](#) or [Delete Backup Tunnel, page 9-44](#)).

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 tunnel 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 Provisioning 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 interfaces 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 Provisioning, 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 9-64.)
9. Run network grooming (see [Grooming](#), page 9-58) to optimize the managed tunnels, so that they will make use of the new node(s).

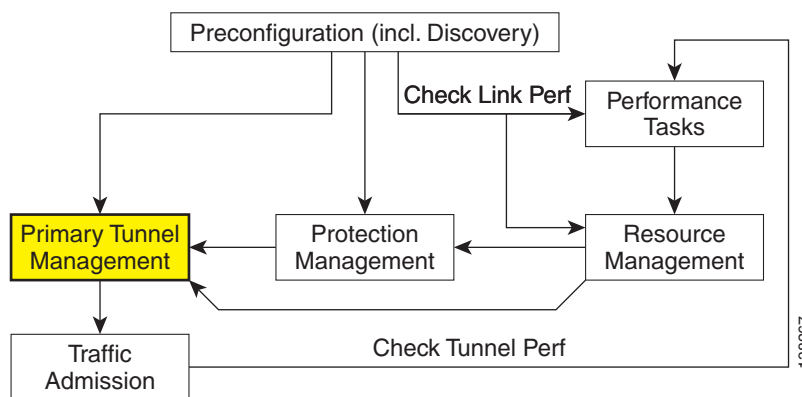
If this check succeeds, the TE node and all TE links and TE interfaces starting at that node are removed from the repository.

## Basic Tunnel Management

This section describes the processes involved in creating primary and backup tunnels with Prime Provisioning. To create a tunnel, certain steps must first be performed as described in previous sections.

The highlighted box in [Figure 9-13](#) shows where in Prime Provisioning primary tunnel management occurs.

**Figure 9-13** Prime Provisioning 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 [Traffic Engineering Management Concepts, page 9-112](#).

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 [Traffic Engineering Management Concepts, page 9-112](#).

This section includes the following:

- [Create TE Policy, page 9-29](#)
- [Create Explicit Path, page 9-30](#)
  - [Delete Explicit Path, page 9-32](#)

- [Primary Tunnel Operations, page 9-32](#)
  - [Create Primary Tunnel, page 9-33](#)
  - [Edit Primary Tunnel, page 9-38](#)
  - [Delete Primary Tunnel, page 9-39](#)
- [Backup Tunnel Operations, page 9-40](#)
  - [Create Backup Tunnel, page 9-40](#)
  - [Edit Backup Tunnel, page 9-43](#)
  - [Delete Backup Tunnel, page 9-44.](#)

## 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 9-30](#).

For other TE policy management operations, see [TE Policies, page 9-72](#).

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 Provisioning 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 [Traffic Engineering Management Concepts, page 9-112](#).

Policies are managed under **Policies** in **Service Design**. For a more detailed explanation of the **Policies** GUI, see [TE Policies, page 9-72](#).

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 9-72](#).

**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 Provisioning 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 Provisioning 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 Provisioning.

When the explicit path configuration is removed from the device, the explicit path will still exist in the Prime Provisioning 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 Provisioning (through CLI on the device itself, for example). However, when a transaction reroutes, deletes, or modifies a tunnel using Prime Provisioning 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 Provisioning 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 [Traffic Engineering Management Concepts, page 9-112](#).

To create a managed or an unmanaged primary tunnel, use the following steps:

- Step 1** Choose **Traffic Engineering**.
- Step 2** Click **Create Managed Tunnel**. The TE Managed Primary Tunnels SR window appears as shown in [Figure 9-14](#).
- or
- Click **Create Unmanaged Tunnel**. The TE Unmanaged Primary Tunnels SR window appears.

**Figure 9-14** Create TE Managed Primary Tunnel

The screenshot displays the 'Create TE Managed Primary Tunnel' configuration window. At the top, it shows 'SR Job ID: New', 'SR ID: New', and 'SR State: REQUESTED'. Below this are several input fields: 'Creator', 'Head Device' (with a 'Select' button), 'Destination Device' (with a 'Select' button), 'Tunnel Policy' (with a 'Select' button), 'Tunnel Bandwidth (Kbps)', 'Description', 'Tunnel Number' (with 'Auto Gen' checked), 'Tunnel ID', and 'Customer'. There is also an 'Auto D/W' section with an 'Enable' checkbox and three input fields for 'Freq (sec)', 'Min (Kbps)', and 'Max (Kbps)'. A 'Path Options' section contains a table with two rows. The first row is for 'System Path' (Explicit) and the second is for 'Dynamic Path' (Dynamic). Both rows have a 'Lock Down' checkbox. At the bottom, there are 'Add', 'Delete', 'OK', and 'Cancel' buttons, and a note: '\* - Required Field'. The page number '238202' is visible on the right side.

| # | Option # | Path Name    | Path Type | Lock Down                |
|---|----------|--------------|-----------|--------------------------|
| 1 | 1        | System Path  | Explicit  | <input type="checkbox"/> |
| 2 | 2        | Dynamic Path | Dynamic   | <input type="checkbox"/> |

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 Provisioning.
- **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 accommodate 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 Provisioning creates new tunnels using auto-gen and this tunnel number might already be used by an MDT GRE tunnel. As a result, Prime Provisioning uses high tunnel numbers to avoid any complications.

---

- **Tunnel ID**—Unique tunnel identifier used within Prime Provisioning.
- **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 9-29](#)) 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 9-30](#).
- 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 9-30](#).
- 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 37) 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 Provisioning 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 Provisioning 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 9-45](#)).



**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 E-106](#).

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 9-38](#).

---

## 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 9-14](#) 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 9-33](#).
- 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 9-38](#) 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 9-26](#)) 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 9-33](#).



---

**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 Provisioning allows you to perform a number of backup tunnel operations, which are described in this section.

[Traffic Engineering Management Concepts, page 9-112](#) 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 9-30](#).

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 Provisioning.
- **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 Conformance = true if it has zero tunnel bandwidth, unlimited backup bandwidth, and an 'exclude address' first path option. Otherwise, it is marked Conformance = 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 [Traffic Engineering Management Concepts, page 9-112](#).
- **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 9-15](#) appears.

**Figure 9-15** Create TE Backup Tunnel

Create TE Managed Primary Tunnel

SR Job ID: New SR ID: New SR State: REQUESTED  
 Creator: Type: ADD

Head Device:   
 Destination Device:   
 Tunnel Policy:   
 Tunnel Bandwidth (Kbps):   
 Description:   
 Tunnel Number: Auto Gen    
 Tunnel ID:   
 Customer:   
 Auto BW: Enable:   
 Freq (sec):   
 Min (Kbps):   
 Max (Kbps):

**Path Options:**

| # | Option #                       | Path Name    | Path Type | Lock Down                |
|---|--------------------------------|--------------|-----------|--------------------------|
| 1 | <input type="text" value="1"/> | System Path  | Explicit  | <input type="checkbox"/> |
| 2 | <input type="text" value="2"/> | Dynamic Path | Dynamic   | <input type="checkbox"/> |

Showing 1 - 2 of 2 records

Rows per page: 10 Page 1 of 1

Note: \* - Required Field

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 [Traffic Engineering Management Concepts, page 9-112](#).

- **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 Provisioning.
- **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 [Traffic Engineering Management Concepts, page 9-112](#).
  - **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 9-30](#).

**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 Provisioning 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 Provisioning 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 E-106](#).

## 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 9-40](#).

**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 9-43](#) 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 9-26](#)) 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 9-40](#).



**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.

## Deleting a Service Request

The Delete operation in the Service Request Manager window is designed to remove a service request from the repository without affecting the network.

The **Delete** button has 2 options:

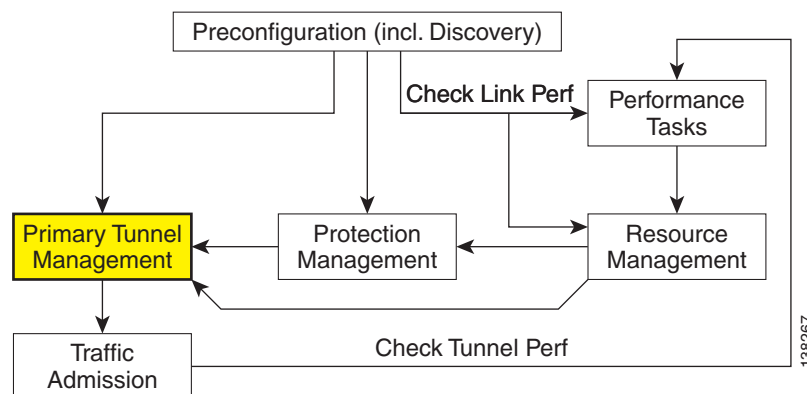
- **Delete**—The regular delete can only be used on the service request in **CLOSED** state. Therefore, it cannot be used on TE Resource, TE Tunnel, or TE Protection service requests because these cannot be decommissioned. These three types of service requests can only be force deleted.
- **Force Delete**—During 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.

## Advanced Primary Tunnel Management

In addition to the basic tunnel management tools described in [Basic Tunnel Management, page 9-28](#), Prime Provisioning 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 9-16](#) shows where in Prime Provisioning primary tunnel management occurs.

**Figure 9-16** Prime Provisioning 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 [Traffic Engineering Management Concepts, page 9-112](#).

This section includes the following:

- [Tunnel Operations, page 9-46](#)
  - [Create Primary Tunnel, page 9-47](#)

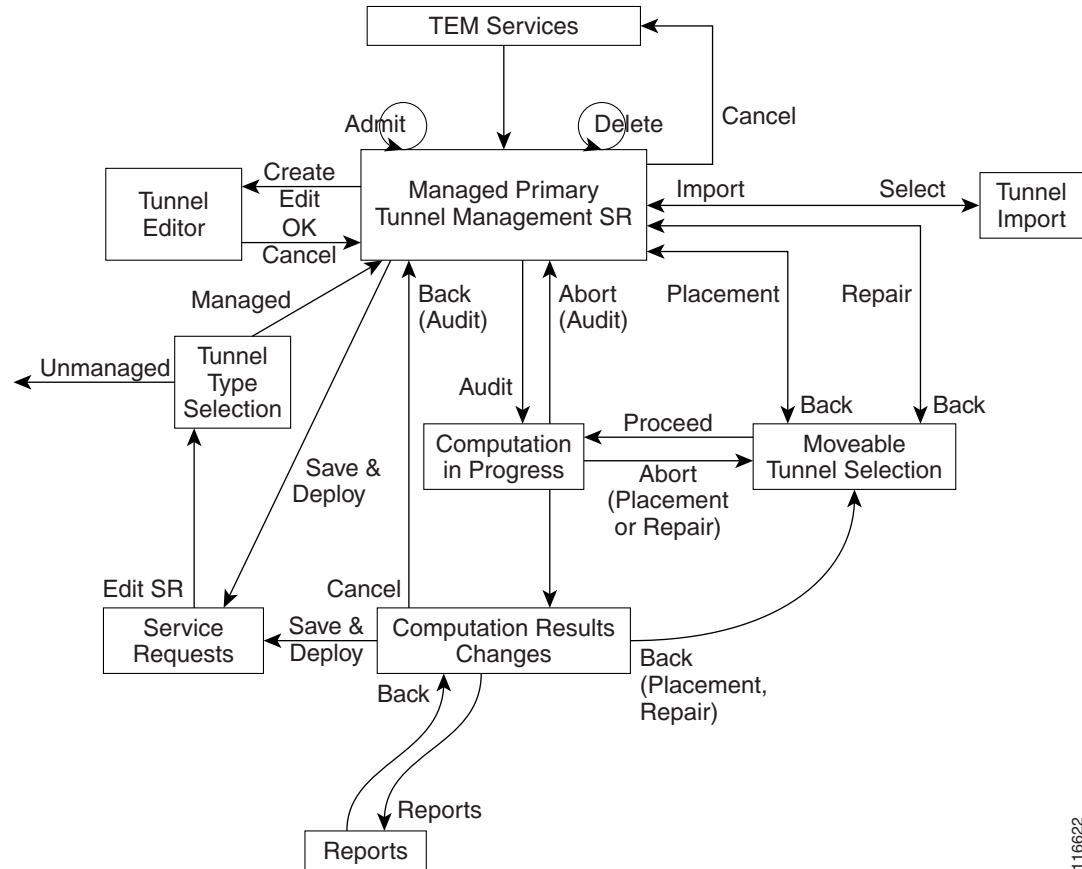
- Edit Primary Tunnel, page 9-50
- Delete Primary Tunnel, page 9-50
- Admit Primary Tunnel, page 9-50
- Import Primary Tunnel, page 9-50
- Planning Strategy, page 9-52
- Placement Tools, page 9-53
  - Tunnel Audit, page 9-53
  - Tunnel Placement, page 9-56
  - Tunnel Repair, page 9-57
  - Grooming, page 9-58.

## Tunnel Operations

This section explains the advanced tunnel operations in Prime Provisioning that incorporate the planning tools.

An overview of the primary tunnel management process is provided in [Figure 9-17](#).

Figure 9-17 Primary Tunnel Management Processes



116622

For **Tunnel Type Selection**, when you select **Unmanaged** the TE Unmanaged Primary Tunnel SR window appears (see [Basic Tunnel Management, page 9-28](#)).

All other elements in [Figure 9-17](#) 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 9-33](#).
- 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 9-33](#).
- The **Path Options** section provides three path types, **System Path**, **Explicit Path**, and **Dynamic Path**.

A **System Path** is an Prime Provisioning 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 9-72](#)) 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 [Traffic Engineering Management Concepts, page 9-112](#).
  - **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 (only managed tunnels)**—Specifies that primary tunnels should be routed only through links and SRLGs that are protected by FRR backup tunnels.
  - **Link, SRLG and Node (only managed tunnels)**—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 Provisioning, delete the system path and subsequently add the explicit path.

For a more detailed explanation of this step, see [Create Primary Tunnel, page 9-33](#).

**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 9-53](#)) 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 9-33](#).

When you click **Save & Deploy**, Prime Provisioning 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 Provisioning 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 12-5](#).

## 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 9-38](#)

## Delete Primary Tunnel

To delete one or more tunnels, see [Delete Primary Tunnel, page 9-39](#).

## 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 <b>TE Managed Primary Tunnel SR</b> , select one or more unverified tunnels to migrate.                                              |
| Step 2 | Click <b>Admit</b> .                                                                                                                        |
|        | The unverified tunnel(s) are verified and, if successful, an <b>ADMIT</b> flag will appear in the <b>Op</b> column.                         |
| Step 3 | Choose <b>Proceed with Changes &gt; Tunnel Placement</b> 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 9-109](#)), and save the XML file together with the DTD file on the Prime Provisioning server under the same directory. To create a valid import file, use the provided command line validation tool (see [Command Line Validation Tool, page 9-51](#)).

The following files are necessary for importing data into the Prime Provisioning 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 Provisioning 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 9-46](#). 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 9-21](#).)

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 9-47](#)), 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 9-99](#)

**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 Provisioning 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 9-47](#)), 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 9-21](#).)

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 9-47](#)), 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 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 9-99](#)

**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 9-99](#)

**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.

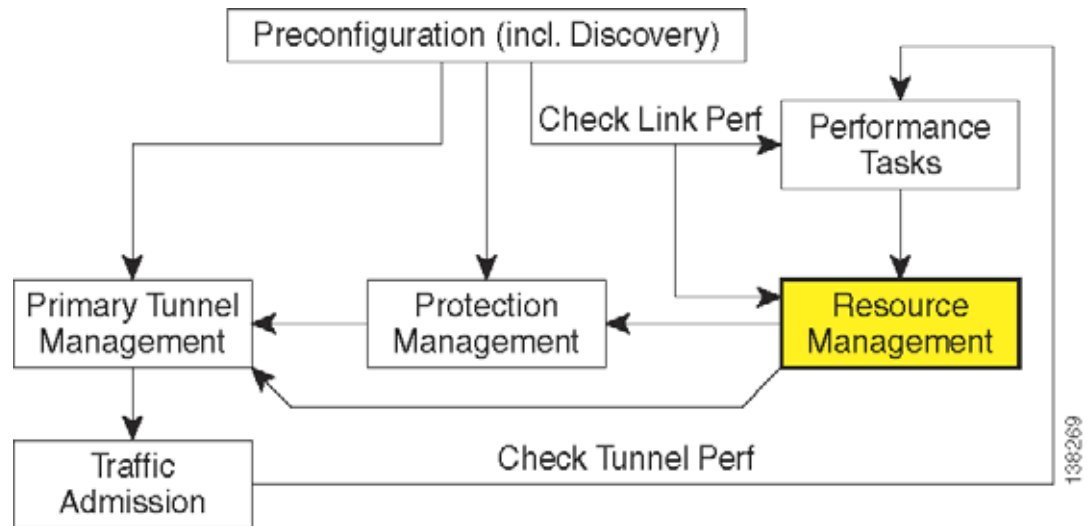
For more information on working with service requests, see the managing service requests part elsewhere in this guide.

## 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 9-28](#) for a description of the process using the basic tools.

The highlighted box in [Figure 9-18](#) shows where in Prime Provisioning protection management occurs.

Figure 9-18 Prime Provisioning Process Diagram - Protection Management



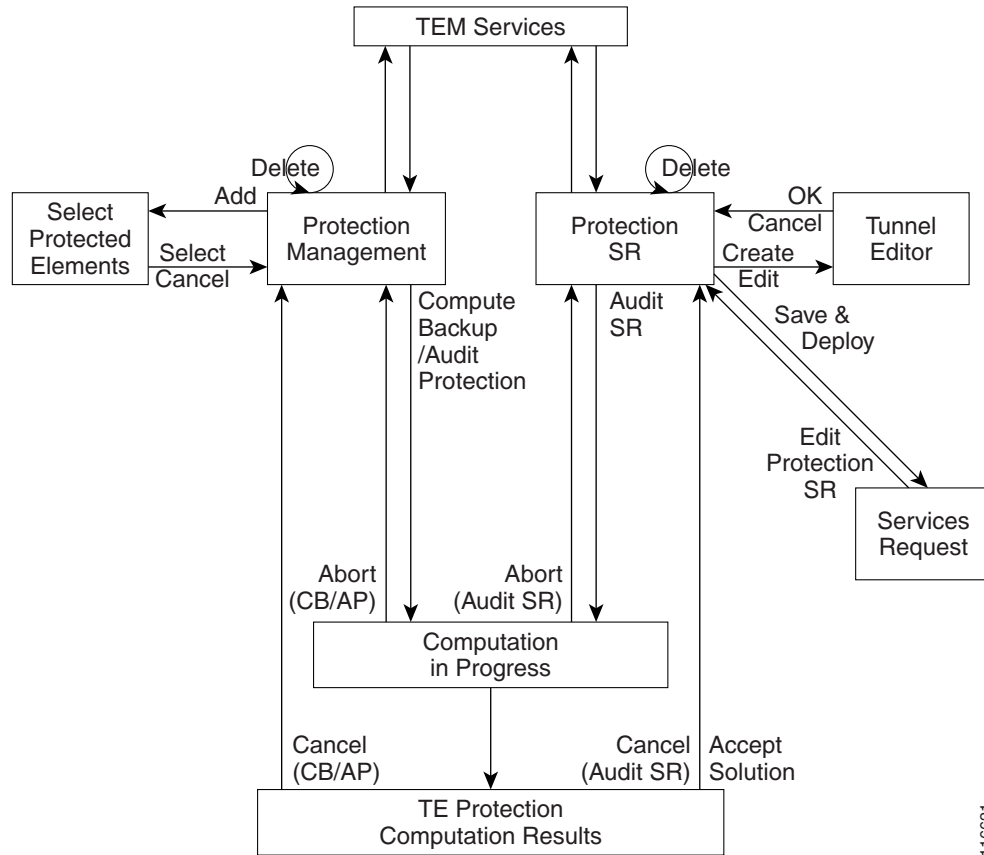
The purpose of protection planning is to protect selected elements in the network (links, routers, or SRLGs) against failure.

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 9-19](#).

Figure 9-19 Protection Management Processes



116621

This section includes the following:

- [SRLG Operations, page 9-62](#)
  - [Create SRLG, page 9-62](#)
  - [Edit SRLG, page 9-62](#)
  - [Delete SRLG, page 9-63](#)
- [Configure Element Protection, page 9-63](#)
- [Protection Tools, page 9-63](#)
  - [Compute Backup, page 9-64](#)
  - [Recompute Backup, page 9-65](#)
  - [Audit Protection, page 9-66](#)
  - [Audit SR, page 9-67.](#)

## 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 9-63.
- 

## Protection Tools

Relying on manual creation of backup tunnels as described in [Basic Tunnel Management](#), page 9-28 has its limitations, not just for larger and more complicated networks.

The protection tools available in Prime Provisioning 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 Provisioning automatically compute the necessary backup tunnels to protect specified network elements. The manual process is described in [Basic Tunnel Management, page 9-28](#)

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 9-63](#).
  - 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 9-99](#)

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 9-40](#).

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 9-67](#)).

**Step 7** Click **Save & Deploy** to deploy the new tunnel SR to the network.

When you click **Save & Deploy**, Prime Provisioning 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 Provisioning 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 E-106](#).

## Recompute Backup

Recompute Backup is used to automatically recompute existing backup tunnels to update protection for specified network elements that are in either **Protected**, **Not Fully Protected**, or **Unknown** state.

The function Compute Backup Tunnels attempts to minimize changes to existing tunnels. Thus, if it can create new backup tunnels to protect the required elements without making any changes to existing tunnels, it will do so. Minimizing change is useful but also has a disadvantage if new resources such as more links or more bandwidth on the links have been added to the network. The current tunnels would not be changed to take advantage of those new resources even though they might provide better and shorter protection paths.

The Recompute Backup Tunnels function is for just these cases. It will compute backup paths for new and existing FRR tunnels without attempting to maintain the current paths.

To run Recompute Backup, use the following steps:

- 
- Step 1** Choose **Traffic Engineering > Protected Elements**.
- Step 2** Select one or more elements on which to recompute a backup path.
- Step 3** Click **Recompute 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.

For a description of these fields, see [Step 4](#) under [Compute Backup, page 9-64](#).

- Step 4** For the rest of the procedure, see Step 5 and onwards in the procedure documented in [Compute Backup, page 9-64](#).
- 

## Audit Protection

As opposed to the Compute Backup tool described on page 64, 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 membership, 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 9-64](#).



**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 9-64](#).

**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 9-99](#)

**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 9-40](#).

**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 9-66](#) for a description of the rest of the process. Detail and report windows are identical in these two processes.

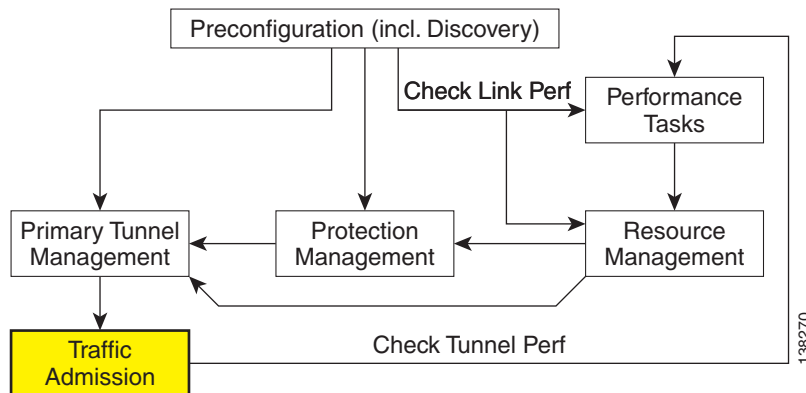
## TE Traffic Admission

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 Provisioning Traffic Engineering Management (Prime Provisioning) 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 9-20](#) shows where in Prime Provisioning TE Traffic Admission occurs.

**Figure 9-20** Prime Provisioning 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, page 9-69](#)
- [Deploying a TE Traffic Admission SR, page 9-70](#)
- [Other Traffic Admission SR Operations, page 9-71](#)
- [Viewing the SR State, page 9-71.](#)

## 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 Provisioning 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 9-70](#).

---

## 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 E-106](#).

## Administration

A number of administrative features in Cisco Prime Provisioning Traffic Engineering Management (TEM) are common to Prime Provisioning. Instructions on how to use these features are described in detail starting in [Cisco Prime Provisioning 7.2 Administration Guide](#).

In this section, only TE-specific administrative features are described.

This section includes the following:

- [TE User Roles, page 9-72](#)
- [TE Policies, page 9-72](#)
  - [Create Policy, page 9-72](#)
  - [Edit Policy, page 9-73](#)
  - [Delete Policy, page 9-74](#)
- [TE Tasks, page 9-74](#)
  - [Creating a TE Task, page 9-74](#)
    - [Creating a TE Functional Audit Task, page 9-75](#)
    - [Creating a TE Interface Performance Task, page 9-76](#)
- [SR History and Configlets, page 9-78](#)
- [Managing the Locking Mechanism, page 9-78.](#)

## 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 Provisioning and how to use them, see [Cisco Prime Provisioning 7.2 Administration Guide](#).

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, page 9-72](#)
- [Edit Policy, page 9-73](#)
- [Delete Policy, page 9-74](#)

## Create Policy

Prime Provisioning 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 appears.
- Step 2** Click **Create** and select **TE** from the drop-down list to set up a new TE policy.

The TE Policy Editor window appears.

It includes the following fields:

- **Policy Name**—Name of the TE policy chosen by the user.
- **Policy 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 [Traffic Engineering Management Concepts, page 9-112](#).
  - **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.
- **Delayed Constraint**—Apply a constraint when optimizing paths or placing tunnels.
  - **Max. Delay (msec)**—Sets the maximum delay allowed for each managed tunnel in a given policy.
- **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 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 9-72](#). 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 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 Provisioning 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 9-11](#).

## 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 [This chapter contains the following sections:, page 12-1](#)

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 [This chapter contains the following sections:, page 12-1](#).
- 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.

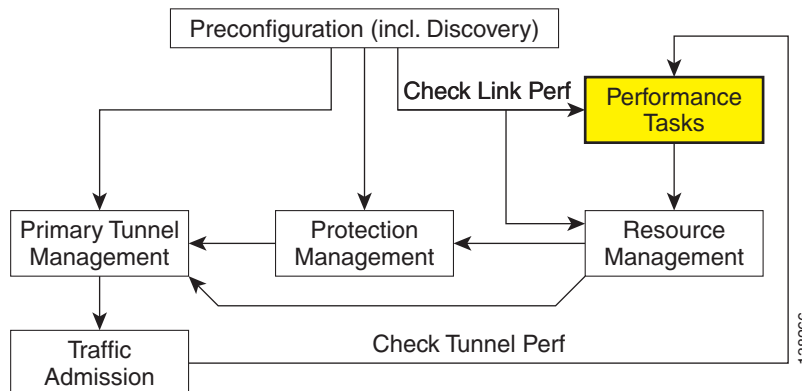
To view the task logs for the created tasks, see [Viewing a Task Log, page E-106](#).

## 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 9-21](#) shows where in Prime Provisioning traffic admission occurs.

**Figure 9-21** Prime Provisioning 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 [This chapter contains the following sections](#), page 12-1.
- 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 E-107](#).

To view the task logs for the created tasks, see [Viewing a Task Log, page E-106](#).

## 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 Provisioning 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.

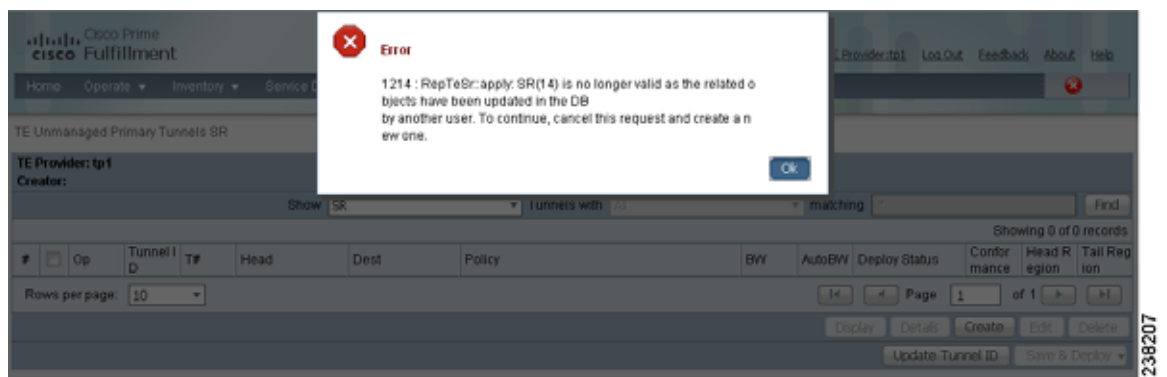
This section describes the following errors:

- [Modifying Locked Object, page 9-80](#)
- [Modifying Object After Lock Is Released, page 9-80](#)
- [Deleting Link with Associated TE Object, page 9-81](#)
- [Deleting Link Without Associated TE Object, page 9-81](#)

## 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 9-22](#).

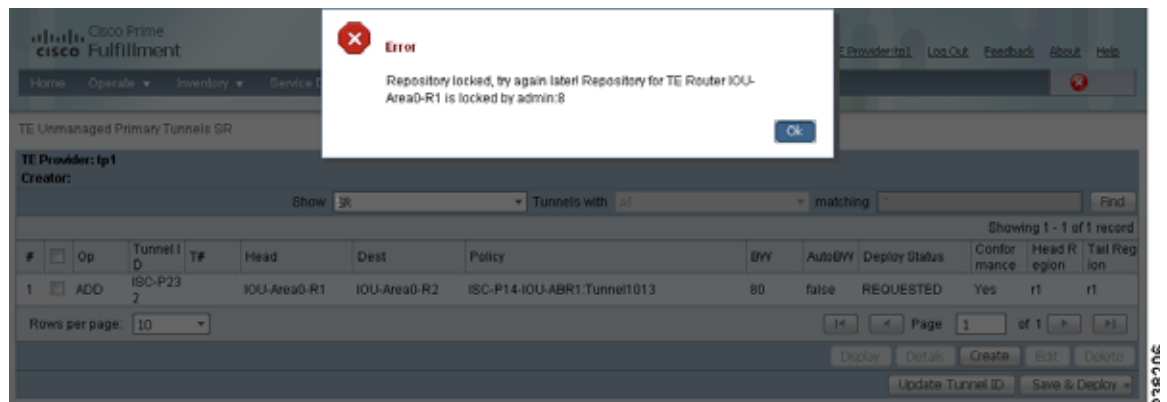
**Figure 9-22** Modifying Locked Object



## Modifying Object After Lock Is Released

If you attempt to modify an object after the lock is released, Prime Provisioning 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 9-23](#).

**Figure 9-23** Modifying Object After Lock Is Released

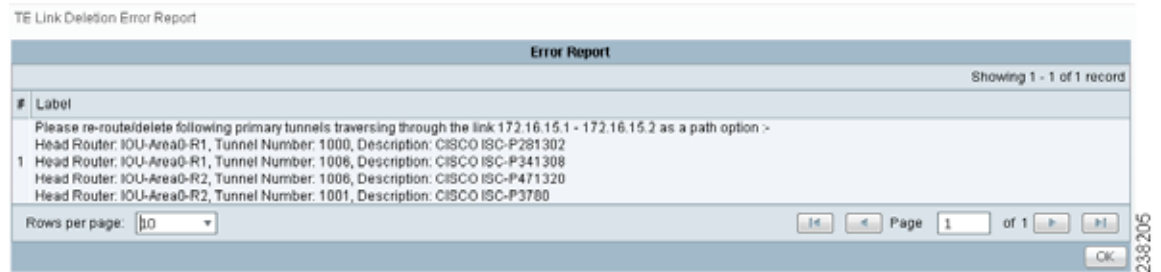


## Deleting Link with Associated TE Object

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 9-24](#) is displayed.

**Figure 9-24** Deleting Link with Associated TE Object

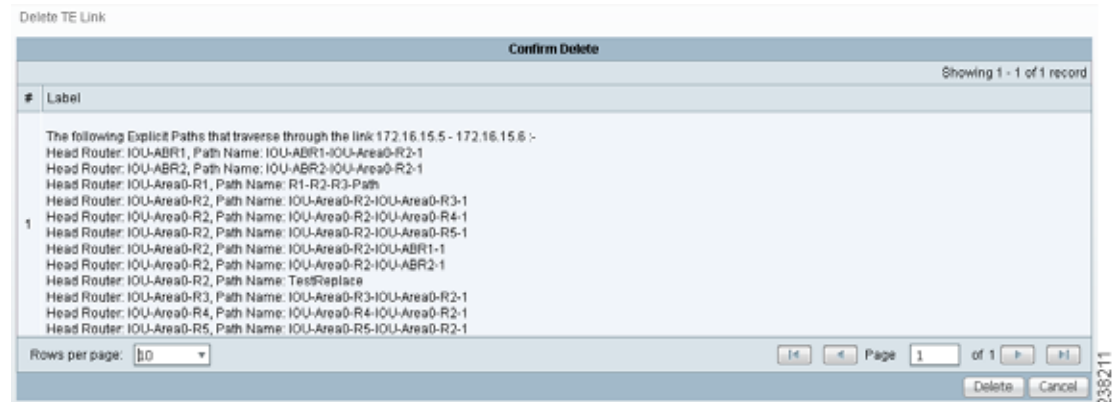


## Deleting Link Without Associated TE Object

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 9-25](#) will be displayed.

**Figure 9-25** 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 Provisioning web client. It gives a graphical representation of the various network elements, including devices, links, and tunnels. It also displays devices that Prime Provisioning 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, page 9-82](#)
  - [Displaying and Saving Layouts, page 9-84](#)
  - [Using Maps, page 9-85](#)
  - [Using Highlighting and Attributes, page 9-87](#)
  - [Using Algorithms, page 9-88.](#)

## 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**.

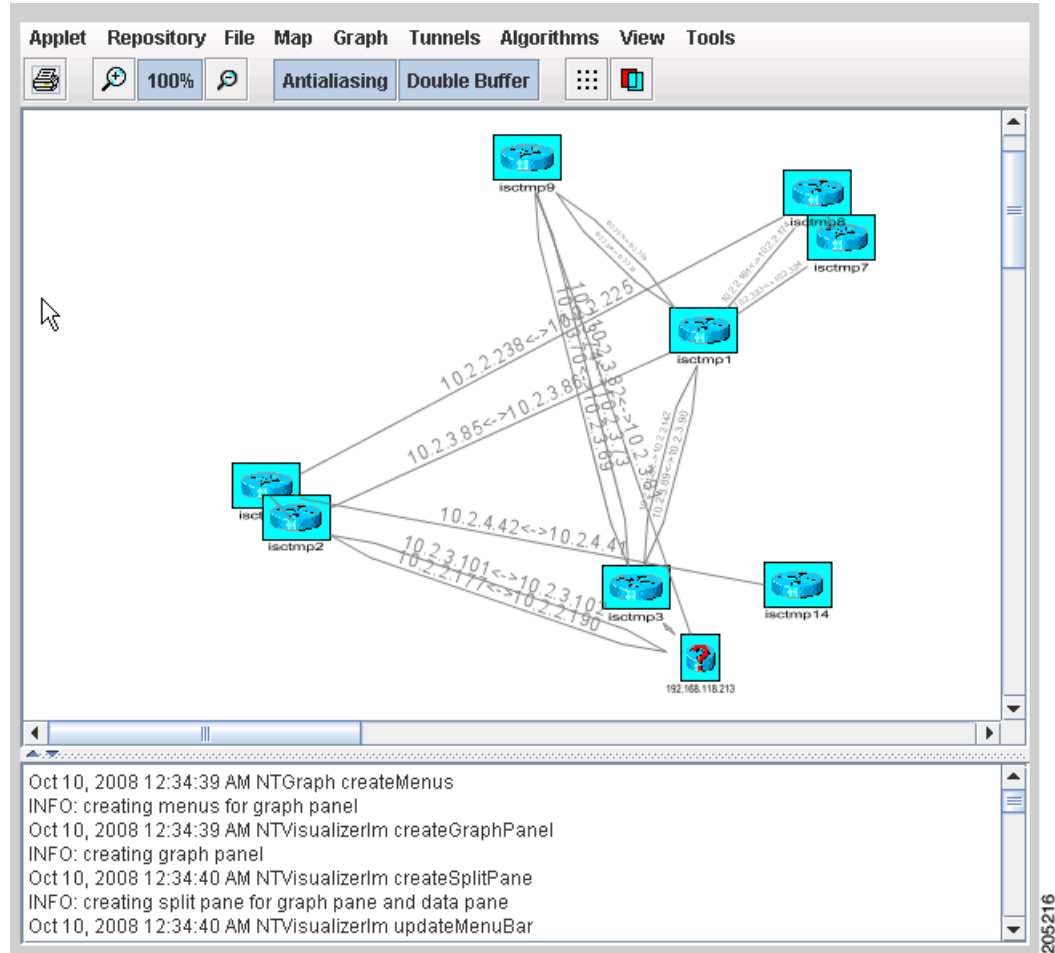
If the security certificate for the topology applet has not been accepted previously, you might get a security warning window.

**Step 3** Click **Yes** or **Always** to accept the authenticity of the security certificate.

The Topology Display applet window in [Figure 9-26](#) appears.

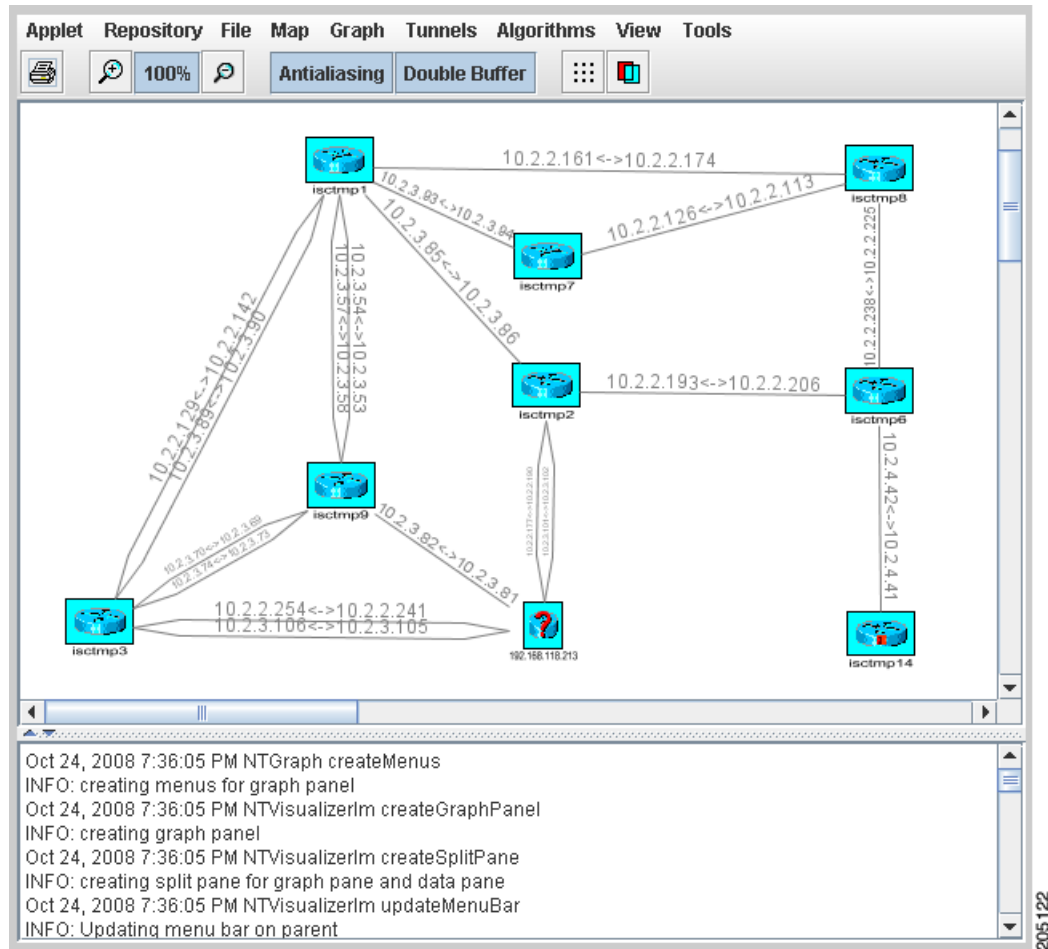


Figure 9-26 Topology Display Applet in Unordered State



After the nodes have been arranged to your liking, you might end up with a topology display similar to the one in [Figure 9-27](#).

Figure 9-27 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:

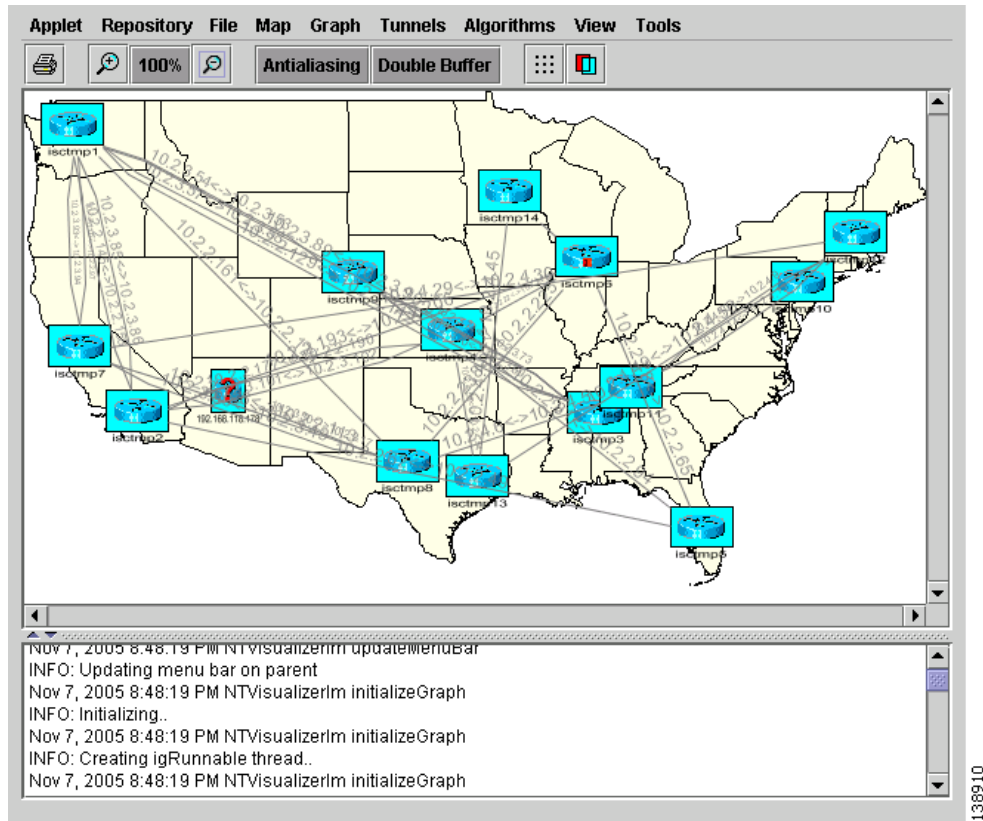
- 
- Step 1** Choose **Traffic Engineering > TE Topology**.
- Step 2** Start the **TM Topology Interface Applet**.
- If link and node data for your network is already in the repository, a Progress Report lists the various network elements as the corresponding data is loaded.
- Step 3** Select the **Map** 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 9-28](#) appears.

Figure 9-28 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 Provisioning 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.

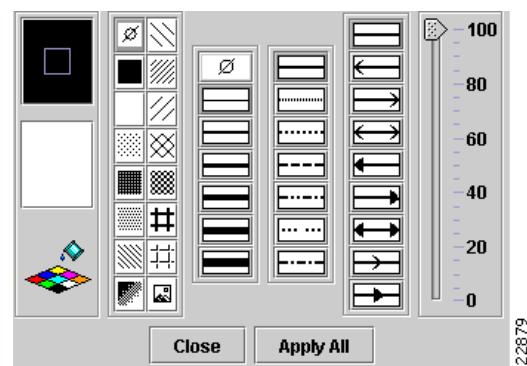
### Clear Highlighting

**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 9-29](#) appears.

**Figure 9-29** 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 Provisioning 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 Provisioning 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 Provisioning.

It includes the following sections:

- [Primary Tunnel Configlet \(IOS\), page 9-90](#)
- [Bandwidth Protection Backup Tunnel Configlet \(IOS\), page 9-91](#)
- [Connectivity Protection Backup Tunnel Configlet \(IOS\), page 9-92](#)
- [TE Traffic Admission Configlet Using CBTS \(IOS\), page 9-93](#)
- [TE Traffic Admission Configlet \(IOS\), page 9-94](#)
- [Primary Tunnel Configlet \(IOS XR\), page 9-95](#)
- [Bandwidth Protection Backup Tunnel Configlet \(IOS XR\), page 9-96](#)
- [Connectivity Protection Backup Tunnel Configlet \(IOS XR\), page 9-97](#)
- [TE Traffic Admission Configlet Using PBTS \(IOS XR\), page 9-98](#)
- [TE Traffic Admission Configlet \(IOS XR\), page 9-99.](#)

## 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                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                         |
|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <pre> ! <b>Explicit path:</b> ip explicit-path name isctmp2-isctmp8-1 enable   next-address 10.2.2.145   next-address 10.2.2.174 ! ! <b>Primary tunnel:</b> 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 ! </pre> | <p>Create an explicit path with the specified next addresses, which indicate the strict path that the tunnel traverses.<br/>This explicit path is used by the primary tunnel detailed above.</p> <p>Create a TE primary tunnel with the following attributes:</p> <ul style="list-style-type: none"> <li>- 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.</li> <li>- Destination 192.168.118.183</li> <li>- TE encapsulation</li> <li>- Setup and hold priorities both 0</li> <li>- Bandwidth global pool 10 kbps</li> <li>- Tunnel affinity 0x0</li> <li>- Explicit first path option</li> <li>- Dynamic second path option</li> </ul> |



## 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                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                        |
|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <pre>! Explicit path: ip explicit-path name isctmp5-isctmp4-1 enable   next-address 10.2.2.145   next-address 10.2.2.174 ! ! 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 ! interface POS0/1   mpls traffic-eng backup-path tunnel 1001 !</pre> | <p>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.</p> <p>Create a TE backup tunnel with the following attributes:</p> <ul style="list-style-type: none"> <li>- Destination 192.168.118.213</li> <li>- TE encapsulation</li> <li>- Protect subpool bandwidth of 30000 kbps</li> <li>- Setup and hold priorities both 0</li> <li>- Tunnel affinity 0x0</li> <li>- Explicit first path option</li> </ul> <p>Backup tunnel 1001 protects interface POS0/1</p> |

## 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                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                 |
|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <pre> ! <b>Explicit path:</b> ip explicit-path name L47-excl enable   exclude-address 192.168.1.18 ! ! ! <b>Backup tunnel:</b> 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 ! interface ATM4/0.1 point-to-point   mpls traffic-eng backup-path Tunnel1000 </pre> | <p>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.</p> <p>Create a TE backup tunnel with the following attributes:</p> <ul style="list-style-type: none"> <li>- Destination 10.52.96.38</li> <li>- TE encapsulation</li> <li>- Setup and hold priorities both 0</li> <li>- Backup tunnel does not reserve any bandwidth</li> <li>- Explicit first path option</li> <li>- Tunnel affinity 0x0</li> <li>- Unlimited backup bandwidth for protecting sub pool</li> </ul> <p>Set up backup path on ATM interface.</p> |

## 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                                                                                                                                                                                                          |
|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <pre>! <b>TE Traffic Admission using CBTS:</b> interface Tunnel1000   tunnel mpls traffic-eng exp 1 2 3 ! ! <b>Static route:</b> ip route 192.168.118.189 255.255.255.255 Tunnel1000</pre> | <p>Class-based tunnel selection where traffic with EXP bit 1, 2, or 3 are selected</p> <p>Create a static route, which admits all traffic destined for 192.168.118.189 into the above-configured Tunnel 1000.</p> |

The above is then deployed to an already existing primary tunnel such as the [Primary Tunnel Configlet \(IOS\)](#), page 9-90.

## 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                                                    |
|----------------------------------------------------------------------------------------------------------------------------------------------------------------|-------------------------------------------------------------|
| <pre>! TE Traffic Admission: interface Tunnel1000   tunnel mpls traffic-eng autoroute   announce   tunnel mpls traffic-eng autoroute metric   relative 0</pre> | <p>Autoroute announce with relative metric, 0 (default)</p> |

The above is then deployed to an already existing primary tunnel such as the [Primary Tunnel Configlet \(IOS\)](#), page 9-90.

## 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                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                      |
|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <pre> ! <b>Explicit path:</b> 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 ! ! <b>Primary tunnel:</b> 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 ! mpls ldp   interface tunnel-te 133 ! </pre> | <p>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.</p> <p>Create a TE primary tunnel with the following attributes:</p> <ul style="list-style-type: none"> <li>- Destination 192.168.118.214</li> <li>- TE encapsulation</li> <li>- Setup priority 0</li> <li>- Hold priority 0</li> <li>- Reserve 13 kbps from global pool</li> <li>- Tunnel affinity 0x0</li> <li>- Explicit first path option</li> <li>- Dynamic second path option</li> <li>- Enable FRR for the tunnel</li> </ul> <p>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</p> |

## 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                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                          |
|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <pre> ! <b>Explicit path:</b> 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 ! ! <b>Backup tunnel:</b> 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 </pre> | <p>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.</p> <p>Create a TE backup tunnel with the following attributes:</p> <ul style="list-style-type: none"> <li>- Destination 10.163.24.131</li> <li>- TE encapsulation</li> <li>- Protect any pool bw of 9600000 kbps</li> <li>- Setup and hold priority of 0</li> <li>- Tunnel affinity 0x0</li> <li>- Explicit first path option</li> </ul> |

## 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                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     |
|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <pre> ! <b>Explicit path:</b> explicit-path name L96-excl   index 1 exclude-address ipv4 unicast   192.168.1.42 ! ! ! <b>Backup tunnel:</b> 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 ! </pre> | <p>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.</p> <p>Create a TE backup tunnel with the following attributes:</p> <ul style="list-style-type: none"> <li>- Destination 10.52.96.37</li> <li>- TE encapsulation</li> <li>- Setup priority 0</li> <li>- Hold priority 0</li> <li>- Explicit first path option</li> <li>- Tunnel affinity 0x0</li> <li>- An unlimited sub pool acts as backup bandwidth</li> </ul> <p>Tunnel 1000 protects interface POS0/1/0/2</p> |

## 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                                    |
|----------------------------------------------------------------------------------------------------------------------------------------------|---------------------------------------------|
| <pre>! TE Traffic Admission using PBTS: interface tunnel-te133   autoroute announce   autoroute metric absolute 100   policy-class 2 !</pre> | 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 9-95.



## 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                                                               |
|-------------------------------------------------------------------------------------------------------------------------------------------|------------------------------------------------------------------------|
| <pre>! TE Traffic Admission Using Static Route: router static   address-family ipv4 unicast     1.2.3.4/32 tunnel-te 1000 123   ! !</pre> | 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 9-95.

## Warnings and Violations

This section lists warnings and violations that might be invoked when using the planning tools in Prime Provisioning (computation engine).

Warnings and violations are tied in with the planning tools (see the Planning Tools section in the [Traffic Engineering Management Concepts](#), page 9-112). 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 9-59).

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](#), page 9-100
- [Violations](#), page 9-101

## 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 warnings, 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 Primary Tunnel, 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 Provisioning.

## 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 Provisioning.

For instructions on how to import tunnels into Prime Provisioning, see [Import Primary Tunnel, page 9-50](#).

This section includes the following:

- [DTD File, page 9-109](#)
- [Example, page 9-112](#)

## DTD File

This is the DTD file provided with Prime Provisioning.

```
<?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 Bandwidth

- #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 9-109](#). 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>
```

## Traffic Engineering Management Concepts

This chapter includes an overview of Cisco Prime Provisioning and of some of the concepts used in this guide. This chapter includes the following sections:

- [Prime Provisioning TEM Overview, page 9-113](#)
- [Features in Prime Provisioning, page 9-113](#)
- [Prime Provisioning TEM Basics, page 9-113](#)
  - [Managed/Unmanaged Primary Tunnels, page 9-113](#)
  - [Conformant/Non-Conformant Tunnels, page 9-114](#)
  - [Multiple Concurrent Users, page 9-115](#)
  - [Multiple OSPF Areas, page 9-116](#)
  - [Bandwidth Pools, page 9-117](#)
  - [Planning Tools, page 9-118](#)
  - [Connectivity Protection \(CSPF\) Backup Tunnels, page 9-119](#)
  - [Class-Based Tunnel Selection, page 9-119](#)

- [Policy-Based Tunnel Selection, page 9-120.](#)

## Prime Provisioning TEM Overview

TEM is the Traffic Engineering Management module of Prime Provisioning. It is a tool for managing Multiprotocol Label Switching Traffic Engineering (MPLS TE) primary tunnels and backup tunnels for the purpose of offering traffic Service Level Agreement (SLA) guarantees. It provides bandwidth protection management, network discovery, and support for configuring MPLS TE. It includes a number of powerful planning tools, including a sophisticated primary path calculation tool and backup tunnel calculation for element protection.

MPLS TE mechanisms are provided to support requirements for predictability, traffic flow matched to QoS requirements, and Fast Restoration with Guaranteed Bandwidth, ensuring that strict SLA performance criteria (availability, delay, jitter) are met.

## Features in Prime Provisioning

Prime Provisioning adds a range of MPLS TE primary tunnel management features:

- Tunnel Audit—finding inconsistencies after making tunnel modifications
- Tunnel Admission—admitting new tunnels onto the network
- Tunnel Repair—fixing tunnel inconsistencies after network and service changes
- Network Grooming—optimizing global network utilization.

In addition, Prime Provisioning offers interaction and integration with Prime Provisioning features:

- Service activation focus
- Integration with other Prime Provisioning modules
- Data Persistence
- Logging of user intent
- Service state management
- Service auditing
- Web-based GUI
- Role-Based Access Control (RBAC).

## Prime Provisioning TEM Basics

To understand how Prime Provisioning works, you need to first know certain key concepts.

### Managed/Unmanaged Primary Tunnels

In Prime Provisioning, the concept of managed tunnels is at the center of TE planning activities.

It is important to understand the differences:

- Managed TE tunnels:
  - (setup/hold) priority zero

- non-zero RSVP bandwidth
- explicit first path option
- auto bandwidth must have a max value
- Unmanaged tunnels: All other tunnels.

In the Prime Provisioning Graphical User Interface (GUI), there is a separate entry point for dealing with managed and unmanaged tunnels.

## Conformant/Non-Conformant Tunnels

Understanding the concepts of conformant and non-conformant tunnels is key to making the most efficient use of Prime Provisioning.

Prime Provisioning only allows the creation of conformant tunnels. Non-conformant tunnels can be introduced through the TE Discovery process (see [TE Network Discovery](#), page 9-11 of the User Guide).

### Defining Conformant/Non-Conformant Tunnels

In the Prime Provisioning design, a sharp distinction has been made between conformant and non-conformant tunnels:

- Conformant tunnel—A well-behaved tunnel that meets Prime Provisioning's TE management paradigm (described below). A managed tunnel can only be a conformant tunnel. A non-zero priority unmanaged tunnel would also be a conformant tunnel. However, a conformant tunnel is not necessarily a managed tunnel.

A connectivity protection tunnel is marked Conformant = true if it has zero tunnel bandwidth, unlimited backup bandwidth, and an 'exclude address' first path option. For the BW Protected setting, a tunnel should have a defined non-zero backup bandwidth, and a strict path option 1.

- Non-conformant tunnel—A TE tunnel, which might impact Prime Provisioning's ability to meet bandwidth guarantees. This could be due to unknown bandwidth requirements such as no max bandwidth configured for auto-bandwidth, potential for pre-emption, dynamic paths, etc. A zero priority unmanaged tunnel would also be a non-conformant tunnel.

The following are examples of non-conformant tunnels:

- a tunnel with zero setup and hold priority, an explicit first path option, but with zero bandwidth;
- a tunnel with zero setup and hold priority, a non zero bandwidth, but with a dynamic first path option;
- a tunnel with zero setup and hold priority, an explicit path option of 1 and an auto bandwidth without a maximum defined.;
- a connectivity protection tunnel marked Conformant = false is reserved for backup tunnels, which have neither zero tunnel bandwidth, unlimited backup bandwidth, or an 'exclude address' first path option.

Why are the above tunnels non-conformant? Because Prime Provisioning attempts to manage all tunnels with zero setup and hold priority, to ensure the links they pass through all have sufficient bandwidth, are affinity consistent, and do not break delay or FRR constraints defined in the TE policy.

But if the tunnel's path is dynamic or the amount of bandwidth it requires is undefined, Prime Provisioning does not have the information with which to manage the tunnel, so it marks it as non-conformant. All the non-conformant tunnels are displayed in the TE Unmanaged Primary Tunnels SR window.



## Managing Non-Conformant Tunnels

It is important to understand that non-conformant tunnels not only might cause the SLAs to be violated, they might also have an adverse effect on the managed tunnels (taking away bandwidth from them, for example).

However, when a non-conformant tunnel is discovered, a warning is logged. Prime Provisioning tracks non-conformant tunnels so that they can be decommissioned.

So conformant tunnels are preferred. They allow the system to offer bandwidth guarantees for managed tunnels. Unmanaged non-conformant tunnels might or might not provide the needed bandwidth and no bandwidth guarantees are given.

The action to take when you have non-conformant tunnels is either to change the setup and hold priorities to non-zero values (so they cannot preempt the managed tunnels) or migrate them to managed tunnels, allowing the tool to find a suitable explicit path.

## Multiple Concurrent Users

In previous releases TEM only supported a single GUI user. This release introduces support for multiple concurrent users, for all browsing, updating, and provisioning operations.

### Concurrent Use with Managed and Unmanaged Tunnels

To understand how the multiple user feature is implemented in TEM, it is important to understand the difference between a managed and an unmanaged tunnel. This is described in the section [Managed/Unmanaged Primary Tunnels, page 9-113](#).

There are important differences between how managed and unmanaged tunnels are handled when it comes to multiple user support:

- For managed tunnels, an SR encapsulates all managed tunnels. A SR operation might optimize all the objects within the snapshot following path computations performed by the Router Generator server.
- For unmanaged tunnels, an SR is defined as a tunnel-head end router. Thus, with unmanaged tunnels there are certain restrictions. For example, two users cannot concurrently provision on the same device.
- TEM prevents Unmanaged Tunnel SRs from provisioning concurrently on the same device but supports Unmanaged Tunnel SRs provisioning concurrently on different devices.
- All managed tunnels are contained within a shared Managed TE Tunnel SR for each TE Provider. For unmanaged tunnels, a distinct Unmanaged TE Tunnel Service Request is created per head device. TEM supports multiple SRs per TE Provider.

Multiple TEM users can browse and provision in TEM. Up to 20 concurrent users are supported, of which up to seven can perform provisioning tasks.

Previously all primary tunnels, managed and unmanaged were in a single TE tunnel SR per TE provider. Now, to facilitate multiple simultaneous changes to managed tunnels, the TE Tunnel SR has been split into one managed tunnel SR per TE provider and one unmanaged tunnel SR per head TE router.

Parallel provisioning is not possible on the same SR, but because SRs exist at router level for unmanaged tunnels, unmanaged tunnels can be provisioned on separate routers at the same time.

## Locking Mechanism

When an unmanaged tunnel is provisioned, the head TE router of the tunnel is locked. This can be seen on the TE Nodes window in the System Lock Status column. The locking prevents any other user from deploying any kind of tunnel to that router until the provisioning task completes and the TE router is unlocked.

The locking mechanism also applies to other Prime Provisioning features, such as backup tunnels, resource SRs, link deletion, and TE traffic admission. Resource SRs include deleting/editing explicit paths, deleting protected elements, deleting/editing SRLG's, etc.

In the case of link deletion, a level of intelligence is built in. When there are no more tunnels to be rerouted or deleted by the user or Prime Provisioning and left with TE associated objects alone, a user intervention will be required to carry out link deletion. As part of this deletion, if there are any backup tunnels protecting any interfaces that have been selected for deletion, the locking mechanism will be in place during deployment of backup tunnels. For further information about deleting TE links, see [Deleting TE Links, page 9-25](#) of the User Guide.

Some of the potential errors you might encounter are described in [Locking Operation Errors, page 9-79](#) of the User Guide.

When a managed primary tunnel or a backup tunnel is provisioned, the TE provider it is associated with is locked. This can be seen on the TE Provider window in the System Lock Status column. A lock at the TE provider level prevents another user from making any tunnel change on this TE provider, irrespective of which TE router the tunnel starts at.

The reason why the locking mechanism of managed tunnels and backup tunnels is different from that of unmanaged tunnel is that the managed tunnels and backup tunnels use a path generation algorithm to find an optimal route for the tunnel that fulfills all constraints, and this algorithm needs a stable global view of the TE topology and all the tunnels in it on which to base its routing decisions. This can only be achieved by allowing only one user to make changes at one time.

For more information about how to manage Prime Provisioning locking mechanism, see [Managing the Locking Mechanism, page 9-78](#) of the User Guide.

## Multiple OSPF Areas

Prime Provisioning supports the discovery, management, and provisioning of TE Tunnels within multiple Open Shortest Path First (OSPF) areas.

Prime Provisioning only manages primary and backup TE tunnels within the scope of an OSPF area. There is no support for the discovery and creation of inter-OSPF areas.

In Prime Provisioning, an OSPF area is represented by a TE provider. After an area is assigned to a TE provider, it might not be changed. Multiple TE providers can be associated with one Prime Provisioning provider.

## Devices Suitable for TE Discovery

In a network with multiple OSPF areas, where each OSPF area is represented by a TE provider, any router in an OSPF area can be used for TE Discovery. Using multiple TE providers (multiple OSPF areas) under one provider allows the provisioning of inter-area L3VPN.



### Note

Prime Provisioning will not discover or provision inter-area TE tunnels (those with a head router in one area and a tail router in a different area).

To discover a multi area network, you have to discover each area in turn using TE Discovery (see [TE Network Discovery, page 9-11](#) of the User Guide). The seed node can be any device within that area, including an Area Border Router (ABR).

### TE Discovery and the TE Area Identifier

TE Discovery is associated with a TE provider and each TE provider is assigned an area. The area is assigned during the process of creating the TE Provider (see [Creating a TE Provider, page 9-8](#) of the User Guide) and can be a simple integer value or dotted decimal notation, Area 0.6.0.0 for example.

TE provider objects are aware of which area they are responsible for, either specified on creation or automatically populated during discovery, and will accommodate conversion between Dot notation and Decimal notation, defaulting to the notation used in the network.

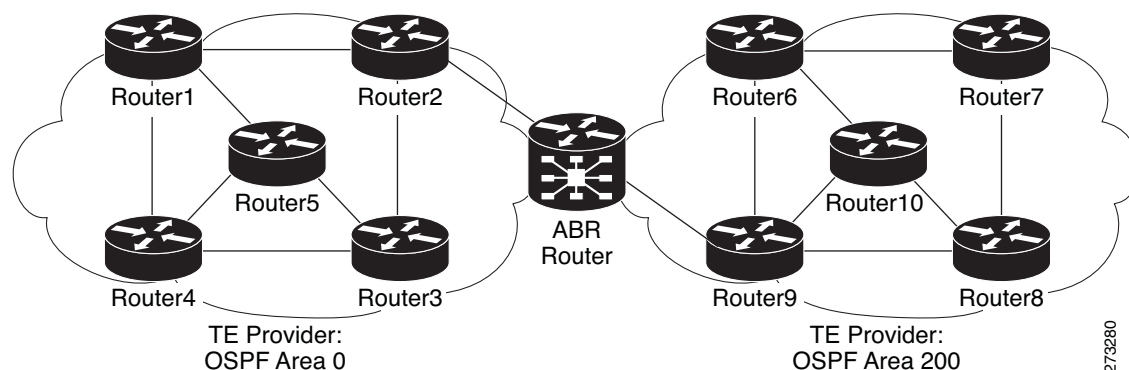
When discovery is run against an area with a selected TE provider, all tunnels and explicit paths associated with that area will be imported into the Prime Provisioning database. The steps for performing a per area discovery are documented in the [Managing Per Area Discovery, page 9-16](#) of the User Guide.

### Example of Multiple OSPF Area Network

TE routers within a TE provider can be assigned to different regions, for example on a geographical basis, so that devices are grouped in regions in a logical way. Also, Prime Provisioning allows you to filter by region. Assigning objects to specific regions is a manual task that is carried out after discovery from **Inventory > Provider Devices**. Here the region of any PE device can be changed via the Select Region pop-up window.

In the following example [Figure 9-30](#), two TE providers are each responsible for one OSPF area that is created and visualized under one Prime Provisioning provider.

**Figure 9-30 Multiple OSPF Areas Network Diagram**



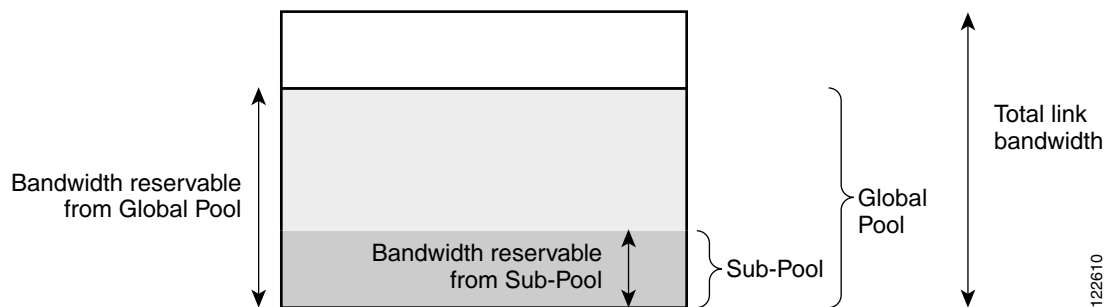
For instructions on how to manage TE providers, see [Creating a TE Provider, page 9-8](#) of the User Guide.

### Bandwidth Pools

The bandwidth of each TE enabled interface is assigned a number of nested bandwidth pools. Currently, IOS supports two, namely Global Pool and Sub Pool.

For a better understanding of bandwidth pools, see [Figure 9-31](#).

Figure 9-31 Bandwidth Pools



As Figure 9-31 illustrates, Sub Pool is nested inside Global Pool. Thus, if a primary tunnel reserves bandwidth from the Sub Pool, it will also reserve the same bandwidth from the Global Pool.

Bandwidth reservations (primary tunnels) from the Sub Pool must not exceed, in total, the Sub Pool size. Likewise, bandwidth reservations from the Global Pool must not exceed, in total, the Global Pool size.

## Planning Tools

They are intended for evaluating planned improvements to a traffic-engineered network based on What-If scenarios.

The planning tools include the following features:

- Primary planning tools:
  - Tunnel Audit—Audits for inconsistencies in primary placement on the existing network with or without proposed tunnel or resource changes.
  - Tunnel Placement—Usually for new tunnels. Tunnel Placement can generate a new route. It can be used for a tunnel that did not have a path before and needs to be placed.
  - Tunnel Repair—Logically performed after Tunnel Audit (if something is wrong). Tunnel Repair has rerouting capabilities and can be used to move tunnels.
  - Grooming—An optimization tool that works on the whole network. It is only available when no tunnel attributes have been changed.
- Protection planning tools:
  - Audit SR—Audits protection for manually added, modified, and deleted backup tunnels before they are deployed.
  - Compute Backup—Automatically calculates the optimal backup tunnel for selected network elements.
  - Audit Protection—Audits protection of the selected elements against the existing backup tunnels.

The planning tools are fully integrated within Prime Provisioning and are available from various locations within the GUI:

- TE Protected Elements (Compute Backup and Audit Protection)
- Create Managed TE Tunnel (Tunnel Audit, Tunnel Placement, Tunnel Repair, Grooming)
- Create TE Backup Tunnel (Audit SR).

## Connectivity Protection (CSPF) Backup Tunnels

In addition to the bandwidth-protected backup tunnels created by TEM, you can create a set of CSPF-routed backup tunnels within Prime Provisioning. These CSPF-routed backup tunnels are managed from the TE Protection SR window.

A connectivity protection backup tunnel uses an “exclude-address” explicit path. This explicit path is created in the TE Explicit Path List window. An exclude address path is different from a strict path in that instead of listing the hops the path should use, it lists the hops the path should avoid. The CSPF algorithm on the router will make the decision as to which precise path to use, but it will be constrained to not be able to use the hops in the exclude address path configuration. This sort of path is particularly useful for backup tunnels, as the interfaces the exclude address path should avoid can be the interfaces that the backup tunnel is protecting.

In Prime Provisioning, these backup tunnels are configured with unlimited backup bandwidth. Unlimited means no bandwidth is guaranteed, but as much as is available at the time of the failure will be used. So in effect the bandwidth protection is best effort but the connectivity is guaranteed. Connectivity protection backup tunnels can be used in addition to or instead of bandwidth protection backup tunnels.

Differences between bandwidth protection and connectivity protection backup tunnels:

- A bandwidth protection backup tunnel has a strict explicit path as its first path option, whilst a connectivity protection tunnel has an exclude address explicit path as its first path option.
- A bandwidth protection backup tunnel has a defined backup bandwidth whilst a connectivity tunnel has unlimited backup bandwidth on a best effort basis.
- A bandwidth protection backup tunnel is passed to the Route Generator algorithm which generates optimal backup tunnels and verifies existing tunnels fully protect the elements, whereas connectivity protection tunnels are not passed to the algorithm and it is up to you to ensure they are fulfilling their purpose.

## Class-Based Tunnel Selection

Multi-Protocol Label Switching Traffic Engineering Class-Based Tunnel Selection (CBTS) enables you to dynamically route and forward traffic with different class of service (CoS) values onto different TE tunnels between the same tunnel head end and the same tail end. The packet’s CoS values are located in the EXP bits. There are 8 EXP bits, numbered 0 to 7.

The set of TE (or DS-TE) tunnels from the same head end to the same tail end can be configured to carry different CoS values. After configuration, CBTS dynamically routes and forwards each packet into the tunnel that:

- is selected for traffic admission using the standard autoroute or static route mechanisms, and
- has EXP bits matching that of the packet.

Thus CBTS is not a form of traffic admission to TE tunnels directly, it is rather an additional criteria that traffic must satisfy before being admitted to tunnels via the autoroute or static route mechanisms that TEM supports.

Because CBTS offers dynamic routing over DS-TE tunnels and requires minimum configuration, it greatly eases deployment of DS-TE in large-scale networks. CBTS can distribute all CoS values onto many different tunnels.

The CBTS feature has the following restrictions:

- For a given destination, all CoS values are carried in tunnels terminating at the same tail end. Either all CoS values are carried in tunnels or no values are carried in tunnels. In other words, for a given destination, you cannot map some CoS values in a DS-TE tunnel and other CoS values in a Shortest Path First (SPF) Label Distribution Protocol (LDP) or SPF IP path.
- CBTS does not allow load-balancing of a given EXP value in multiple tunnels. If two or more tunnels are configured to carry a given experimental (EXP) value, CBTS picks one of these tunnels to carry this EXP value.
- The operation of CBTS is not supported with Any Transport over MPLS (AToM), MPLS TE Automesh, or label-controlled (LC)-ATM.

When traffic admission to tunnels is achieved using global static routes, and when there is more than one tunnel to a given destination with the same administrative weight, the CBTS attribute acts as a tiebreaker in selecting the right tunnel. (See above discussion of load-balancing with CBTS.)

## Policy-Based Tunnel Selection

Multi-Protocol Label Switching Traffic Engineering Policy-Based Tunnel Selection (PBTS) enables you to dynamically route and forward traffic based on a policy onto different TE tunnels between the same tunnel head end and the same tail end. The routing algorithm is performed on the headend router's ingress interface prior to forwarding lookup.

In the Prime Provisioning implementation of PBTS, traffic is directed into specific TE tunnels using the interface command `policy-class`. Whereas CBTS is aimed at IOS devices, PBTS is strictly designed for IOS XR devices.

Like CBTS, PBTS is not a form of traffic admission to TE tunnels directly, but rather an additional criteria that traffic must satisfy before being admitted to tunnels via the autoroute or static route mechanisms that TEM supports.



### Note

---

Prime Provisioning itself does not provision the policy class, it merely associates a tunnel with an existing policy class. This is done by specifying the `policy-class` attribute in the range 1 to 7.

---

For more information on CBTS, see [Class-Based Tunnel Selection, page 9-119](#).

For general information on PBTS and IOS XR, see

[http://www.cisco.com/en/US/docs/ios\\_xr\\_sw/iosxr\\_r3.7/mpls/configuration/guide/gc37te.html#wp1325561](http://www.cisco.com/en/US/docs/ios_xr_sw/iosxr_r3.7/mpls/configuration/guide/gc37te.html#wp1325561).



## Managing Service Requests

---

This chapter describes how to manage Prime Provisioning service requests through the Service Request Manager window. It contains the following sections:

- [Accessing the Service Request Manager Window, page 10-1](#)
- [Viewing Service Request Details, page 10-2](#)
- [Viewing the Status of Service Requests, page 10-7](#)
- [Previewing Configlets for Deploy and Decommission, page 10-8](#)
- [Editing Service Requests, page 10-8](#)
- [Deploying Service Requests, page 10-9](#)
- [Decommissioning Service Requests, page 10-12](#)
- [Deleting Service Requests, page 10-13](#)
- [Service Request States, page 10-14](#)

## Accessing the Service Request Manager Window

To manage service requests, choose **Operate > Service Requests > Service Request Manager**.

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 Provisioning.
- **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 11, “Managing Templates and Data Files.”](#)
- **State**—The transition state for the service request. See [Service Request States, page 10-14](#) 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 Provisioning 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 10-2](#).
- **Status**—View links and access any available logs for a selected service request. For more details, see [Viewing the Status of Service Requests, page 10-7](#).
- **Preview**—Preview configlets that will be sent to a device for a specific service request. For more details, see [Previewing Configlets for Deploy and Decommission, page 10-8](#).
- **Edit**—Edit a service request. For more details, see [Editing Service Requests, page 10-8](#).
- **Deploy**—Deploy a service request. For more details, see [Deploying Service Requests, page 10-9](#).
- **Decommission**—Decommission a service request. For more details, see [Decommissioning Service Requests, page 10-12](#).
- **Delete**—Delete a service request. For more details, see [Deleting Service Requests, page 10-13](#).

## 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 > History**—Service request history report.
  - **Details > Audit**—Not supported by Prime Provisioning.
  - **Details > Configlets**—View the Prime Provisioning generated configlet for the service request.
- 

The following sections describe the history, audit, and configlet details for a service request.



## Viewing Service Request History Information

To view history information about the service request, perform the following steps.

- 
- Step 1** Click **History** in 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 Provisioning 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 Provisioning 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 Provisioning, a configuration audit occurs. When a configuration audit occurs, Prime Provisioning 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 Provisioning does not recognize commands in the configuration file. The default behavior in Prime Provisioning is to skip unrecognized commands in the configuration file during the configuration audit. Such unrecognized commands might have 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 Provisioning will skip the initial command and continue to parse the subcommands in the block. This might lead Prime Provisioning 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 Provisioning Task Manager. For information on how to create a task to manually schedule a configuration audit, see [This chapter contains the following sections](#), page 12-1.

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 Provisioning 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 Provisioning Task Manager. For information on how to create a task to manually schedule a functional audit, see [This chapter contains the following sections:](#), page 12-1.

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 Provisioning 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 10-5](#).

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 Provisioning 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 for Deploy and Decommission

The preview configlet operation allows you to preview the configlet(s) that are sent to a device (or devices) for a selected service request before the device is actually provisioned. This allows you to ensure that the service request is generating the expected configlet(s), including relevant templates that may be applied.

Note the following caveats:

- The preview deployed configlet feature is available to service requests in all states except Delete.
- 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 the **Preview** drop-down list.
- Step 3** To preview the configlet for deploy, select **Preview Deploy**.
- Step 4** To preview the configlet for decommissioning, select **Preview Decommissioning**.

The Configlet Preview window displays the generated configlets for each device in the service request. This operation may take some time, as the configlet(s) must be uploaded from the device.



**Note** Preview Decommission does not support the display of configlets derived from Templates. When you request Preview Decommissioning for a service Request associated with Templates, it will throw the exception/error, “Preview Decommission is not supported for Service Requests associated with templates.”

---

- Step 5** After you review the configlets, click **OK** to return to the Service Request Manager window.
- 

## 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 10-9](#). After deployment, look for the service request state to go to Deployed to indicate a successful deployment.

## Deploying Service Requests

To apply either policies or device changes to network devices, you must first deploy the service request. When you deploy a service request, Prime Provisioning compares the device information in the Repository (the Prime Provisioning database) with the current device configuration and generates a configlet.

While deploying multiple service requests, the below mentioned points can result in deployment failure.

- The type of service request being performed, and the resources that would be required for these services.
- The device(s) that are being interacted with and if these devices are used by other Service Requests being requested at the same time.
- The current running-configuration on the device.
- The number of services that are currently present and modelled on each device.
- Network latency between the NMS and the managed devices

For a successful deployment, it is advised to have 2 to 3 seconds time gap between each service request deployment.

## 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 the following deployment options:

- **Deploy Now**—This performs an immediate Force Deploy of the service request.

- **Deploy Later**—This brings up the scheduler, which allows you to schedule when you want to deploy the service request. This performs a Force Deploy of the service request.
- **Simulated Deploy Now**—See [Simulated Deployment of Service Requests, page 10-11](#), for information on this choice.

If you choose **Deploy Later**, the Deploy Service Request dialog box appears.

**Step 4** Complete the fields in this dialog box to schedule the service requested as needed.

**Step 5** When satisfied with the schedule settings, click **Save**.

You return to the Service Request Manager window.

**Step 6** Once the service request has been deployed, 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 10-7](#).

---

## 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 Provisioning.

**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 Provisioning 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 Now
  - Deploy Later
  - Simulate Deploy Now
- Step 4** Choose **Simulate Deploy Now**.  
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 10-7](#).

## Echo Mode

This explanation of Echo mode is specified in the following subsections:

- [What is Echo Mode?, page 10-12](#)
- [Who Should Use Echo Mode and When Should It Be Used?, page 10-12](#)

- [How Should You Use Echo Mode?](#), page 10-12

## What is Echo Mode?

Echo mode is a setting in Prime Provisioning that is accessible through the Prime Provisioning configuration window. Echo mode affects service provisioning. When you set Prime Provisioning to run in echo mode, Prime Provisioning performs 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.

## Who Should Use Echo Mode and When Should It Be Used?

In a production environment, echo mode can be used to perform service provisioning on devices that are either temporarily offline or not yet commissioned. The service provisioning only occurs within the Prime Provisioning Repository. When these devices become active, you can force the deployment of the previously provisioned services and Prime Provisioning downloads the configurations to the devices.

Echo mode is a global configuration setting that affects the Service Provisioning for *all* users. Therefore, echo mode should be used with care. To enable echo mode, set the Dynamic Component Properties Library (DCPL) **GTL/echo-mode** to **true** (**Administration > Control Center > Hosts**, as explained in Appendix C, Property Settings of the *Cisco Prime Provisioning 7.2 User Guide*). When echo mode is enabled, no attempt is made to contact any devices and no attempt is made to audit the Service Request. This affects all Service Requests during the time period when echo mode is enabled.

## How Should You Use Echo Mode?

Because echo mode affects all of Prime Provisioning's provisioning, be sure that all provisioning requests that require device access are complete before turning on echo mode.

Turn on echo mode, as explained in the [“Who Should Use Echo Mode and When Should It Be Used?” section on page 10-12](#).

Configure your Service Request as normal for the device that is not commissioned or is offline. Save and deploy the Service Request. No attempt is made to contact the device or audit the Service Request. The Service Request transitions into the Deployed state.

Now, you can disable echo mode, by changing the **GTL/echo-mode** property to **false** (see the [“Who Should Use Echo Mode and When Should It Be Used?” section on page 10-12](#)). From this point forward, all provisioning requests contact the devices and all provisioning requests are audited. You can now safely resume provisioning for all users.

After the device has been commissioned or brought back online, Force deploy the provisioning request for this device (see Chapter 3 in the *Cisco Prime Provisioning 7.2 User Guide*). This forces the provisioning request to go through the provisioning cycle and deploy the configlet onto the device.

---

## 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.
- Step 3** Click the **Decommission** drop-down list.  
Choose one of the following options:
- **Decommission Now**—This performs an immediate decommissioning of the service request.  
A popup window asks you to confirm the action. If you choose ‘yes’, the service request’s OpType is set to Delete and the deployment of this service request is performed immediately. If you choose ‘no’, the request is canceled.
  - **Decommission Later**—This brings up the scheduler which allows you to be schedule the decommissioning at a specified date and time by specifying the number of times the service request is to be decommissioned.
- Step 4** Complete the fields in this dialog box to schedule the service requested as needed.
- Step 5** Click **Save**.  
The Service Request Manager window is displayed.
- Step 6** Once the service request has been decommissioned, a pop-up window displays the Status of the decommissioning in the lower right corner of your screen.
- Step 7** After deployment, verify that the State of the service request state is changed to Closed. This indicates 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.

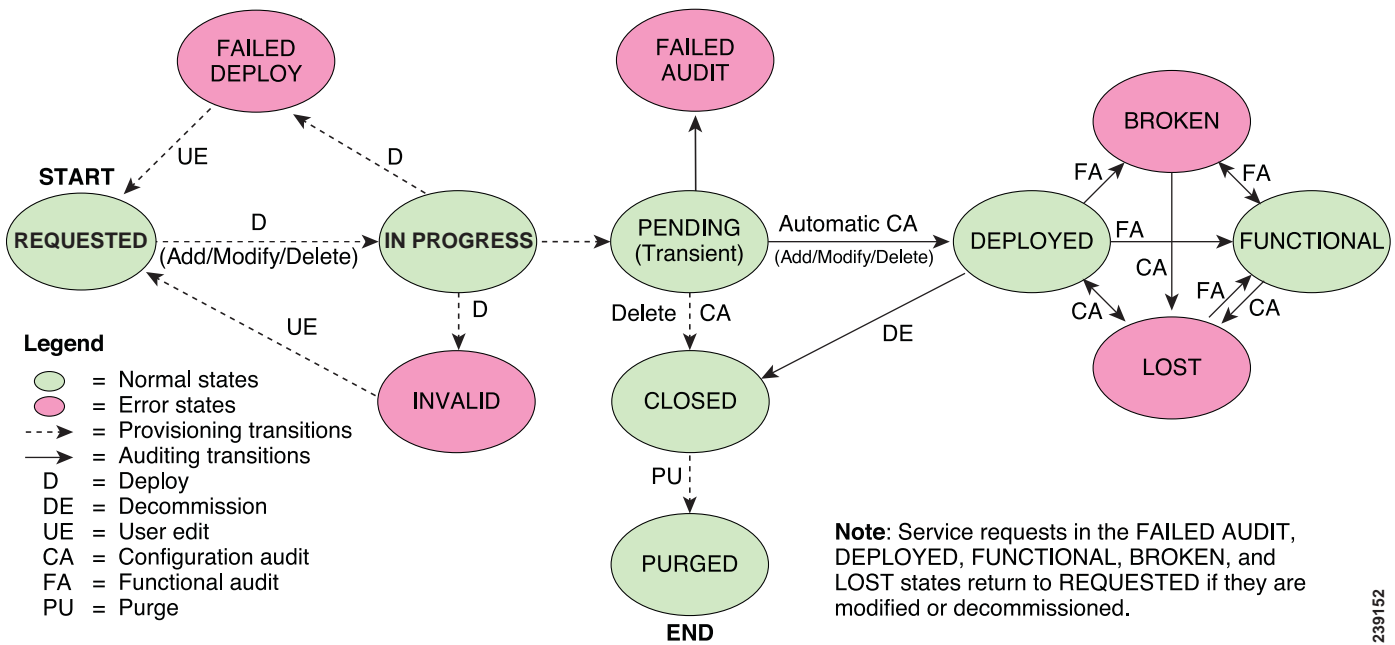
## Service Request States

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 Provisioning compares the device information in the Repository (the Prime Provisioning 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 Provisioning 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 10-1, “Service Requests States Transition Diagram,” shows a high-level diagram of the relationships and movement among Prime Provisioning service request states.

Figure 10-1 Service Requests States Transition Diagram



239152

Table 10-1, “Summary of Prime Provisioning Service Request States,” describes the functions of each Prime Provisioning service request state. They are listed in alphabetical order.

**Table 10-1 Summary of Prime Provisioning Service Request States**

| Service Request Type                                | Description                                                                                                                                                                                                                                                                                                                                                                                                                                   |
|-----------------------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Broken</b><br>(valid only for MPLS services)     | <p>The router is correctly configured but the service is unavailable (due to a broken cable or Layer 2 problem, for example).</p> <p>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.</p>                                                                                                                                           |
| <b>Closed</b>                                       | <p>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 Provisioning does not remove a service request from the database to allow for extended auditing. Only a specific administrator delete action results in service requests being removed.</p> |
| <b>Deployed</b>                                     | <p>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 Provisioning downloaded the configlets to the routers and the service request passed the audit process.</p>                            |
| <b>Failed Audit</b>                                 | <p>This state indicates that Prime Provisioning 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).</p>                    |
| <b>Failed Deploy</b>                                | <p>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).</p>                                                                                                                                                                      |
| <b>Functional</b><br>(valid only for MPLS services) | <p>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.</p>                                                                                                                                                               |
| <b>Invalid</b>                                      | <p>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.</p>                             |
| <b>Lost</b>                                         | <p>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.</p>                                                                             |

**Table 10-1** Summary of Prime Provisioning Service Request States (continued)

| Service Request Type | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                |
|----------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Pending</b>       | <p>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.</p> <p>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.</p> |
| <b>Requested</b>     | <p>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.</p>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  |
| <b>In Progress</b>   | <p>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.</p>                                                                                                                                                                                                                                                                                                               |

Table 10-2, “[User Operations on Prime Provisioning Service Requests](#),” describes user operations and their impact on Prime Provisioning service requests.

**Table 10-2** User Operations on Prime Provisioning Service Requests

| User Operations     | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                   |
|---------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Decommission</b> | <p>This user operation removes the service from all devices in the service request.</p>                                                                                                                                                                                                                                                                                                                                                                                       |
| <b>Force Deploy</b> | <p>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.</p>                                                                                                                                                                                        |
| <b>Force Delete</b> | <p>This user operation removes a service request from the database irrespective of its state. If you Force Delete a service request from the Prime Provisioning 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 Provisioning.</p> |
| <b>Delete</b>       | <p>When a service request is deleted, it is removed from the Prime Provisioning database.</p>                                                                                                                                                                                                                                                                                                                                                                                 |



# Managing Templates and Data Files

---

This chapter explains the use of templates and data files in Prime Provisioning. It contains the following sections:

- [Overview, page 11-1](#)
- [Basic Template and Data File Tasks, page 11-5](#)
- [Using Templates with Policies, page 11-20](#)
- [Using Templates with Service Requests, page 11-24](#)
- [Template Examples, page 11-31](#)
- [Summary of Repository Variables, page 11-33](#)
- [Importing and Exporting Templates, page 11-55](#)
- [Frequently Asked Questions, page 11-56](#)

## Overview

Templates provide a means to deploy commands and configurations not normally supported by Prime Provisioning 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 Provisioning. 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 Provisioning provides a way to integrate a template with Prime Provisioning 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 Provisioning 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 Provisioning configlet. Prime Provisioning downloads the combined Prime Provisioning 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 Provisioning, especially those that have an impact on working with policies and service requests.

### Template Attributes

The Prime Provisioning 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 Provisioning 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 Provisioning 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 Provisioning 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 Provisioning 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/data 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 Provisioning. You create both the positive and negate template. You can assign a positive template/data file to a policy. Prime Provisioning calls the appropriate negate template at the appropriate time, as the negate template has a direct relationship with the deploy template. Prime Provisioning 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 Provisioning Releases

Prime Provisioning maintains compatibility with the template mechanism in previous Prime Provisioning releases. Templates created in earlier versions of Prime Provisioning 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 Provisioning 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 Provisioning.

### 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 is 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 Provisioning repository variables for MPLS, L2VPN, VPLS, and FlexUNI/EVC. For a list of supported repository variables, see [Summary of Repository Variables](#), page 11-33.

### 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 a service has multiple lines, only one of which have been edited, etc. For documentation on DCPL properties related to templates, see [Appendix G, “Property Settings”](#) or see the instructions in the *Cisco Prime Provisioning 7.2 Administration Guide*.

### Importing and Exporting Templates

Prime Provisioning provides a mechanism to import and export templates and data files. See [Template Examples](#), page 11-31, 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 Provisioning.

### 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 11-20, 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 in service requests, see the section [Using Templates and Data Files in the Service Request Workflow, page 11-28](#).

## 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, page 11-5](#)
- [Creating Folders and Subfolders, page 11-6](#)
- [Copying Folders or Subfolders, page 11-6](#)
- [Creating Templates, page 11-7](#)
- [Creating Data Files, page 11-16](#)
- [Editing Templates and Data Files, page 11-18](#)
- [Deleting Templates and Data Files, page 11-19](#)
- [Listing Service Requests Associated with a Data File, page 11-20](#)
- [Listing Policies Associated with a Data File, page 11-20](#)

## 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 11-1](#).

**Figure 11-1** *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.

---

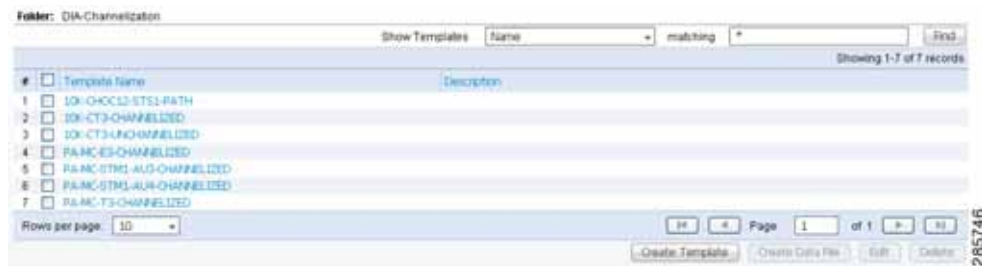
## Creating Templates

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 11-2](#).

**Figure 11-2** 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 11-5](#).
- Step 4** Click the **Create Template** button and you receive a window as shown in [Figure 11-3](#).

**Figure 11-3** *Template Editor*

Template Editor

**Template Information**

Template Name \* :

Description:

Body :

Has Negate Template:

Has User Reference :

Select Save Close

Note: \* - Required Field

285747

**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 11-9](#)
- [User Section, page 11-9](#)
- [Optional Attributes, page 11-10](#)
- [Sub-Template, page 11-12](#)
- [Variables, page 11-13](#)
- [Validate, page 11-16](#)

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 elsewhere in this guide).

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 11-4](#).

**Figure 11-4** *Negate Template Editor*

Negate Template Editor

Negate Template Editor for Template: /DIA-Channelization/10K-CHOC12-STS1-PATH

Description:

Body\*

Has User Section:

Select Save Cancel

Note: \* - Required Field

285748

- 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 Section

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 11-5](#).

Figure 11-5 User Reference Editor

- Step 2** In Figure 11-5, you can add information in the available fields, **Template** and **Body**.
- Step 3** When you click the **OK** button, the information updates in Figure 11-3. When you click **Cancel**, you return to Figure 11-3 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 Provisioning 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 11-6.

Figure 11-6 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 Provisioning 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 elsewhere in this guide).
- 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 11-7](#).



**Note** Before clicking the **Edit** button, you must check the check box for the one attribute set (row of information) in [Figure 11-6](#) that you want to edit. You cannot edit multiple rows at the same time.

**Figure 11-7** Optional Template Attributes Editor

**Step 5** In [Figure 11-7](#), 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 G](#), “Property Settings” or see the instructions in the *Cisco Prime Provisioning 7.2 Administration Guide*.)



**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 11-6](#) 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 11-8](#).

**Figure 11-8** 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 elsewhere in this guide), 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 information on using sub-templates, see [Associating Subtemplates During Service Provisioning, page 11-25](#).

## 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 11-9](#).

**Figure 11-9** *Template Variables*

| # | Variable                                     | Type   | Description |
|---|----------------------------------------------|--------|-------------|
| 1 | <input checked="" type="radio"/> cableLength | String |             |
| 2 | <input type="radio"/> clockSource            | String |             |
| 3 | <input type="radio"/> ctrlName               | String |             |
| 4 | <input type="radio"/> description            | String |             |
| 5 | <input type="radio"/> dsuBandwidth           | String |             |
| 6 | <input type="radio"/> dsuMode                | String |             |
| 7 | <input type="radio"/> idlePattern            | String |             |

Showing 1-7 of 7 records

Rows per page: 10 Page 1 of 1

Edit OK

- 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) —If 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 11-9](#), 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 11-3](#). Create a Data File is shown in the “[Creating Data Files](#)” section on page 11-16, **Edit** is shown in the “[Editing Templates and Data Files](#)” section on page 11-18, and **Delete** is shown in the “[Deleting Templates and Data Files](#)” section on page 11-19.

**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 11-9](#), 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 11-3](#). Create a Data File is shown in the “[Creating Data Files](#)” section on page 11-16, **Edit** is shown in the “[Editing Templates and Data Files](#)” section on page 11-18, and **Delete** is shown in the “[Deleting Templates and Data Files](#)” section on page 11-19.

**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 11-9](#), 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 11-3](#). Create a Data File is shown in the “[Creating Data Files](#)” section on page 11-16, **Edit** is shown in the “[Editing Templates and Data Files](#)” section on page 11-18, and **Delete** is shown in the “[Deleting Templates and Data Files](#)” section on page 11-19.

**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 11-9](#), 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 11-3](#). Create a Data File is shown in the “[Creating Data Files](#)” section on page 11-16, **Edit** is shown in the “[Editing Templates and Data Files](#)” section on page 11-18, and **Delete** is shown in the “[Deleting Templates and Data Files](#)” section on page 11-19.

**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 11-3](#). Create a Data File is shown in the “[Creating Data Files](#)” section on page 11-16, **Edit** is shown in the “[Editing Templates and Data Files](#)” section on page 11-18, and **Delete** is shown in the

“Deleting Templates and Data Files” section on page 11-19.

## Validate

To validate the information you entered in [Figure 11-3](#) (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 11-7.

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 11-2](#).  
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 11-10](#).

**Figure 11-10** Choose Existing Template, Another Way



Click **Create Data File**. An example of a window that appears is shown in [Figure 11-11](#).

Figure 11-11 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 11-11](#), in the **Variables** area, **ctrlName** 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** **ctrlName** can *only* be a string variable.

**Step 6** If you click **Vars** as shown in [Figure 11-11](#), you receive a window as shown in [Figure 11-12](#).

Figure 11-12 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 11-11](#), 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 11-11](#) 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 11-2](#). To edit a data file, a window appears as shown in [Figure 11-10](#).



**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 11-5](#).

**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 11-3](#). Then proceed as in [Step 5](#) in the [Creating Templates](#) section. When editing a data file, you receive a window as shown in [Figure 11-10](#). Then proceed as 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 11-2](#). To delete a data file, a window appears as shown in [Figure 11-10](#).

**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 11-5](#).

**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 11-2](#), or [Figure 11-10](#), 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 11-10](#), **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 11-13](#). If **No** appears in the **In SR Use** column, the **List All SRs** button is disabled.

From [Figure 11-13](#), if you click the **Close** button, the previous window appears.

**Figure 11-13** 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 11-13](#), **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 **In 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 Provisioning policies. It contains the following sections:

- [Overview, page 11-21](#)
- [Associating Templates and Data Files to a Policy, page 11-21](#)

## Overview

Prime Provisioning 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 Provisioning 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 11-23](#). 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 Provisioning.




---

**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 Provisioning, see [Overview, page 11-1](#). 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 Provisioning to append the template-generated CLIs to the regular Prime Provisioning (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 Provisioning 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 11-21](#).
- This feature 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 Provisioning 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, page 11-24](#)
- [Using Templates and Data Files in the Service Request Workflow, page 11-28](#)

### Overview

This section provides overview information about template usage in service requests. It covers the following topics:

- [Associating Templates to a Service Request, page 11-24](#)
- [Associating Subtemplates During Service Provisioning, page 11-25](#)
- [Creating Data Files During Service Request Creation, page 11-26](#)
- [Using Negate Templates to Decommission Template Configurations, page 11-27](#)
- [Using Templates and Data Files in the Service Request Workflow, page 11-28](#)

For details on how these features are implemented in the Prime Provisioning GUI, see the section [Using Templates and Data Files in the Service Request Workflow, page 11-28](#).

### Associating Templates to a Service Request

The template mechanism in Prime Provisioning 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 11-28](#), 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 Provisioning 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 11-1](#). 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 Provisioning 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 11-1](#).

**Table 11-1** 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 |
| 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        |

**Table 11-1** *Default SubTemplate Matching Criteria (continued)*

| Matching Order | Role Type   | Device Type | Line Card | Port Type | Software Version |
|----------------|-------------|-------------|-----------|-----------|------------------|
| 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 Provisioning 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 11-29](#), 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 Provisioning automatically applies the appropriate negate template during the decommission of the service request. For instructions on how to use the Prime Provisioning Template Manager to create negate templates, see [Negate Template, page 11-9](#)

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 Provisioning 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 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).
- Rehomeing 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 11-28](#)
- [Creating a Data File in the Service Request Workflow, page 11-29](#)
- [Decommissioning Service Requests with Added Templates, page 11-30](#)
- [Viewing Templates from the Service Requests Window, page 11-30](#)

### 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 11-23](#), 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 11-29.](#)

- 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 Provisioning 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 11-24.](#)
- 

## 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 11-28.](#) 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 Provisioning.
- 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 11-1](#)

**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 Provisioning service requests that have added templates.



### Note

For general information on how templates are used in Prime Provisioning, see [Overview, page 11-1](#)

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 Provisioning 7.2 API Programmer Guide*.
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 11-2](#) documents some of the available template examples. Refer to the Prime Provisioning GUI for a complete listing of available examples.

Table 11-2 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.                  |
| 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 11-4 on page 11-43, “MPLS Repository Variables”](#)
- [Table 11-3 on page 11-33, “L2VPN Repository Variables”](#)
- [Table 11-7 on page 11-52, “VRF Repository Variables”](#)
- [Table 11-5 on page 11-46, “FlexUNI/EVC Repository Variables”](#)
- [Table 11-6 on page 11-46, “VPLS Repository Variables”](#)

[Table 11-3](#) provides a summary of the MPLS Repository variables available from Prime Provisioning Templates.

**Table 11-3** *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.                                                   |
| 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.                                    |

Table 11-3 MPLS Repository Variables (continued)

| Repository Variable                           | Dimension | Description                                                                                                                                      |
|-----------------------------------------------|-----------|--------------------------------------------------------------------------------------------------------------------------------------------------|
| 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.                                             |
| 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.     |



Table 11-3 MPLS Repository Variables (continued)

| Repository Variable                         | Dimension | Description                                                                                                                                                                  |
|---------------------------------------------|-----------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 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.                                                                                                    |
| Import_Map                                  | 0         | Name of the import map associated with the VRF.                                                                                                                              |
| Is_Default_Info_Originate                   | 0         | Flag to indicate whether the <b>default-information originate</b> 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.                                                                                                               |

Table 11-3 MPLS Repository Variables (continued)

| Repository Variable                    | Dimension | Description                                                                                                                                                     |
|----------------------------------------|-----------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 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.                                          |
| 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.                                               |

Table 11-3 MPLS Repository Variables (continued)

| Repository Variable                           | Dimension | Description                                                                                                               |
|-----------------------------------------------|-----------|---------------------------------------------------------------------------------------------------------------------------|
| 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.                                        |
| 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         | BGPNeighbor MaxPrefix value for BGP routing protocol.                                                                     |
| PE_CE_BGP_MAX_PREFIX_NUMBER_IPV6              | 0         | BGPNeighbor MaxPrefix value for BGP routing protocol, if the Address family is IPv6.                                      |
| PE_CE_BGP_MAX_PREFIX_RESTART                  | 0         | BGPNeighborMaxprefix restart value for BGP routing protocol.                                                              |

Table 11-3 MPLS Repository Variables (continued)

| Repository Variable                           | Dimension | Description                                                                                                                                                                     |
|-----------------------------------------------|-----------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| PE_CE_BGP_MAX_PREFIX_RESTART_IPV6             | 0         | BGPNeighborMaxprefix restart value for BGP routing protocol, if the address family is IPv6.                                                                                     |
| PE_CE_BGP_MAX_PREFIX_THRESHOLD                | 0         | BGPNeighborMaxprefix threshold value for BGP routing protocol.                                                                                                                  |
| PE_CE_BGP_MAX_PREFIX_THRESHOLD_IPV6           | 0         | BGPNeighborMaxprefix threshold value for BGP routing protocol, if the address family is IPv6.                                                                                   |
| PE_CE_BGP_MAX_PREFIX_WARNINGS_ONLY            | 0         | BGPNeighborMaxprefix warnily_only (enable/disable).                                                                                                                             |
| PE_CE_BGP_MAX_PREFIX_WARNINGS_ONLY_IPV6       | 0         | BGPNeighborMaxprefix 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_KEYCHAIN_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_KEYCHAIN_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 |
| 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.                                                                                                          |

Table 11-3 MPLS Repository Variables (continued)

| Repository Variable                         | Dimension | Description                                                                                                                   |
|---------------------------------------------|-----------|-------------------------------------------------------------------------------------------------------------------------------|
| 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 encapsulation.                                                                               |
| PE_EIGRP_AS_ID                              | 0         | EIGRP AS ID on a PE when the routing protocol between a CE and a PE is EIGRP.                                                 |
| 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. |

Table 11-3 MPLS Repository Variables (continued)

| Repository Variable                     | Dimension | Description                                                                                                                     |
|-----------------------------------------|-----------|---------------------------------------------------------------------------------------------------------------------------------|
| 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.                            |
| 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.                                                              |

Table 11-3 MPLS Repository Variables (continued)

| Repository Variable                           | Dimension | Description                                                                                                                                       |
|-----------------------------------------------|-----------|---------------------------------------------------------------------------------------------------------------------------------------------------|
| 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.                                |
| 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.                                               |

Table 11-3 MPLS Repository Variables (continued)

| Repository Variable                             | Dimension | Description                                                                                                                                         |
|-------------------------------------------------|-----------|-----------------------------------------------------------------------------------------------------------------------------------------------------|
| 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.                                                                                                      |
| 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 11-4 provides a summary of the L2VPN Repository variables available from Prime Provisioning Templates.



Table 11-4 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 Provisioning 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).                                                                     |
| 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).                                       |

Table 11-4 L2VPN Repository Variables (continued)

| Repository Variable        | Dimension | Description                                                                                                                            |
|----------------------------|-----------|----------------------------------------------------------------------------------------------------------------------------------------|
| 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.                                          |
| 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.                                                                                                    |

Table 11-4 L2VPN Repository Variables (continued)

| Repository Variable           | Dimension | Description                                                                                                |
|-------------------------------|-----------|------------------------------------------------------------------------------------------------------------|
| 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.                                                       |
| 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.                                                   |

**Table 11-4** L2VPN Repository Variables (continued)

| Repository Variable | Dimension | Description                                                                  |
|---------------------|-----------|------------------------------------------------------------------------------|
| 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 11-5 provides a summary of the VRF Repository variables available from Prime Provisioning Templates.

**Table 11-5** 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 11-6 provides a summary of the FlexUNI/EVC Repository variables available from Prime Provisioning Templates.

**Table 11-6** 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.      |

Table 11-6 FlexUNI/EVC Repository Variables (continued)

| Repository Variable  | Dimension | Description                                                                                                                                                                                                     |
|----------------------|-----------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 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 administrative 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 TDM-CEM service. May be E1 or T1.                                                                                                                                        |
| 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.                                                                                                                                                                                  |

Table 11-6 FlexUNI/EVC Repository Variables (continued)

| Repository Variable              | Dimension | Description                                                                                                                                |
|----------------------------------|-----------|--------------------------------------------------------------------------------------------------------------------------------------------|
| 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.                                                                      |
| HVPLS_ROLE                       | 0         | H-VPLS role type. This returns integer values, which are mapped as follows:<br>0 = HUB, 1 = SPOKE_WITH_SPOKES,<br>2 = SPOKE, -1 = UNKNOWN. |
| 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.                                                                                                 |
| IS_UPE_FLEX_UNI_LINK             | 0         | Flag to set U-PE link as a FlexUNI/EVC link.                                                                                               |

Table 11-6 FlexUNI/EVC Repository Variables (continued)

| Repository Variable | Dimension | Description                                                                                                                                                      |
|---------------------|-----------|------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 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 service type can be either SAToP_UNFRAMED or CESoPN_TIMESLOT.                                                                                                |
| SERVICE_INSTANCE_ID | 0         | Service instance ID (a number: 1 to 8000) corresponding to the EFP for a FlexUNI/EVC enabled link.                                                               |

Table 11-6 FlexUNI/EVC Repository Variables (continued)

| Repository Variable         | Dimension | Description                                                                                                                                                                                    |
|-----------------------------|-----------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 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 mode type. Returns integer values, which are mapped with the following values: 0 = vt-15, 1 = vt15-t1, -1 = UNKNOWN.                                                                       |
| 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.                                                                                                                        |
| Timer2                      | 0         | Second MCPT timer value in microseconds. A number between 1000 and 10000.                                                                                                                      |



Table 11-6 FlexUNI/EVC Repository Variables (continued)

| Repository Variable       | Dimension | Description                                                                                                                                                                                                   |
|---------------------------|-----------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 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_ENCAPSULATION_TYPE    | 0         | Encapsulation type of the interface on the U-PE.                                                                                                                                                              |
| 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.                                       |
| UPE_MATCH_OUTER_VLANS     | 0         | Contains a VLAN ID that needs to be matched for the ingress frame's outer VLAN tag at the UNI device of L2 access link.                                                                                       |
| UPE_PUSH_INNER_VLAN_ID    | 0         | Push a second Dot1q VLAN tag onto an ingress frame of the UNI device of the L2 access link. Contains a VLAN ID to impose upon incoming frames that fulfill the match criteria. Can be a value from 1 to 4096. |
| UPE_PUSH_OUTER_VLAN_ID    | 0         | Push a Dot1q VLAN (outer) tag onto an ingress frame of the UNI device of the L2 access link. Contains a VLAN ID to impose on incoming frames that fulfill the match criteria. Can be a value from 1 to 4096.  |
| UPE_SERVICE_INSTANCE_ID   | 0         | The service instance ID, which represents an Ethernet Flow Point (EFP) on an interface in the EVC infrastructure.                                                                                             |
| UPE_SERVICE_INSTANCE_NAME | 0         | The service instance name for UNI device of L2 access link.                                                                                                                                                   |

**Table 11-6 FlexUNI/EVC Repository Variables (continued)**

| Repository Variable         | Dimension | Description                                                                                                                                                                                                     |
|-----------------------------|-----------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| UPE_TRANSLATE_INNER_VLAN_ID | 0         | Target inner VLAN ID of a frame that is being translated (VLAN translation). The inner tag of all the incoming frames that fulfill the match criteria are translated to this ID. Can be a value from 1 to 4096. |
| UPE_TRANSLATE_OUTER_VLAN_ID | 0         | Target outer VLAN ID of a frame that is being translated (VLAN translation). The outer tag of all the incoming frames that fulfill the match criteria are translated to this ID. Can be a value from 1 to 4096. |
| 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.                                                                                                           |
| VFI_Name                    | 0         | The virtual forwarding instance (VFI) name.                                                                                                                                                                     |
| 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 11-7 provides a summary of the VPLS Repository variables available from Prime Provisioning Templates.

**Table 11-7 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.                                             |

Table 11-7 VPLS Repository Variables (continued)

| Repository Variables         | Dimension | Description                                                                                                                                    |
|------------------------------|-----------|------------------------------------------------------------------------------------------------------------------------------------------------|
| 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.                                                                                     |
| 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.                     |

Table 11-7 VPLS Repository Variables (continued)

| Repository Variables       | Dimension | Description                                                                                                                |
|----------------------------|-----------|----------------------------------------------------------------------------------------------------------------------------|
| 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.                                                                     |
| 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 Provisioning 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: **\$PRIMEP\_HOME/bin/importExportTemplateDB.sh**.



**Note** The template script (`./importExportTemplateDB.sh`) needs to be run in shell mode. To enter into shell mode you need to issue “`./prime.sh shell`”, and then execute the template script “`./importExportTemplateDB.sh`” in shell mode.

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 Provisioning 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 Provisioning 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?, page 11-57](#)
- [How do I obtain address information from the given IP address?, page 11-57](#)
- [How do I obtain the octets from the given IP address?, page 11-57](#)
- [How do I call a subtemplate in a template?, page 11-58](#)
- [How do I concatenate two strings?, page 11-58](#)
- [How can I convert a string to an integer and how can I increase the last octet of the IP address by one?, page 11-58](#)
- [Can I use nested if statements?, page 11-59](#)
- [How can I perform basic arithmetic operations?, page 11-59](#)

- [How can I retrieve data from a two-dimensional array and what is the use of \\$velocityCount?, page 11-59](#)
- [How can I print \\$a instead of its value?, page 11-60](#)
- [What is the difference between #include\(\) and #parse\(\)?, page 11-60](#)
- [What is a macro and how is it used?, page 11-61](#)
- [What is a range operator and how can I use it?, page 11-62](#)
- [How can I split strings containing special characters?, page 11-62](#)
- [How can I use repository variables?, page 11-62](#)
- [How can I use a variable as a dynamic URL?, page 11-62](#)
- [Can I see more examples?, page 11-63](#)

## How do I split a string?

Prime Provisioning provides a function `substringToDelim()`, 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: `$b=$TMSsystem.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 Provisioning provides the functions that can be used to get the address, mask, and reverse mask from the given IP address.

Usage:

`$TMSsystem.getAddr ("10.33.4.5/30")` returns 10.33.4.5

`$TMSsystem.getMask ("10.33.4.5/30")` returns 255.255.255.252

`$TMSsystem.getReverseMask ("10.33.4.5/30")` returns 0.0.0.3

`$TMSsystem.getNetworkAddr ("10.33.4.5/30")` returns 10.33.4.4

`$TMSsystem.GetClassfulNetworkAddr ("10.33.4.5/30")` returns 10.0.0.0

`$TMSsystem.CurrentTimeInIOSFormat ()` returns hh:mm:ss day\_of\_month month\_of\_year year

## How do I obtain the octets from the given IP address?

Prime Provisioning 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=vpncs** and **\$b=properties**

then: **#{a}#{b}** concatenates these two strings and gives the result as **vpncsproperties**.

or, **#{a}\_#{b}** gives the result as **vpncs\_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
#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:

`$PRIMEP_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:

`$PRIMEP_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="c1","c2","c3")
```

```
#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 c1 30:20
```

```
ipcommunity-list permit c2 30:20
```

```
ipcommunity-list permit c3 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 11-63](#)
- [Usage of a Macro, page 11-64](#)
- [Usage of Subtemplates, page 11-65](#)

## 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")
```

**Figure 11-14**     *Template Data File Editor*

The screenshot shows a web-based interface titled "Template Editor". It features a "Template Information" section with the following fields:

- Template Name \***: A text input field.
- Description**: A text input field.
- Body \***: A large text area for entering the template content.

Below these fields are two checkboxes:

- Has Negate Template**:
- Has User Reference**:

At the bottom right, there are three buttons: "Select", "Save", and "Close". A note at the bottom left states "Note: \* - Required Field". A vertical ID number "285747" is visible on the right side of the interface.

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.





## Monitoring: Task Manager

---

This chapter explains how to use the Task Manager to monitor and perform various tasks in Prime Provisioning.

Task Manager 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 chapter contains the following sections:

- [Tasks, page 12-1](#)
- [Task Logs, page 12-5](#)

## Tasks

This section contains the following topics:

- [Starting Task Manager, page 12-1](#)
- [Create, page 12-2](#)
- [Audit, page 12-3](#)
- [Details, page 12-3](#)
- [Schedules, page 12-4](#)
- [Logs, page 12-4](#)
- [Delete, page 12-4](#)
- [Collect Config from Files, page 12-4](#)

## 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 12-1](#).
- **Collect Config**—Collects configuration from devices.
  - **Collect Config From Files**—Collects configurations from files.
  - **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**—Performs discovery of all TE enabled devices. The discovery task runs without stopping until all devices have been discovered.
  - **TE Incremental Discovery**—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.
  - **TE Interface Performance**—Calculates tunnel and interface bandwidth utilization using SNMP.

**Figure 12-1** Create Tasks

Create Task

Config Collection - Task Information

Name\*: Collect Config 2012-07-12 09:43:14.199

Type: Collect Config

Description: Created on 2012-07-12 09:43:14.199

Back Next Finish Close

Note: \* - Required Field

285769

- 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 Provisioning 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 Provisioning server into the **Configuration File Directory** text box, to indicate the directory on the Prime Provisioning 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 Provisioning 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 12-5.

## 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.

## Collect Config from Files

To use this feature, you should have the following:

- Configlets of a device saved as a XML file in the below format.

```
<?xml version="1.0" encoding="UTF-8" ?>
-<Device_Config>
 -<Running_Config>
 <![CDATA[]]>
 -</Running_Config>
```

```

--<Version>
 <![CDATA[]]>
</Version>
</Device_Config>

```

- Directory details where the XML file is placed.

To collect configuration details from a file, perform the below steps:

- 
- Step 1** Click **Operate > Tasks > Task Manager**.
- Step 2** In the Task Manager window, click **Create**.
- Step 3** Choose **Collect Config from Files** from the dropdown list.
- Step 4** In the Create Task window, you are able to modify the name and description details.
- Step 5** Click **Next**.
- Step 6** In Collect Config Task window, enter the directory details of the XML file in the **Configuration File Directory** field.
- Step 7** Choose **Submit**.



**Note** The device should be available in the Inventory for the collect config task to run successfully.

---

## 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.
-





## Using Inventory Manager

---

This chapter explains how Inventory Manager provides a method of managing mass changes to inventory and service model data in the Prime Provisioning provisioning process. In this process, Inventory Manager enables an operator to import network-specific data into the Prime Provisioning Repository (Repository) in bulk mode. Prime Provisioning 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, page 13-1](#)
- [Prime Network Device Import, page 13-11](#)
- [Changing a Node to Unmanaged State, page 13-14](#)

### Inventory - Device Console



Note

---

The Device Console is now enabled/disabled using a DCPL property. For information about using DCPL properties, see the [Cisco Prime Provisioning 7.2 Administration Guide](#) or see [Appendix G, “Property Settings”](#)

---

**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 Provisioning 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 appears 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 appears 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.

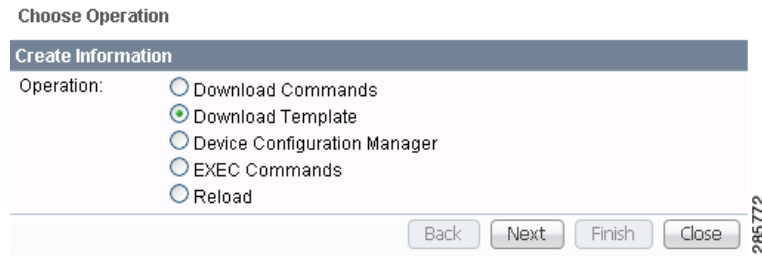
To download a template, follow these steps:

- Step 1** Choose **Inventory > Device Tools > Device Console** .

**Step 2** Select **Download Template** and click **Next**.

A window appears 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 appears 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 appears as shown in [Figure 13-7](#).

**Figure 13-7** Download Template Summary

Download Template

**Template Summary**

Devices: iscind-7609-1  
iscind-7609-2

Device Groups:

Template: /Examples/Configure\_PE\_as\_ASBR\_VPN\_Specific\_Template\_TMPL\_

Upload Config After Download

Retrieve device attributes

Back Next Finish Close

285775

- 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 appears 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 G, “Property Settings”](#) and can be set in the software UI as specified in the [Cisco Prime Provisioning 7.2 Administration Guide](#)

- **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 appears as shown in [Figure 13-9](#).

**Figure 13-9** Device Console—EXEC Commands: Select Devices



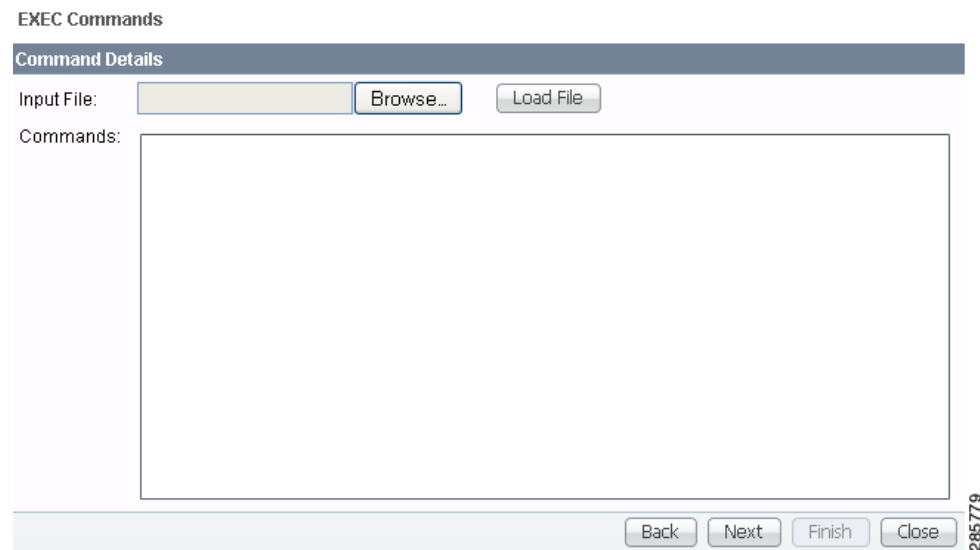
- Step 3** Continue with [Step 4](#) if you want to add devices; proceed to [Step 7](#) to delete devices; or click **Next** to proceed to [Step 9](#) for **3. Select Device Groups**.
- Step 4** Click **Add**, as shown in [Figure 13-9](#), to **2. Select Devices**.
- Step 5** From the resulting window, 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 appears 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 appears 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** to select device groups.
- Step 11** From the resulting window, check the check box(es) for each device group you want to select. Then click **Select**.
- Step 12** Return to the window 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 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 appears as shown in [Figure 13-14](#).

**Figure 13-14** 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 Provisioning 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 [Cisco Prime Provisioning 7.2 Administration Guide](#) or see [Appendix G, “Property Settings”](#)



### 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 Provisioning collects information. Every network element that Prime Provisioning 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](#)
- [Importing Device\(s\) from Prime Network in Multi Instance Environment, page 13-13](#)

## Device Import Prerequisite

To import the device(s) from Prime Network into Prime Provisioning, you must first import the Prime Network certificate into Prime Provisioning Trust Store. For more information about this, refer to the section **Integrating with Prime Network for Device Import** in [Cisco Prime Provisioning 6.8 Installation Guide](#).

## 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 IOS/IOS-XR 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](#)



### Note

If you have configured a Prime Network Gateway in DCPL properties, you are able to view a **Import** button next to the **Management IP Address** field in the Cisco Device creation page. By providing the IP address, you are able to import the device from the configured Prime Network.

---

**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 Abstraction (ANA) and imported into Prime Provisioning. 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 2-11](#)
  - [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.

## Importing Device(s) from Prime Network in Multi Instance Environment

Device(s) which already exist in multi instance of Prime Network can be imported directly into Prime Provisioning using the option available in the Inventory Manager window.

To perform single or bulk import of Cisco and non-Cisco devices, follow these steps:

- 
- Step 1** Choose **Inventory > Physical Inventory > Inventory Manager**.  
The **Device List** window appears.
- Step 2** Click **Import Devices** drop down list.
- Step 3** Select Prime Network.
- Step 4** The **Inventory Import Filter** window appears.
- a. You can filter the import of devices from Prime Network before getting it into Prime Provisioning.
    - The devices available in Prime Network can be filtered based on Device Host Name, Management IP Address and Device Platform.
    - Once filtration 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 Provisioning.

- b. If you want to import all the devices available in Prime Network, click **OK** button without providing any filtering criteria on the filter screen.

**Step 5** The **Device List** window appears.

**Step 6** The Config Collect and Ring Discovery can be scheduled during device import. Click on **Action** button to schedule:

- Config Collect
- Config Collect + Ring Discovery
- MPLS Label Sync
- MPLS-TP Discovery
- Ring Discovery

**Step 7** Click **Save**.

The Devices window reappears with the new devices added.

---

## Changing a Node to Unmanaged State

In some situations, it can be advantageous to make a node unmanaged. For example, if a node has to be removed, service requests that included this node can fail. To avoid this, one solution is to make the node unmanaged.

To make a node unmanaged, perform the following steps:

---

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

**Step 2** Click **Open** and choose **Provider**.

**Step 3** In the Select provider pop-up screen, select a Provider.

**Step 4** Click **Attributes** and choose **PE Attributes**.

**Step 5** Select the device that you want to make Unmanaged.

**Step 6** Check the checkbox next to the header **Managed**.

**Step 7** Select **Edit** and uncheck the box next to the attribute Managed.

**Step 8** Click **Save**.

The value of the Managed column appears as **NO** for the chosen device.

---



## Prime Provisioning XML Reference

This appendix contains an alphabetical listing of the XML rules, tags, and attributes that are used in the XML files used for Prime Provisioning Discovery.

**Table A-1** *Prime Provisioning XML Rules, Tags, and Attributes*

| Tag                                  | Description                                                                                                                                                                                                                                                                                                                                                                      |
|--------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <code>&lt;as-number&gt;</code>       | Specifies the autonomous system (AS) number for the provider. The AS number can be an integer between 1 and 65535.                                                                                                                                                                                                                                                               |
| <code>&lt;CDP&gt;</code>             | <p>Starts a <code>&lt;CDP&gt;</code> tag. The <code>&lt;CDP&gt;</code> tag specifies an seed IP address and a hop count.</p> <p>The <code>&lt;CDP&gt;</code> tag must contain the following attributes:</p> <ul style="list-style-type: none"> <li>• <b>ipaddress</b></li> <li>• <b>hop</b></li> </ul>                                                                           |
| <code>&lt;connection&gt;</code>      | <p>Starts a <code>&lt;connection&gt;</code> tag. The <code>&lt;connection&gt;</code> tag must specify the following attributes:</p> <ul style="list-style-type: none"> <li>• <b>discovery-protocol</b></li> <li>• <b>fromDevice</b></li> <li>• <b>FromIP</b></li> <li>• <b>FromInterface</b></li> <li>• <b>toDevice</b></li> <li>• <b>toIP</b></li> <li>• <b>toIF</b></li> </ul> |
| <code>&lt;create-customer&gt;</code> | <p>Starts a <b>create-customer</b> rule. The <b>create-customer</b> rule creates a region object. the create-customer rule must contain the following tags:</p> <ul style="list-style-type: none"> <li>• <code>&lt;customer-name&gt;</code></li> <li>• <code>&lt;create-site&gt;</code></li> </ul>                                                                               |

Table A-1 IPrime Provisioning XML Rules, Tags, and Attributes (continued)

| Tag                             | Description                                                                                                                                                                                                                                                                                                                                                                                                                        |
|---------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>&lt;create-provider&gt;</b>  | <p>Starts a <b>create-provider</b> rule. The create-provider rule creates a service provider object.</p> <p>The <b>create-provider</b> rule must contain the following tags:</p> <ul style="list-style-type: none"> <li>• <b>&lt;provider-name&gt;</b></li> <li>• <b>&lt;as-number&gt;</b></li> <li>• <b>&lt;create-region&gt;</b></li> </ul>                                                                                      |
| <b>&lt;create-region&gt;</b>    | <p>Starts a <b>create-region</b> rule. The <b>create-region</b> rule creates a region object. The <b>create-region</b> rule must contain a <b>region-name</b> tag.</p>                                                                                                                                                                                                                                                             |
| <b>&lt;create-site&gt;</b>      | <p>Starts a <b>create-site</b> rule. The create-site rule must contain a <b>&lt;site-name&gt;</b> tag.</p>                                                                                                                                                                                                                                                                                                                         |
| <b>&lt;customer-name&gt;</b>    | <p>Specifies a customer name. Required within the <b>create-customer</b> rule.</p>                                                                                                                                                                                                                                                                                                                                                 |
| <b>&lt;device&gt;</b>           | <p>Starts a <b>&lt;device&gt;</b> tag. The <b>&lt;device&gt;</b> tag must contain the following tags:</p> <ul style="list-style-type: none"> <li>• <b>&lt;device-name&gt;</b></li> <li>• <b>&lt;ip-address&gt;</b></li> </ul> <p>The following tags are optional within the <b>&lt;device&gt;</b> tag:</p> <ul style="list-style-type: none"> <li>• <b>&lt;system-object-id&gt;</b></li> <li>• <b>&lt;snmp-info&gt;</b></li> </ul> |
| <b>&lt;device-name&gt;</b>      | <p>Specifies the name of the device. Required within the <b>&lt;device&gt;</b> tag.</p>                                                                                                                                                                                                                                                                                                                                            |
| <b>&lt;DISCOVERY_METHOD&gt;</b> | <p>Starts a <b>&lt;DISCOVERY_METHOD&gt;</b> tag. The <b>&lt;DISCOVERY_METHOD&gt;</b> tag must contain a <b>&lt;CDP&gt;</b> tag.</p>                                                                                                                                                                                                                                                                                                |
| <b>discovery-protocol</b>       | <p>Specifies the Discovery protocol used to discover the network topology. Normally, this is “CDP.”</p>                                                                                                                                                                                                                                                                                                                            |
| <b>fromDevice</b>               | <p>Specifies the name of the device from which the Named Physical Circuit starts. Required attribute for the <b>&lt;connection&gt;</b> tag.</p>                                                                                                                                                                                                                                                                                    |
| <b>FromInterface</b>            | <p>Specifies the name of the device interface from which the Named Physical Circuit starts. Required attribute for the <b>&lt;connection&gt;</b> tag.</p>                                                                                                                                                                                                                                                                          |
| <b>FromIP</b>                   | <p>Specifies the management IP address of the device from which the Named Physical Circuit starts. Required attribute for the <b>&lt;connection&gt;</b> tag.</p>                                                                                                                                                                                                                                                                   |

Table A-1 IPrime Provisioning XML Rules, Tags, and Attributes (continued)

| Tag                             | Description                                                                                                                                                                |
|---------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>hop</b>                      | Specifies the number of hops from the device identified by the <b>ipaddress</b> attribute to go in discovering devices. Required attribute for the <b>&lt;CDP&gt;</b> tag. |
| <b>ipaddress</b>                | Specifies the IP address of a seed device. Required attribute for the <b>&lt;CDP&gt;</b> tag.                                                                              |
| <b>&lt;ip-address&gt;</b>       | Specifies the IP address of the device. Required within the <b>&lt;device&gt;</b> tag.                                                                                     |
| <b>&lt;provider-name&gt;</b>    | Specifies the name of the provider.                                                                                                                                        |
| <b>&lt;region-name&gt;</b>      | Specifies the name of a region.                                                                                                                                            |
| <b>&lt;ro-community&gt;</b>     | Specifies the level of SNMP access for the device. Normally, this should be “public.” Required within the <b>&lt;snmp-info&gt;</b> tag.                                    |
| <b>&lt;site-name&gt;</b>        | Specifies a site name.                                                                                                                                                     |
| <b>&lt;snmp-info&gt;</b>        | Specifies SNMP information for the device. The <b>&lt;snmp-info&gt;</b> tag must contain a <b>&lt;ro-community&gt;</b> tag. Optional within the <b>&lt;device&gt;</b> tag. |
| <b>&lt;system-object-id&gt;</b> | (optional) Can be included to specify the SNMP Object ID (OID) for the device. If this is provided, it is specified within the <b>&lt;device&gt;</b> tag.                  |
| <b>toDevice</b>                 | Specifies the name of the device to which the Named Physical Circuit connects. Required attribute for the <b>&lt;connection&gt;</b> tag.                                   |
| <b>toIF</b>                     | Specifies the device interface on the device to which the Named Physical Circuit connects. Required attribute for the <b>&lt;connection&gt;</b> tag.                       |
| <b>toIP</b>                     | Specifies the management IP address of the device from which the Named Physical Circuit connects. Required attribute for the <b>&lt;connection&gt;</b> tag.                |







## 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, page B-1](#)
- [Setting Up an NPC Access Ring with Two N-PEs, page B-3](#)
- [Using N-PE Redundancy in FlexUNI/EVC Service Requests, page B-3](#)
- [Using N-PE Redundancy in MPLS Service Requests, page B-4](#)
- [Additional Network Configurations and Sample Configlets, page B-5](#)

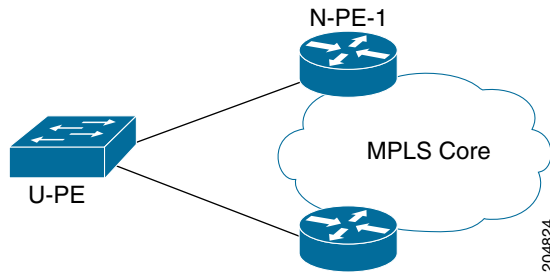
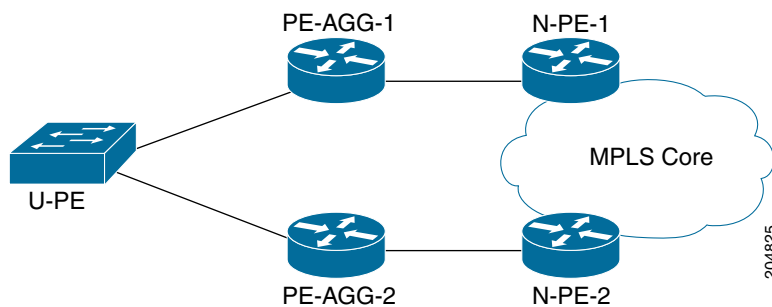
### Overview

Prime Provisioning 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 Provisioning 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 B-1](#) and [Figure B-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 B-1** N-PE Redundancy, Starting at the U-PE**Figure B-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 B-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 B-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 B-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 Provisioning 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 [Cisco Prime Provisioning 7.2 Release Notes](#), 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 Provisioning. The basic steps for doing this are described [Setting Up Logical Inventory, page 2-55](#). Additional information is provided in this guide in the section [Creating Named Physical Circuits, page 3-12](#).

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 Provisioning 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 Provisioning allows the use of loopback interfaces in a ring.

**Note**

Note that Prime Provisioning 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 Provisioning 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 Provisioning 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 Provisioning 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 B-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.

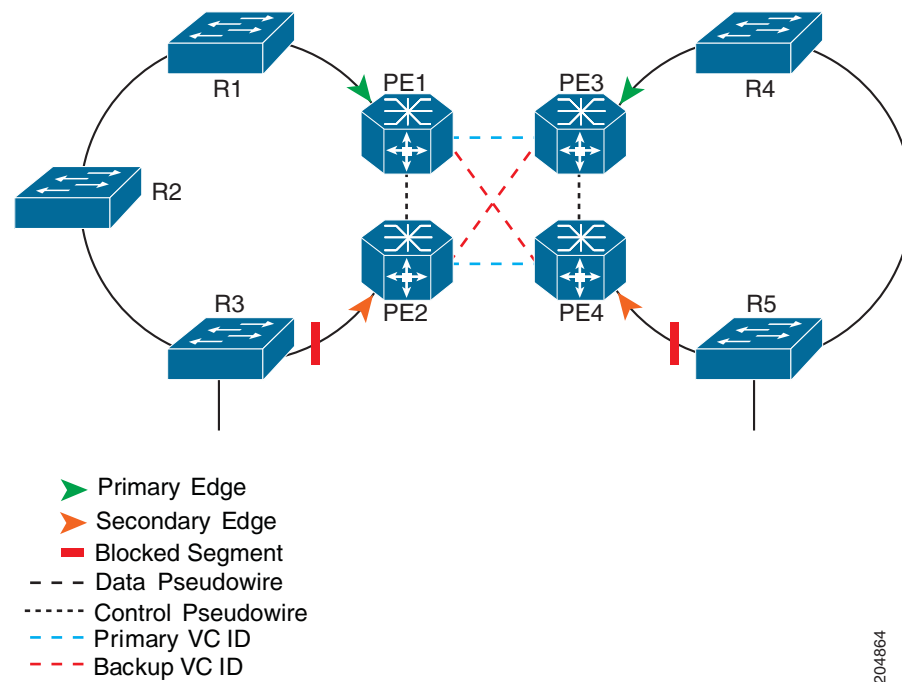
# Additional Network Configurations and Sample Configlets

This section provides additional network scenarios for reference, along with sample configlets for associated network devices.

## Example 1: Pseudowire Connectivity (A)

Figure B-3 illustrates a network configuration with pseudowire connectivity with dual-homed N-PEs on both sides of the network and with pseudowire redundancy.

**Figure B-3** Pseudowire Connectivity, Dual-Homed N-PEs on Both Sides of the Network, with Pseudowire Redundancy



204864

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**

```

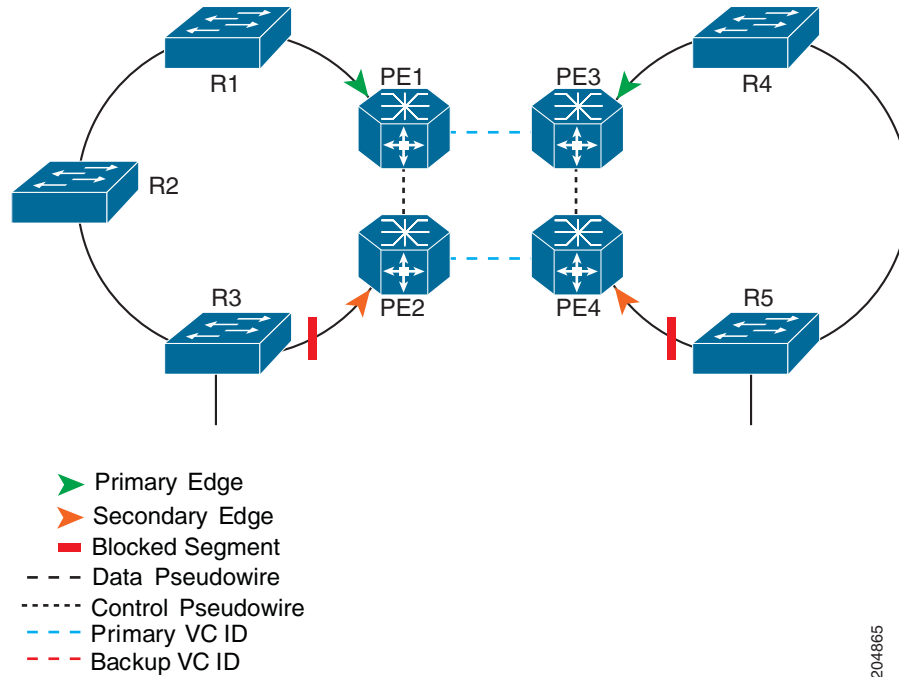
vlan <S-Vlan>
!
interface <UNI-to-R5>
 switchport
 switchport trunk encapsulation dot1q
 switchport mode trunk
 switchport trunk allowed vlan add <S-Vlan>
!
interface vlan <S-Vlan>
 xconnect <PE2 loopback> <PrimaryVcId> encapsulation mpls
 backup peer <PE1 loopback> <BackupVcId>

```

## Example 2: Pseudowire Connectivity (B)

Figure B-4 illustrates a network configuration using pseudowire connectivity, with dual-homed N-PEs on both sides of the network without pseudowire redundancy.

**Figure B-4** Pseudowire Connectivity, Dual-Homed N-PEs on Both Sides of the Network, with No Pseudowire Redundancy



204865

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**

```

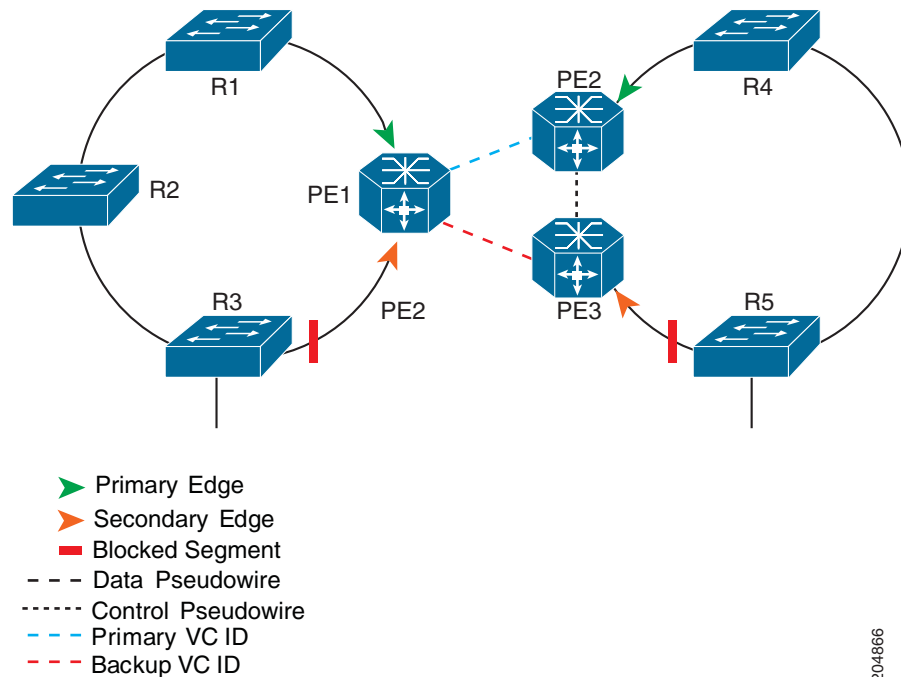
vlan <S-Vlan>
!
interface <UNI-to-R5>
 switchport
 switchport trunk encapsulation dot1q
 switchport mode trunk
 switchport trunk allowed vlan add <S-Vlan>
!
interface vlan <S-Vlan>
 xconnect <PE2 loopback> <PrimaryVcId> encapsulation mpls

```

## Example 3: Pseudowire Connectivity (C)

Figure B-5 illustrates a network configuration using pseudowire connectivity with dual-homed N-PEs at one side of the network and with pseudowire redundancy.

**Figure B-5** Pseudowire Connectivity, Dual-Homed N-PEs on One Side of the Network, with Pseudowire Redundancy



204866



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
```

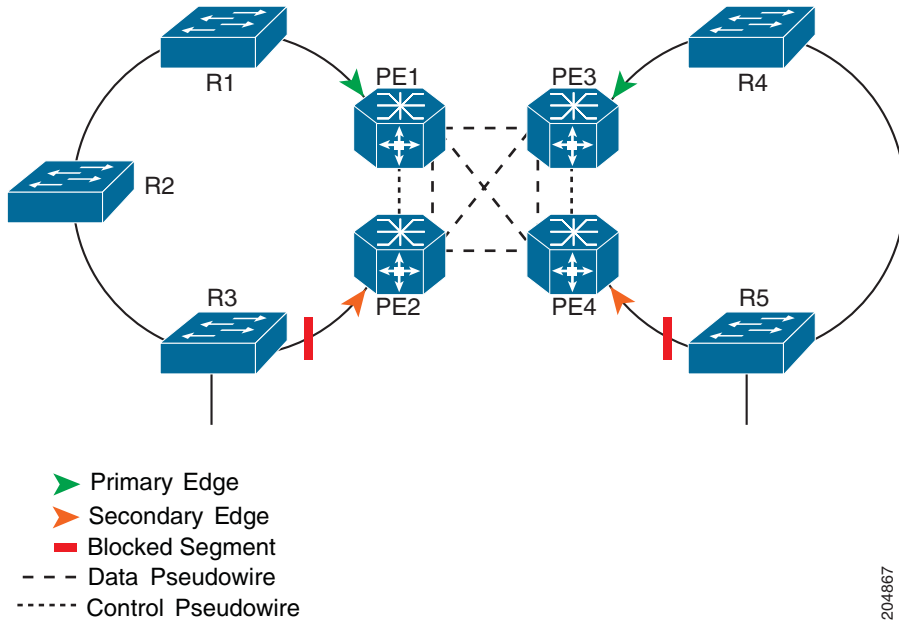
#### PE3

```
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 B-6](#) illustrates a network configuration using VPLS connectivity with dual-homed N-PEs on both sides of the network.

Figure B-6 VPLS Connectivity, Dual-Homed N-PEs on Both Sides of the Network



204867

Sample configlets for the devices are provided below.

**PE1**

```
vlan <S-Vlan>
!
12 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>
!
12 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>
!
12 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>
!
12 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>
```





## 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, page C-1](#)
- [Using Views in Prime Provisioning, page C-2](#)

## Creating Repository Views

This section describes how to create views in Sybase repository and Oracle repository.

- [Creating Views Sybase Repository, page C-1](#)
- [Creating Views in Oracle Repository, page C-2](#)

## Creating Views Sybase Repository

### New and Upgrade Installation

All the views available in Prime Provisioning (see the [Using Views in Prime Provisioning, page C-2](#)) are created as part of the new and upgrade installation of Prime Provisioning .7.1.

## Creating Views in Oracle Repository

### New and Upgrade Installation

To create repository views (see the [Using Views in Prime Provisioning, page C-2](#)) in new and upgrade installation of Prime Provisioning 6.5, 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 Provisioning 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 Provisioning

The different views available in Prime Provisioning are as follows:

- [Summary View, page C-2](#)
- [Site View, page C-4](#)
- [Customer View, page C-5](#)
- [Region View, page C-5](#)

### Summary View

You can query using the column name for summary view. [Table C-1](#) describes the column name and its type name.

*Table C-1 Summary View Column Names*

| Column Name           | Type Name |
|-----------------------|-----------|
| SR_Number             | Integer   |
| SR_STATE              | Integer   |
| SR_Last_Modified_Time | Varchar   |

**Table C-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 Provisioning 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              | In Valid         |
| 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 view. [Table C-2](#) describes the column name and its type name.



**Table C-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 C-3](#) describes the column name and its type name.

**Table C-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 C-4](#) describes the column name and its type name.

**Table C-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;
```



## Adding Additional Information to Services

This appendix describes how the additional information feature is supported in Prime Provisioning.



Note

For MPLS and EVC services, it is recommended that you use a new policy customization feature. For more information, see [Chapter 8, “Customizing EVC, MPLS and MPLS-TP Policies”](#).

It contains the following sections:

- [Overview, page D-1](#)
- [Prerequisites and Limitations, page D-1](#)
- [Summary of the Additional Information GUI Workflow, page D-2](#)
- [Setting Additional Information in the Policy Workflow, page D-2](#)
- [Setting Additional Information in the Service Request Workflow, page D-4](#)
- [Using Additional Attributes with Templates and Data Files, page D-5](#)
- [Using Additional Attributes with xDE Provisioning, page D-6](#)
- [Creating the Additional Information Definition File, page D-7](#)
- [Example of the Additional Information Feature, page D-10](#)

## 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 Provisioning.

## 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 D-7](#).

## Summary of the Additional Information GUI Workflow

The following steps summarize the tasks you need to perform to implement additional information in Prime Provisioning. 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 D-1](#). This window looks and functions the same in all of the policy types that support the additional information feature.

**Figure D-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 D-7.

The default path and name of the definition file is:

\$PRIMEP\_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 D-2](#).

**Figure D-2 Attributes Loaded from an External XML Definition File**

| # | Name           | Value  | Range/Units | Type   | Description |
|---|----------------|--------|-------------|--------|-------------|
| 1 | DisplayName1 * | Value1 |             | String |             |
| 2 | DisplayName2 * | Value2 |             | String |             |
| 3 | DisplayName3 * | Value3 |             | String |             |

**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 Provisioning.
- 

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 addition 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 Provisioning 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.

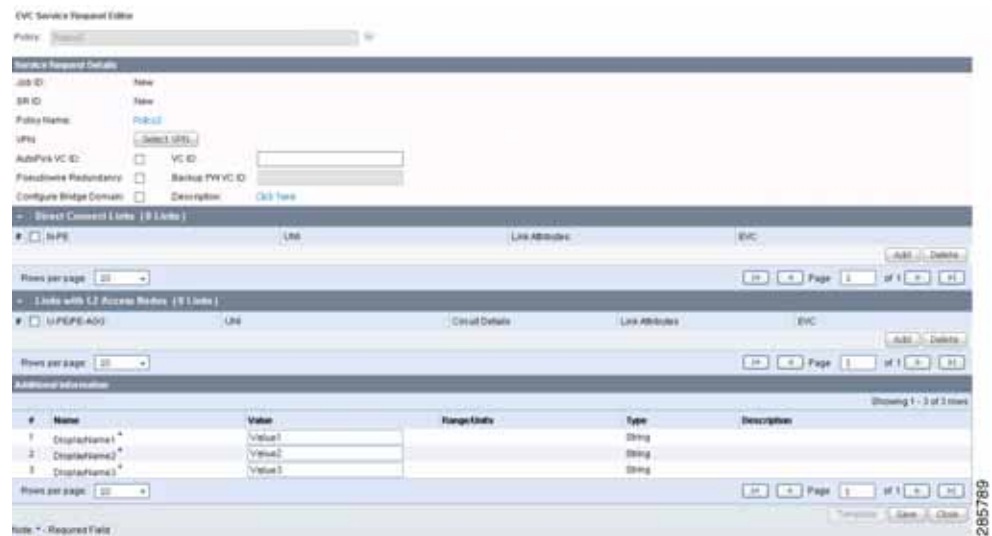
For more information about validation checks done on the additional information in the policy workflows, see [How the XSD is Validated, page D-10](#).

## 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 D-3](#).

**Figure D-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 D-5.
- 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 Provisioning:

- 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 D-2.

## Minimum Mandatory XML Elements

[Example D-1](#) is an example additional information definition file that contains the minimum information needed to define an additional information attribute.

### **Example D-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 definition file also may contain optional XML elements. This section describes the following optional elements:

- *group*
- *attribute/displayName*
- *attribute/description*
- *attribute/required*
- *attribute/type*
- *attribute/type/string*
- *attribute/type/integer*
- *attribute/type/ipv4Address*
- *attribute/type/ipv6Address*
- *attribute/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 D-10](#).

## 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:

\$PRIMEP\_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 data 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>
```

■ Example of the Additional Information Feature





## Deprecated Features: Layer 2 Legacy Services and Other Services

---

This appendix describes Layer 2 services and the reports feature that have been deprecated and are no longer directly accessible from the product. They can be reactivated using DCPL properties. See your Cisco representative for further details.

This appendix describes how to use policies and service requests to manage various legacy L2VPN and VPLS services as well as how to manage reports in Prime Provisioning. It contains the following sections:

- [Getting Started with L2VPN Services, page E-2](#)
- [Setting Up the Prime Provisioning Services, page E-6](#)
- [Creating an L2VPN Policy, page E-19](#)
- [Managing an L2VPN Service Request, page E-24](#)
- [Creating a VPLS Policy, page E-35](#)
- [Managing a VPLS Service Request, page E-38](#)
- [Deploying, Monitoring, and Auditing Service Requests, page E-44](#)
- [Setting Up VLAN Translation for L2VPN ERS \(EVPL\) Services, page E-45](#)
- [Policy and Service Request Attributes Reference Tables, page E-50](#)
- [Sample Configlets, page E-63](#)
- [Reports, page E-88](#)
- [EMAIL, page E-108](#)



**Note**

---

The recommended way of managing the service requests described in this appendix is via EVC.

---

# Getting Started with L2VPN Services

This section provides a road map to help you get started using the L2VPN component in Cisco Prime Provisioning 7.2. It contains the following sections:

- [Overview, page E-2](#)
- [Prepopulating a Service by Selecting Endpoints in Prime Network, page E-2](#)
- [Installing Prime Provisioning and Configuring the Network, page E-3](#)
- [Configuring the Network to Support Layer 2 Services, page E-3](#)
- [Setting Up Basic Prime Provisioning Services, page E-3](#)
- [Working with L2VPN and VPLS Policies and Service Requests, page E-5](#)
- [A Note on Terminology Conventions, page E-5](#)

## 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 Provisioning and L2VPN services. The following subsections provide a summary of the key tasks you must accomplish to be able to provision L2VPN and VPLS services using Prime Provisioning. 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 Provisioning documentation set are provided. See the referenced documentation for more detailed information. After the basic installation and configuration steps are completed for both Prime Provisioning and the L2VPN component, see the subsequent sections to create and provision L2VPN and VPLS services.

## Prepopulating a Service by Selecting Endpoints in Prime Network

It is possible to create service by picking endpoints on a map in Prime Network Vision, when Prime Provisioning and Prime Network are integrated with Prime Central

- 
- Step 1** On any map, select one or more endpoint devices by using CTRL click.
  - Step 2** In the right click menu, select **Fulfill/Create Service**.
  - Step 3** You will be taken to the same first screen as you see when creating a service in Prime Provisioning.
  - Step 4** Pick a policy.  
Depending on the number of endpoints selected, not all policies will work. For example, you cannot create a point-to-point service if you have five endpoints selected, but you can create a VPLS or a L3 VPN.
  - Step 5** Once you have selected the policy, the service request main page will appear as usual, prepopulated with links and with the selected devices.
-

## Installing Prime Provisioning and Configuring the Network

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



Note

To use the L2VPN component within Prime Provisioning, 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 Provisioning, you must perform the following network configuration steps to support Layer 2 services. Information on doing these steps is not provided in the Prime Provisioning 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 Provisioning Services

After the basic network configuration tasks are completed to support Prime Provisioning and L2 services, you use Prime Provisioning to define elements in the Prime Provisioning 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 Provisioning tasks are covered in [Chapter 2, “Before Setting Up Prime Provisioning.”](#) You can also find a summary of some important Prime Provisioning set up tasks in [Setting Up the Prime Provisioning Services, page E-6](#). The information below is a checklist of basic Prime Provisioning 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 Provisioning repository. These are global resources that can be used by all Prime Provisioning 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 2-39](#). See also [Defining a Service Provider and Its Regions, page E-9](#).
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 2-39](#). See also [Defining Customers and Their Sites, page E-9](#).

3. **Import or add raw devices.** Every network element that Prime Provisioning manages must be defined as a device in the Prime Provisioning repository. An element is any device from which Prime Provisioning can collect information. In most cases, devices are Cisco IOS routers and switches. You can set up devices in Prime Provisioning manually or through importing device configuration files.
4. **Assign devices roles as PE or CE.** After devices are created in Prime Provisioning, 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 Provisioning 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 Provisioning, 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 E-4](#).

## Setting Up Prime Provisioning Resources for L2VPN and VPLS Services

Some Prime Provisioning resources, such as access domains, VLAN pools, and VC pools are set up to support Prime Provisioning 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 Provisioning 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 Provisioning. 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 2-39](#). See also [Creating Access Domains, page E-9](#).
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 Provisioning 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 2-39](#). See also [Creating VLAN Pools, page E-10](#).
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 2-39](#). See also [Creating a VC ID Pool, page E-11](#).

## 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 2-55](#). See also [Creating Named Physical Circuits, page E-12](#).

## 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 2-55](#). See also [Defining VPNs, page E-9](#).

## Working with L2VPN and VPLS Policies and Service Requests

After you have set up providers, customers, devices, and resources in Prime Provisioning, you are ready to create 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.



### Note

---

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.

---

1. **Review overview information about L2 services concepts.** See the chapter “Prime Provisioning Layer 2 VPN Concepts” in the *Cisco Prime Provisioning 7.2 Administration Guide*.
  2. **Set up an L2VPN or VPLS policy.** See the appropriate section, depending on the type of policy you want to create:
    - [Creating an L2VPN Policy, page E-19](#)
    - [Creating a VPLS Policy, page E-35](#)
  3. **Provision the L2VPN, or VPLS service request.** See the appropriate section, depending on the type service request you want to provision:
    - [Creating an L2VPN Policy, page E-19](#)
    - [Creating an L2VPN Policy, page E-19](#)
    - [Managing an L2VPN Service Request, page E-24](#)
    - [Managing a VPLS Service Request, page E-38](#)
  4. **Deploy the service request.** See [Deploying, Monitoring, and Auditing Service Requests, page E-44](#).
  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 E-44](#).
    - Audit service requests. See [Deploying, Monitoring, and Auditing Service Requests, page E-44](#).
- 

## A Note on Terminology Conventions

The Prime Provisioning 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 E-1](#).

See the chapter “Prime Provisioning Layer 2 VPN Concepts,” in the *Cisco Prime Provisioning 7.2 Administration Guide*, for more information on terminology conventions and how these align with underlying network technologies.

**Table E-1** *Ethernet Service Terminology Mappings*

Term Used in GUI and This User Guide	Current MEF Equivalent Term
<b>L2VPN over MPLS Core</b>	
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)	—
<b>VPLS Over MPLS Core</b>	
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)
<b>VPLS over Ethernet Core</b>	
Ethernet Wire Service (EWS)	Ethernet Private LAN (EP-LAN)
Ethernet Relay Service (ERS)	Ethernet Virtual Private LAN (EVP-LAN)

## Setting Up the Prime Provisioning Services

To create L2VPN and VPLS 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 Provisioning 7.2 resources for L2VPN services. It contains the following sections:

- [Creating Target Devices and Assigning Roles \(N-PE or U-PE\), page E-7](#)
- [Configuring Device Settings to Support Prime Provisioning, page E-7](#)
- [Defining a Service Provider and Its Regions, page E-9](#)
- [Defining Customers and Their Sites, page E-9](#)
- [Defining VPNs, page E-9](#)
- [Creating Access Domains, page E-9](#)
- [Creating VLAN Pools, page E-10](#)
- [Creating a VC ID Pool, page E-11](#)
- [Creating Named Physical Circuits, page E-12](#)
- [Creating and Modifying Pseudowire Classes, page E-15](#)
- [Defining L2VPN Group Names for IOS XR Devices, page E-18](#)

**Note**

This section presents high-level information on Prime Provisioning services that are relevant to L2VPN. For more detailed information on setting up these and other basic Prime Provisioning services, see [Chapter 2, “Before Setting Up Prime Provisioning.”](#)

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

Every network element that Prime Provisioning manages must be defined as a device in the system. An element is any device from which Prime Provisioning 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 Provisioning

Two device settings must be configured to support the use of Prime Provisioning 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 Provisioning 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 Provisioning 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 > 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 **LDP Termination 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 Provisioning 7.2, 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 Provisioning as an IOS XR device, as follows:
- a. Create the Cisco device by choosing **Inventory > Devices > Create Cisco Device**.
  - 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 instructions in the [Cisco Prime Provisioning 7.2 Administration Guide](#).

---

**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 E-63](#).

## 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 2-39](#).

## 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 2-39](#).

## Defining 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. For detailed steps to create VPNs, see [Setting Up Logical Inventory, page 2-55](#).



---

**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. During SR creations with one vpn, you cannot use the same VPN again for continuous SR creation. It is recommended to use different VPN's for different SR's.

---

## Creating Access Domains

For L2VPN and VPLS, you create an Access Domain if you provision an Ethernet-based service and want Prime Provisioning 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 Provisioning. 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 2-39](#).

## Creating VLAN Pools

For L2VPN and VPLS, you create a VLAN pool so that Prime Provisioning 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 Provisioning 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 Provisioning service, Prime Provisioning warns you if the VLAN ID is outside the valid range of the defined VLAN pool. If so, Prime Provisioning 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 Provisioning automatically assign a VLAN to the links, perform the following steps.

---

**Step 1** Choose **Service Design > 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 E-9](#).) 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.

**Step 9** Verify the pool allocation and available states once the autopick options are selected, deployed and released.

## 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 Provisioning service, Prime Provisioning warns you if the VC ID is outside the valid range of the defined VC ID pool. If so, Prime Provisioning 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 > 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 E-13](#).
- Through the autodiscovery process. For details on how to do this, see [Creating NPC Links Through the Autodiscovery Process, page E-15](#).

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 > 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 2-55](#).

**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 > 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 Provisioning 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 Provisioning, see [Appendix B, “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 Provisioning database. NPCs are further abstracted from the discovered connectivity. For detailed steps to create NPCs using autodiscovery, see [Setting Up Logical Inventory, page 2-55](#).

## 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.



Note

The pseudowire class feature is supported on both IOS and IOS XR devices. For IOS XR devices, the pseudowire class feature is supported on IOS XR version 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 Provisioning 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. The pseudowire class is a separately defined object in the Prime Provisioning repository that can be attached to an L2VPN service policy or service request.

This section describes how to create and modify pseudowire classes. For information on how the pseudowire class is used in policies and service requests, see later sections of this guide on setting attributes for specific services.

### Creating a Pseudowire Class

To create a pseudowire class, perform the following steps.

- 
- Step 1** Choose **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 or 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 or IOS XR being used. If pseudowire class is not supported in a particular version of IOS or 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 E-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 configure 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 Provisioning 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 Provisioning, 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 Provisioning.

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 or IOS XR. It is required for IOS XR 3.6.1 and optional for IOS XR 3.7 and above.

## Modifying a Pseudowire Class

To modify (edit) a pseudowire class, perform the following steps.

- Step 1** Choose **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 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, perform the following steps.



---

**Note** A Pseudowire Class that is in use with a service request or policy cannot be deleted.

---

**Step 1** Choose **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 or 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 Provisioning, navigate to **Administration > 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 Provisioning 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 or 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 Provisioning, navigate to **Administration > 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 L2VPN Policy

This section covers the basic steps to create L2VPN policies. It contains the following subsections:

- [Overview, page E-19](#)
- [Defining L2VPN Ethernet ERS and EWS Policies, page E-20](#)
- [Defining Frame Relay Policies, page E-21](#)
- [Defining ATM Policies, page E-22](#)



## Note

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.

## Overview

You must define an L2VPN policy before you can provision a Prime Provisioning 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 Provisioning templates and data files with a policy. See [Chapter 11, “Managing Templates and Data Files”](#) for more information about using templates and data files.

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 D, “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)—See [Defining L2VPN Ethernet ERS and EWS Policies, page E-20](#).

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 Provisioning 7.2 Administration Guide](#).

- Point-to-point Ethernet Wire Service (EWS)—See [Defining L2VPN Ethernet ERS and EWS Policies, page E-20](#).

The MEF name for this service is Ethernet Private Line (EPL).

- Frame Relay over MPLS (FRoMPLS)—See [Defining Frame Relay Policies, page E-21](#).
- ATM over MPLS (ATMoMPLS)—See [Defining ATM Policies, page E-22](#).

Information on how to create policies for these services is provided in the following sections.

For information on creating L2VPN service requests, see [Managing an L2VPN Service Request](#), page E-24.

## Defining L2VPN Ethernet ERS and EWS Policies

To define an L2VPN Ethernet ERS or EWS policy (with or without a CE), perform the following steps.

---

**Step 1** Choose **Service Design > Create Policy**.

The Policy Editor window appears.

**Step 2** Choose **L2VPN** from the Policy Type drop-down list.

The Policy Editor window appears.

**Step 3** Enter a **Policy Name** for the policy.

**Step 4** Choose the **Policy Owner** for the policy.

There are three types of 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 Provisioning Role-Based Access Control (RBAC) comes into play. For example, an 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 5** 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 6** 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

This section covers the L2VPN ERS and L2VPN EWS service types.

**Step 7** Check the **CE Present** check box if you want Prime Provisioning to ask the service operator who uses this 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 Provisioning asks the service operator, during service activation, only for the U-PE or the N-PE router and customer-facing interface.

**Step 8** Click **Next**.

The Interface Type window appears.

**Step 9** Set the attributes in the Interface Type window as described in [Table E-2](#).



**Note** Attributes that appear in the GUI are determined by the type of policy being defined and whether or not a CE has been specified.

**Step 10** When you have set the attributes, click **Next** to proceed to the next window (or else click **Finish** to save the policy).

**Step 11** If you would like to use user-defined attributes within this policy, click **Next** (before clicking **Finish**). An additional window appears the policy workflow. 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 D, “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 12** If you would like to enable template association for this policy, click **Next** (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 [Chapter 11, “Managing Templates and Data Files”](#) for more information about using templates and data files. 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 13** To save the L2VPN Ethernet ERS or EWS policy, click **Finish**.

---

To create a service request based on an L2VPN Ethernet ERS or EWS policy, see [Managing an L2VPN Service Request, page E-24](#).

## Defining Frame Relay Policies

To define a Frame Relay policy (with or without a CE present), perform the following steps.

**Step 1** Choose **Service Design > Create Policy**.

The Policy Editor window appears.

**Step 2** Choose **L2VPN** from the Policy Type drop-down list.

The Policy Editor window appears.

**Step 3** Enter a **Policy Name** for the policy.

**Step 4** Choose the **Policy Owner** for the policy.

There are three types of 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 Provisioning Role-Based Access Control (RBAC) comes into play. For example, an 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 5** 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 6** Choose the **Service Type** of the L2VPN policy (in this case, Frame Relay).

**Step 7** Check or uncheck the **CE Present** check box **as required**.

**Step 8** Click **Next**.

The Interface Type window appears.

**Step 9** Set the attributes in the Interface Type window as described in [Table E-3](#).



**Note**

Attributes that appear in the GUI are determined by the type of policy being defined and whether or not a CE has been specified.

**Step 10** When you have set the attributes, click **Next** to proceed to the next window (or else click **Finish** to save the policy).

**Step 11** If you would like to use user-defined attributes within this policy, click **Next** (before clicking **Finish**).

An additional window appears the policy workflow. 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 D, "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 12** If you would like to enable template association for this policy, click **Next** (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 [Chapter 11, "Managing Templates and Data Files"](#) for more information about using templates and data files. 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 13** To save the Frame Relay policy, click **Finish**.

To create a service request based on a Frame Relay policy, see [Managing an L2VPN Service Request, page E-24](#).

## Defining ATM Policies

To define an ATM policy (with or without a CE present), perform the following steps.

**Step 1** Choose **Service Design > Create Policy**.

The Policy Editor window appears.

**Step 2** Choose **L2VPN** from the Policy Type drop-down list.

The Policy Editor window appears.

**Step 3** Enter a **Policy Name** for the policy.

**Step 4** Choose the **Policy Owner** for the policy.

There are three types of 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 Provisioning Role-Based Access Control (RBAC) comes into play. For example, a 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 5** 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 6** Choose the **Service Type** of the L2VPN policy (in this case, ATM).

**Step 7** Check or uncheck the **CE Present** check box as required.

**Step 8** Click **Next**.

The Interface Type window appears.

**Step 9** Set the attributes in the Interface Type window as described in [Table E-4](#).



---

**Note** Attributes that appear in the GUI are determined by the type of policy being defined and whether or not a CE has been specified.

---

**Step 10** When you have set the attributes, click **Next** to proceed to the next window (or else click **Finish** to save the policy).

**Step 11** If you would like to use user-defined attributes within this policy, click **Next** (before clicking **Finish**).

An additional window appears the policy workflow. 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 D, "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 12** If you would like to enable template association for this policy, click **Next** (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 [Chapter 11, "Managing Templates and Data Files"](#) for more information about using templates and data files. 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 13** To save the ATM policy, click **Finish**.

---

To create a service request based on an ATM policy, see [Managing an L2VPN Service Request, page E-24](#).

# Managing an L2VPN Service Request

This section covers the basic steps to provision an ERS, EWS, ATM, or Frame Relay L2VPN service. It contains the following subsections:

- [Overview, page E-24](#)
- [Creating an L2VPN Service Request, page E-25](#)
- [Using Templates and Data Files with an L2VPN Service Request, page E-33](#)
- [Saving an L2VPN Service Request, page E-33](#)
- [Modifying an L2VPN Service Request, page E-33](#)

## Overview

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. To create a service request, a Service Policy must already be defined, as described in [Creating an L2VPN Policy, page E-19](#).



### 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 E-19](#).

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.

You can also associate Prime Provisioning templates and data files with a service request. See [Chapter 11, “Managing Templates and Data Files”](#) for more information about using templates and data files.

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 D, “Adding Additional Information to Services.”](#)

The following steps are involved in creating a service request for Layer 2 connectivity between customer sites:

1. Choose a CE Topology for ERS (EVPL)/Frame Relay/ATM services.
2. Choose the endpoints (CE and PE) that must be connected. For each end-to-end Layer 2 connection, Prime Provisioning creates an end-to-end wire object in the repository for the service request.
3. Choose a CE or PE interface.
4. Choose a Named Physical Circuit (NPC) for the CE or PE.
5. Edit the end-to-end connection.
6. Edit the link attributes.
7. Associate templates and data files to devices in the service request. (Optional)

For sample configlets for L2VPN scenarios, see [Sample Configlets, page E-63](#).



## Creating an L2VPN Service Request

For information on creating specific types of L2VPN service requests, see the following sections:

- [Creating an ERS, ATM, or Frame Relay L2VPN Service Request with a CE, page E-25.](#)
- [Creating an ERS, ATM, or Frame Relay L2VPN Service Request without a CE, page E-27.](#)
- [Creating an EWS L2VPN Service Request with a CE, page E-30.](#)
- [Creating an EWS L2VPN Service Request without a CE, page E-31.](#)

### Creating an ERS, ATM, or Frame Relay L2VPN Service Request with a CE

To create an ERS, ATM, or Frame Relay L2VPN service request with a CE present, perform the following steps.

---

**Step 1** Choose **Operate > Create Service Request**.

The Service Request Editor window appears.

**Step 2** From the policy picker choose an appropriate policy from the policies previously created.

The L2VPN Service Request Editor window appears.

**Step 3** 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 Provisioning automatically creates six links with full mesh topology. With hub and spoke topology, however, Prime Provisioning creates only three links.

---

**Step 4** 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 5** Click **Select CE** in the CE column.

The Select CPE Device window appears. This window displays the list of currently defined CEs.

- From the **Show CPEs with** drop-down list, you can display CEs by Customer Name, by Site, or by Device Name.
- You can use the **Find** button to either search for a specific CE, or to refresh the display.
- You can set the **Rows per page** to 5, 10, 20, 30, 40, or All.

**Step 6** In the Select column, choose a CE for the L2VPN link.

**Step 7** Click **Select**.

The Service Request Editor window appears displaying the name of the selected CE in the CE column.

**Step 8** Choose the CE interface from the interface picker.



**Note**

When you provision an L2VPN ERS (EVPL) service, when you choose a UNI for a particular device, Prime Provisioning 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 Provisioning 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 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 than 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 10** 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 11** Continue to specify additional CEs, as in previous steps.

Prime Provisioning creates the links between CEs based on the Topology that you chose.

**Step 12** Click **OK**.

For ERS (EVPL), ATM, and Frame Relay, the EndToEndWire window appears.

**Step 13** 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 14** Choose a **VPN Name** and click **Select**.

The L2VPN Service Request Editor window appears with the VPN name displayed.

**Step 15** If necessary, click **Add AC** in the Attachment Circuit2 (AC2) column, and repeat previous steps for AC2.

The EndToEndWire window displays the complete end-to-end wire.

**Step 16** Specify remaining items in the EndToEndWire window as necessary for your configuration. Notes:

- 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 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 Provisioning 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 Provisioning validates if the entered value is available or allocated. If the entered value has been already allocated, Prime Provisioning 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 Provisioning 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 17** When you are finished editing the end-to-end wires, click **Save**.

The service request is created and saved into Prime Provisioning.

---

For additional information on working with L2VPN service requests, see the following sections:

- [Using Templates and Data Files with an L2VPN Service Request, page E-33](#)
- [Saving an L2VPN Service Request, page E-33](#)
- [Modifying an L2VPN Service Request, page E-33](#)
- [Deploying, Monitoring, and Auditing Service Requests, page E-44.](#)

## Creating an ERS, ATM, or Frame Relay L2VPN Service Request without a CE

To create an ERS, ATM, or Frame Relay L2VPN service request without a CE present, perform the following steps.

---

**Step 1** Choose **Operate > Create Service Request**.

The Service Request Editor window appears.

**Step 2** From the policy picker choose an appropriate policy from the policies previously created.

The L2VPN Service Request Editor window appears.

**Step 3** 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 Provisioning automatically creates six links with full mesh topology. With hub and spoke topology, however, Prime Provisioning creates only three links.

---

**Step 4** Click **Add Link**.

**Step 5** Specify the N-PE/PE-AGG/U-PE endpoints, as covered in the following steps.

**Step 6** 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 7** In the **Select** column, choose the PE device name for the L2VPN link.

**Step 8** 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 9** Choose the UNI interface from the interface picker.

**Step 10** To choose the UNI interface, click on the toggle button in the **Select One** field of the UNI Interface column.

The Interface Selection window appears. This window 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.

**Step 11** Choose the UNI interface by clicking the radio button next to the interface name.




---

**Note** When you provision an L2VPN ERS (EVPL) service, when you choose a UNI for a particular device, Prime Provisioning 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 12** 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 13** Choose the name of the NPC from the **Select** column.

**Step 14** 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 15** 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 16** After you specify all the PEs, Prime Provisioning creates the links between PEs based on the Topology that you chose.
- Step 17** Click **OK**.  
For ERS (EVPL), ATM, and Frame Relay, the EndToEndWire window appears.
- Step 18** 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 19** 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.
  - 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 10-12](#), 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 20** When you are finished editing the end-to-end wires, click **Save**.  
The service request is created and saved into Prime Provisioning.

---

For additional information on working with L2VPN service requests, see the following sections:

- [Using Templates and Data Files with an L2VPN Service Request, page E-33](#)
- [Saving an L2VPN Service Request, page E-33](#)
- [Modifying an L2VPN Service Request, page E-33](#)
- [Deploying, Monitoring, and Auditing Service Requests, page E-44.](#)

## Creating an EWS L2VPN Service Request with a CE

To create an EWS L2VPN service request with a CE present, perform the following steps.

- 
- Step 1** Choose **Operate > Create Service Request**.
- The Service Request Editor window appears.
- Step 2** From the policy picker choose an appropriate policy from the policies previously created.
- The L2VPN Service Request Editor window appears.
- Step 3** 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 4** Choose a **VPN Name** in the Select column.
- Step 5** Click **Select**.
- The L2VPN Service Request Editor window appears with the VPN name displayed.
- Step 6** 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 7** Click **Add AC** in the Attachment Circuit1 (AC1) column.
- The Customer and Link Selection window appears.
- Step 8** 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 9** In the Select column, choose a CE for the L2VPN link.
- Step 10** Click **Select**.
- Step 11** In the Customer and Link Selection window, choose a CE interface from the interface picker.
- Step 12** 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 than 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 13** Click **OK**.  
The EndToEndWire window appears displaying the name of the selected CE in the AC1 column.
- Step 14** 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.
- Step 15** Click **OK**.
- Step 16** Repeat steps (as above) for **AC2**.
- Step 17** When you are finished editing the end-to-end wires, click **Save**.  
The service request is created and saved in Prime Provisioning.
- 

For additional information on working with L2VPN service requests, see the following sections:

- [Using Templates and Data Files with an L2VPN Service Request, page E-33](#)
- [Saving an L2VPN Service Request, page E-33](#)
- [Modifying an L2VPN Service Request, page E-33](#)
- [Deploying, Monitoring, and Auditing Service Requests, page E-44](#).

## Creating an EWS L2VPN Service Request without a CE

To create an EWS L2VPN service request without a CE present, perform the following steps.

---

- Step 1** Choose **Operate > Create Service Request**.  
The Service Request Editor window appears.
- Step 2** From the policy picker choose an appropriate policy from the policies previously created.  
The L2VPN Service Request Editor window appears.
- Step 3** 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 4** Choose a **VPN Name** in the Select column.
- Step 5** Click **Select**.  
The EndToEndWire window appears with the VPN name displayed.
- Step 6** Click **Add AC** in the Attachment Circuit 1(AC1) column.  
The Customer and Link Selection window appears.
- 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.
- From the **Show PEs with** drop-down list, you can display PEs by Customer Name, by Site, or by Device Name.
  - You can use the **Find** button to either search for a specific PE, or to refresh the display.
  - You can set the **Rows per page** to 5, 10, 20, 30, 40, or All.
- Step 8** In the Select column, choose a PE for the L2VPN link.

**Step 9** Click **Select**.

The Customer and Link Selection window appears.

**Step 10** Choose the UNI interface from the interface picker.

**Step 11** To choose the UNI interface, click on the toggle button in the **Select One** field of the UNI Interface column.

The Interface Selection window appears. This window 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.

**Step 12** Choose the UNI interface by clicking the radio button next to the interface name.

**Step 13** If the PE role type is N-PE, the columns Circuit Selection and Circuit Details are disabled. In this case, skip to Step 18.

**Step 14** 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 15** If applicable, choose the name of the NPC from the Select column.

**Step 16** 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 17** Click **OK**.

The L2VPN Service Request window appears displaying the name of the selected PE in the Attachment Circuit1 (AC1) column.

**Step 18** Click the **Edit** link in the AC1 Attributes and edit the attributes, if desired.

**Step 19** Repeat steps (as above) for Attachment Circuit2.

**Step 20** 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 21** When you are finished editing the end-to-end wires, click **Save**.

The service request is created and saved in Prime Provisioning.



For additional information on working with L2VPN service requests, see the following sections:

- [Using Templates and Data Files with an L2VPN Service Request, page E-33](#)
- [Saving an L2VPN Service Request, page E-33](#)
- [Modifying an L2VPN Service Request, page E-33](#)
- [Deploying, Monitoring, and Auditing Service Requests, page E-44.](#)

## Using Templates and Data Files with an L2VPN Service Request

The template mechanism in Prime Provisioning 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 [Chapter 11, “Managing Templates and Data Files”](#) for more information about using templates and data files.

## Saving an 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.
- 

## Modifying an 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 the attachment circuits.

Perform the following steps.

- 
- Step 1** Choose **Operate > 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 Provisioning 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 Provisioning validates if the entered value is available or allocated. If the entered value has been already allocated, Prime Provisioning 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 Provisioning 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 10-12](#) 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 11-1](#). For more information on how to use templates and data files in service requests, see [Chapter 11, “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 Provisioning to append the template generated CLI to the regular Prime Provisioning (non-template) CLI. Prepend is the reverse and does not append the template to the Prime Provisioning 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.
- 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**.
- 

## Creating a VPLS Policy

This section contains the basic steps to create a VPLS policy. It contains the following subsections:

- [Overview, page E-35](#)
- [Defining a VPLS Policy, page E-36](#)



### Note

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.

---

## Overview

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 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.

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 Provisioning templates and data files with a policy. See [Chapter 11, “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 D, “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 Provisioning 7.2 Administration Guide*.
- Ethernet Multipoint Service (EMS). The MEF name for EMS is Ethernet Private LAN (EP-LAN).

Information on how to create policies for these services is provided in the following sections.



Note

For a general overview of VPLS support in Prime Provisioning, see the chapter “Layer 2 Concepts” in the *Cisco Prime Provisioning 7.2 Administration Guide*.

## Defining a VPLS Policy

To define a VPLS policy, perform the following steps.



Note

This is a general workflow that covers all core types and service types.

- Step 1** Choose **Service Design > Create Policy**.  
The Policy Editor window appears.
- Step 2** Choose **VPLS** from the Policy Type drop-down list.  
The Policy Editor window appears.
- Step 3** Enter a **Policy Name** for the VPLS policy.
- Step 4** 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 Provisioning 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 5** Click **Select** to choose the owner of the VPLS policy.

The policy owner was established when you created customers or providers during Prime Provisioning setup. If the ownership is global, the Select function does not appear.

**Step 6** Choose the **Core Type** of the VPLS policy per your requirements.

There are two core types for VPLS policies:

- MPLS—running on an IP network
- Ethernet—all PEs are on an Ethernet provider network

**Step 7** Choose the **Service Type** of the VPLS policy per your requirements.

There are two service types for VPLS policies:

- Ethernet Relay Multipoint Service (ERMS)
- Ethernet Multipoint Service (EMS)

**Step 8** Check the **CE Present** check box if you want Prime Provisioning 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 Provisioning asks the service operator, during service activation, only for the PE router and customer-facing interface.

**Step 9** Click **Next**.

The Interface Type window appears.

**Step 10** Set the attributes in the Interface Type window as described in [Table E-5](#).



**Note** Attributes that appear in the GUI are determined by the type of policy being defined and whether or not a CE has been specified.



**Note** The VC ID is mapped from the VPN ID. By default, Prime Provisioning 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 2-55](#).

**Step 11** When you have set the attributes, click **Next** to proceed to the next window (or else click **Finish** to save the policy).

**Step 12** If you would like to use user-defined attributes within this policy, click **Next** (before clicking **Finish**).

An additional window appears the policy workflow. 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 D, “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** If you would like to enable template association for this policy, click **Next** (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 [Chapter 11, “Managing Templates and Data Files”](#) for more information about using templates and data files. 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 14** To save the VPLS policy, click **Finish**.
- 

To create a service request based on a VPLS policy, see [Managing a VPLS Service Request, page E-38](#).

## Managing a VPLS Service Request

This section contains the basic steps to provision a VPLS service. It contains the following subsections:

- [Overview, page E-38](#)
- [Creating a VPLS Service Request, page E-39](#)
- [Using Templates and Data Files with a VPLS Service Request, page E-43](#)
- [Saving the VPLS Service Request, page E-43](#)
- [Modifying the VPLS Service Request, page E-44](#)

### Overview

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.

To create a service request, a service policy must already be defined, as described in [Creating a VPLS Policy, page E-35](#). 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.

You can also associate Prime Provisioning templates and data files with a service request. See [Chapter 11, “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 D, “Adding Additional Information to Services.”](#)

The following steps are involved in creating a service request for Layer 2 connectivity between customer sites:

1. Choose a VPLS policy.
2. Choose a VPN. For more information, see [Defining VPNs, page E-9](#).
3. Add a link.

4. Choose a CE or UNI interface.
5. Choose a Named Physical Circuit (NPC) if more than one NPC exists from the CE or the UNI interface.
6. Edit the link attributes.

For sample configlets for VPLS scenarios, see [Sample Configlets, page E-63](#).

## Creating a VPLS Service Request

For information on creating specific types of VPLS service requests, see the following sections:

- [Creating a VPLS Service Request with a CE, page E-39](#)
- [Creating a VPLS Service Request without a CE, page E-41](#)

### Creating a VPLS Service Request with a CE

To create a VPLS service request with a CE present, perform the following steps.



**Note** 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.

**Step 1** Choose **Operate > Create Service Request**.

The Service Request Editor window appears.

**Step 2** From the policy picker, choose a VPLS policy from the policies previously created (see [Creating a VPLS Policy, page E-35](#)).

The new service request inherits all the properties of that VPLS policy, such as all the editable and noneditable features and preset attributes.

The Edit VPLS Link window appears.

**Step 3** 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 Provisioning 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 2-55](#).

**Step 4** Choose a **VPN Name** in the Select column.

**Step 5** Click **Select**.

The Edit VPLS Link window appears with the VPN name displayed.

**Step 6** Click **Add Link**.

The window updates, allowing you specify the CE endpoints.

**Step 7** 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 8** 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 9** In the Select column, choose a CE for the VPLS link.

**Step 10** Click **Select**.

The Edit VPLS Link window appears displaying the name of the selected CE in the CE column.

**Step 11** Choose the CE interface from the interface picker.




---

**Note** When you provision an ERMS (EVP-LAN) service (and when you choose a UNI for a particular device), Prime Provisioning 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 12** 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 13** Choose the name of the NPC from the Select column.

**Step 14** 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 15** 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 16** The Circuit ID is created automatically, based on the VLAN data for the circuit.

**Step 17** 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.

**Step 18** Set attributes in this window per your requirements.




---

**Note** For more information on setting attributes in this window, see the corresponding attributes for the VPLS policy as described in [Table E-5](#).

---

**Step 19** Continue to specify additional CEs, as in previous steps, if desired.

**Step 20** Click **OK**.



**Step 21** Click **Save**.

The service request is created and saved into Prime Provisioning.

---

For additional information on working with VPLS service requests, see the following sections:

- [Using Templates and Data Files with a VPLS Service Request, page E-43](#)
- [Saving the VPLS Service Request, page E-43](#)
- [Modifying the VPLS Service Request, page E-44](#)
- [Deploying, Monitoring, and Auditing Service Requests, page E-44](#)

## Creating a VPLS Service Request without a CE

To create a VPLS service request without a CE present, perform the following steps.



**Note**

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.

---

**Step 1** Choose **Operate > Create Service Request**.

The Service Request Editor window appears.

**Step 2** From the policy picker, choose a VPLS policy from the policies previously created (see [Creating a VPLS Policy, page E-35](#)).

The new service request inherits all the properties of that VPLS policy, such as all the editable and noneditable features and preset attributes.

The Edit VPLS Link window appears.

**Step 3** 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 Provisioning 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 2-55](#).

---

**Step 4** Choose a **VPN Name** in the Select column.

**Step 5** Click **Select**.

The Edit VPLS Link window appears with the VPN name displayed.

**Step 6** 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 7** 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 8** 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 9** In the **Select** column, choose the PE device name for the VPLS link.

**Step 10** 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 11** To choose the UNI interface, click on the toggle button in the **Select One** field of the UNI Interface column.

The Interface Selection window appears. This window 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.

**Step 12** Choose the UNI interface by clicking the radio button next to the interface name.




---

**Note** When you provision an ERMS service (and when you choose a UNI for a particular device), Prime Provisioning 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 13** 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 14** Choose the name of the NPC from the **Select** column.

**Step 15** 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 16** 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 17** 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 the corresponding attributes for the VPLS policy as described in [Table E-5](#).

---

**Step 18** Continue to specify additional PEs, as in previous steps, if desired.

**Step 19** Click **Save**.

The service request is created and saved into Prime Provisioning.

---

For additional information on working with VPLS service requests, see the following sections:

- [Using Templates and Data Files with a VPLS Service Request, page E-43](#)
- [Saving the VPLS Service Request, page E-43](#)
- [Modifying the VPLS Service Request, page E-44](#)
- [Deploying, Monitoring, and Auditing Service Requests, page E-44](#)

## Using Templates and Data Files with a VPLS Service Request

The template mechanism in Prime Provisioning 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 [Chapter 11, “Managing Templates and Data Files”](#) for more information about using templates and data files.

## 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 requests 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.

**Step 3** If you are ready to deploy the service request, see [Deploying, Monitoring, and Auditing Service Requests, page E-44](#).

---

## Modifying the VPLS Service Request

To modify a VPLS service request, perform the following steps.

- 
- Step 1 Choose **Operate > 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.
  - 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.
  - Step 7 Click **OK**.
- 

## Deploying, Monitoring, and Auditing Service Requests

To apply EVC, L2VPN, or VPLS policies to network devices, you must deploy the service request. When you deploy a service request, Prime Provisioning compares the device information in the Repository (the Prime Provisioning database) with the current device configuration and generates a configlet. Additionally, you can perform various monitoring and auditing tasks on service requests. Information about common tasks that apply to all types of Prime Provisioning service requests is provided in [Chapter 10, “Managing Service Requests.”](#)

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 > 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 Provisioning create the minimum VLAN list. This is the default.
- **abort** to have Prime Provisioning 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 Provisioning allow all VLANs.

Step 6 Click **Set Property**.

---

## 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:

- [VLAN Translation Overview, page E-45](#)
- [Setting Up VLAN Translation, page E-45](#)
- [Platform-Specific Usage Notes, page E-49](#)



Note

For helpful information to be aware of before you create policies and services using VLAN translation, review [Platform-Specific Usage Notes, page E-49](#).

---

## 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, page E-46](#)
- [Creating a Service Request, page E-46](#)
- [Modifying a Service Request, page E-48](#)

- [Deleting a Service Request, page E-48](#)

## 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 E-46](#).
- **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 E-46](#).

## 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 Provisioning does an integrity check before saving the service request. For 1:1 VLAN translation, Prime Provisioning 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 Provisioning does an integrity check before saving the service request. For 2:1 VLAN translation, Prime Provisioning 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 Provisioning does not know beforehand whether a particular line card will accept or reject the VLAN translation CLI commands. In this case, Prime Provisioning attempts to push down the commands and the deployment fails. An **Invalid** status means Prime Provisioning 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 Provisioning 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.

# Policy and Service Request Attributes Reference Tables

This section provides reference information for attributes appearing in windows in L2VPN and VPLS policies and service requests. To find attributes and descriptions refer to the appropriate section for the service:

- [L2VPN Service Attributes, page E-50](#)
- [VPLS Service Attributes, page E-58](#)

## L2VPN Service Attributes

This section describes attributes available in the L2VPN policy workflow:

- [Table E-2, “L2VPN Ethernet ERS and EWS Interface Attributes,” on page 50](#)
- [Table E-3, “Frame Relay Interface Type Attributes,” on page 54](#)
- [Table E-4, “ATM Interface Type Attributes,” on page 56](#)

**Table E-2** L2VPN Ethernet ERS and EWS Interface Attributes

Attribute	Description
Standard UNI Port	<p>Check the 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.</p> <ul style="list-style-type: none"> <li>• The <b>Standard UNI Port</b> attribute will be unavailable within service requests based on this policy if the UNI is on an N-PE device running IOS XR.</li> <li>• 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 <b>Standard UNI Port flag</b> for the EWS (EPL) side of the connection and uncheck the standard UNI flag for the NNI side of the connection.</li> <li>• In the case of hybrid services, UNI on an N-PE running IOS XR is not supported.</li> </ul>

Table E-2 L2VPN Ethernet ERS and EWS Interface Attributes (continued)

Attribute	Description
Interface Type	<p>Choose an Interface Type from the drop-down list. You can choose a particular interface on a CE, U-PE, or N-PE interface depending on how you have set up the policy and based on the service provider's POP design. The interfaces are:</p> <ul style="list-style-type: none"> <li>• <b>ANY</b> (Any interface can be chosen.)</li> <li>• <b>Port-Channel</b> (A bundle of ports that share the same characteristics—this gives the service provider the ability to aggregate bandwidth and protection.)</li> <li>• <b>Ethernet</b></li> <li>• <b>FastEthernet</b></li> <li>• <b>GE-WAN</b></li> <li>• <b>GigabitEthernet</b></li> <li>• <b>TenGigabitEthernet</b></li> <li>• <b>TenGigE</b></li> </ul> <p>The value defined here functions as a filter to restrict the interface types an operator can see during L2VPN service request creation.</p>
Interface Format	<p>Enter a slot number/port number for the interface (for example, <b>1/0</b> 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.</p>
Encapsulation	<p>Choose a type. The choices are:</p> <ul style="list-style-type: none"> <li>• <b>DOT1Q</b></li> <li>• <b>DEFAULT</b></li> </ul> <p>If <b>DEFAULT</b> is the encapsulation type, Prime Provisioning shows another field for the UNI port type. If the Interface Type is ANY, Prime Provisioning will not ask for an <b>Encapsulation</b> type in the policy.</p>
UNI Shutdown	<p>Check the 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.</p>
Keep Alive	<p>Check the box to configure keepalives on the UNI port. By default, this check box is unchecked, which causes the command <b>no keepalive</b> 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.</p>
ANY	<p>Check the 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.</p>
UNI	<p>Check the 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.</p>
VLAN ID AutoPick	<p>Check the box if you want Prime Provisioning 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.</p>
VC ID AutoPick	<p>Check the box if you want Prime Provisioning 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.</p>
VLAN NAME (optional)	<p>Enter 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.</p>

Table E-2 L2VPN Ethernet ERS and EWS Interface Attributes (continued)

Attribute	Description
Use PseudoWireClass	Check the box to enable the selection of a pseudowire class. If the check box is checked, an additional attribute, <b>PseudoWireClass</b> , appears in the GUI. Click the <b>Select</b> button of PseudoWireClass attribute to choose a pseudowire class previously created in Prime Provisioning. The pseudowire class name is used for provisioning pw-class commands on IOS and IOS XR devices. See <a href="#">Creating and Modifying Pseudowire Classes, page E-15</a> , for additional information on pseudowire class support.
L2VPN Group Name	Choose a name from the drop-down list. The choices are: <ul style="list-style-type: none"> <li>• <b>ISC</b></li> <li>• <b>VPNSC</b></li> </ul> This attribute is used for provisioning the L2VPN group name on IOS XR devices. 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 <a href="#">Defining L2VPN Group Names for IOS XR Devices, page E-18</a> .
E-Line Name	Enter the point-to-point (p2p) E-line name. This attribute is only applicable for IOS XR devices. If no value is specified for the <b>p2p</b> name, Prime Provisioning 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.
Link Media (optional)	Enter None, auto-select, rj45, or sfp. Usage notes: <ul style="list-style-type: none"> <li>• The default is None.</li> <li>• When this attribute is used, a new CLI will be generated in the UNI interface to define the media type.</li> <li>• The Link Media attribute is supported only for ME3400 platforms.</li> </ul>
Link Speed (optional)	Enter None, 10, 100, 1000, Auto, or nonegotiate.
Link Duplex (optional)	Enter None, Full, Half, or Auto.
Use Existing ACL Name	Check the box if you want assign your own named access list to the port. By default, this box is unchecked and Prime Provisioning automatically assigns a MAC-based ACL on the customer facing UNI port, based on values you enter in <b>UNI MAC addresses</b> (below).
Port-Based ACL Name	Enter a Port-Based ACL Name (if you checked the <b>Use Existing ACL Name</b> check box, as mentioned above).  Prime Provisioning 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.
UNI MAC addresses	Enter one or more Ethernet MAC addresses. This selection is present only if you uncheck the <b>Use Existing ACL Name</b> check box. Click the <b>Edit</b> 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.
UNI Port Type	Choose a type. The choices are: <ul style="list-style-type: none"> <li>• <b>Access Port</b></li> <li>• <b>Trunk with Native VLAN</b></li> </ul> Enter a UNI Port Type only if the encapsulation type is DEFAULT.

Table E-2 L2VPN Ethernet ERS and EWS Interface Attributes (continued)

Attribute	Description
UNI Port Security	<p>Check the box if you want to provision port security-related CLIs to the UNI port by controlling the MAC addresses that are allowed to go through the interface.</p> <ul style="list-style-type: none"> <li>• For <b>Maximum Number of MAC address</b>, enter the number of MAC addresses allowed for port security.</li> <li>• For <b>Aging</b>, enter the length of time the MAC address can stay on the port security table.</li> <li>• For <b>Violation Action</b>, choose what action will occur when a port security violation is detected: <ul style="list-style-type: none"> <li>– <b>PROTECT</b>—Drops packets with unknown source addresses until a sufficient number of secure MAC addresses are removed to drop below the maximum value.</li> <li>– <b>RESTRICT</b>—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.</li> <li>– <b>SHUTDOWN</b>—Puts the interface into the error-disabled state immediately and sends an SNMP trap notification.</li> </ul> </li> <li>• In the <b>Secure MAC Addresses</b> field, enter one or more Ethernet MAC addresses.</li> </ul>
Enable Storm Control	<p>Check the 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.</p>
Protocol Tunnelling	<p>Check the 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:</p> <ul style="list-style-type: none"> <li>• <b>Enable cdp</b>—Enable Layer 2 tunnelling on Cisco Discover Protocol (CDP).</li> <li>• <b>cdp shutdown threshold</b>—Enter the number of packets per second to be received before the interface is shut down.</li> <li>• <b>cdp drop threshold</b>—Enter the number of packets per second to be received at which point the interface will start dropping CDP packets.</li> <li>• <b>Enable vtp</b>—Enable Layer 2 tunnelling on VLAN Trunk Protocol (VTP).</li> <li>• <b>vtp shutdown threshold</b>—Enter the number of packets per second to be received before the interface is shut down.</li> <li>• <b>vtp drop threshold</b>—Enter the number of packets per second to be received at which point the interface will start dropping VTP packets.</li> <li>• <b>Enable stp</b>—Enable Layer 2 tunnelling on Spanning Tree Protocol (STP).</li> <li>• <b>stp shutdown threshold</b>—Enter the number of packets per second to be received before the interface is shut down.</li> <li>• <b>stp drop threshold</b>—Enter the number of packets per second to be received at which point the interface will start dropping STP packets.</li> <li>• <b>Recovery Interval</b>—Enter the amount of time, in seconds, to wait before recovering a UNI port.</li> </ul>

Table E-2 L2VPN Ethernet ERS and EWS Interface Attributes (continued)

Attribute	Description
N-PE Pseudo-wire On SVI	Check the 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. The <b>N-PE Pseudo-wire on SVI</b> attribute will be unavailable within service requests based on this policy for devices running IOS XR.
MTU Size	Enter the 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 Provisioning 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 Provisioning 6.3, different platforms support different ranges. <ul style="list-style-type: none"> <li>For the 3750 and 3550 platforms, the MTU range is 1500-1546.</li> <li>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 Provisioning 6.3 uses 9216 in both cases.</li> <li>For the 7600 SVI (interface VLAN), the MTU size is 1500-9216.</li> </ul>
VLAN Translation	Specify the type of VLAN Translation for this policy by clicking the appropriate radio button. The choices are: <ul style="list-style-type: none"> <li><b>No</b>—No VLAN translation is performed. (This is the default.)</li> <li><b>1:1</b>—1:1 VLAN translation.</li> <li><b>2:1</b>—2:1 VLAN translation.</li> </ul> For detailed coverage of setting up VLAN translation, see <a href="#">Setting Up VLAN Translation for L2VPN ERS (EVPL) Services</a> , page E-45.
PW Tunnel Selection	Check the 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 Provisioning 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 Provisioning does not check the validity of the value. That is, Prime Provisioning does not verify the existence of the tunnel.  The <b>PW Tunnel Selection</b> attribute will be unavailable within service requests based on this policy for devices running IOS XR.

Table E-3 Frame Relay Interface Type Attributes

Attribute	Description
UNI Shutdown	Check the 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.

Table E-3 Frame Relay Interface Type Attributes (continued)

Attribute	Description
Interface Type	<p>Choose the type for the PE or CE from the drop-down list. The choices are:</p> <ul style="list-style-type: none"> <li>• <b>ANY</b></li> <li>• <b>Serial</b></li> <li>• <b>MFR</b></li> <li>• <b>POS</b></li> <li>• <b>Hssi</b></li> <li>• <b>BRI</b></li> </ul>
Interface Format	<p>Enter the slot number/port number for the interface (for example, <b>1/0</b> 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.</p>
Encapsulation Type	<p>Choose the PE or CE encapsulation type. The choices are:</p> <ul style="list-style-type: none"> <li>• <b>FRAME RELAY</b></li> <li>• <b>FRAME RELAY IETF</b></li> </ul> <p>If the Interface Type is ANY, Prime Provisioning will not ask for an <b>Encapsulation</b> type in the policy.</p>
Use PseudoWireClass	<p>Check the box to enable the selection of a pseudowire class. If the check box is checked, an additional attribute, <b>PseudoWireClass</b>, appears in the GUI. Click the <b>Select</b> button of PseudoWireClass attribute to choose a pseudowire class previously created in Prime Provisioning. The pseudowire class name is used for provisioning pw-class commands on IOS and IOS XR devices. See <a href="#">Creating and Modifying Pseudowire Classes, page E-15</a>, for additional information on pseudowire class support.</p>
L2VPN Group Name	<p>Choose a name from the drop-down list. The choices are:</p> <ul style="list-style-type: none"> <li>• <b>ISC</b></li> <li>• <b>VPNSC</b></li> </ul> <p>This attribute is used for provisioning the L2VPN group name on IOS XR devices. 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 <a href="#">Defining L2VPN Group Names for IOS XR Devices, page E-18</a>.</p>
E-Line Name	<p>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 Provisioning 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.</p>

Table E-3 Frame Relay Interface Type Attributes (continued)

Attribute	Description
PW Tunnel Selection	<p>Check the 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</p> <p>Subsequently, when you create a service request based on this policy, you must specify the TE tunnel ID in a field provided. Prime Provisioning 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 Provisioning does not check the validity of the value. That is, Prime Provisioning does not verify the existence of the tunnel.</p>

Table E-4 ATM Interface Type Attributes

Attribute	Description
Transport Mode	<p>Choose the Transport Mode from the drop-down list. The choices are:</p> <ul style="list-style-type: none"> <li>• <b>VP</b>—Virtual path mode. This is the default.</li> <li>• <b>VC</b>—Virtual circuit mode.</li> <li>• <b>PORT</b>—Port mode. (Only supported for the IOS XR 3.7 platform.) Usage notes: <ul style="list-style-type: none"> <li>– If you choose PORT as the transport mode, the attributes <b>ATM VCD/Sub-interface #</b> and <b>ATM VPI</b> will be disabled in the Link Attributes window of the service request based on this policy.</li> <li>– If you choose PORT as the transport mode, three attributes for setting timer values will appear in the Link Attributes window of the service request based on this policy. These attributes are <b>Timer1</b>, <b>Timer2</b>, and <b>Timer3</b>. 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.</li> <li>– 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 <b>Maximum no. of cells to be packed</b> and <b>Cell packing timer</b>. This feature is supported only for an N-PE as a UNI device.</li> </ul> </li> </ul>
Interface Type	<p>Choose the CE or PE Interface Type from the drop-down list. The choices are:</p> <ul style="list-style-type: none"> <li>• <b>ANY</b></li> <li>• <b>ATM</b></li> <li>• <b>Switch</b></li> </ul>
Interface Format	<p>The slot number/port number for the interface (for example, <b>1/0</b> 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.</p>



Table E-4 ATM Interface Type Attributes (continued)

Attribute	Description
CE Encapsulation	<p>Choose the CE encapsulation type. The choices are:</p> <ul style="list-style-type: none"> <li>• <b>AAL5SNAP</b></li> <li>• <b>AAL5MUX</b></li> <li>• <b>AAL5NLPID</b></li> <li>• <b>AAL2</b></li> </ul> <p>If the Interface Type is ANY, Prime Provisioning will not ask for an <b>Encapsulation</b> type in the policy.</p>
PE Encapsulation	<p>Choose a PE encapsulation type. The choices are:</p> <ul style="list-style-type: none"> <li>• <b>AAL5SNAP</b></li> <li>• <b>AAL5MUX</b></li> <li>• <b>AAL5NLPID</b></li> <li>• <b>AAL5</b></li> <li>• <b>AAL0</b></li> </ul> <p>If the Interface Type is ANY, Prime Provisioning will not ask for an <b>Encapsulation</b> type in the policy.</p>
UNI Shutdown	<p>Check the 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.</p>
Use PseudoWireClass	<p>Check the box to enable the selection of a pseudowire class. If the check box is checked, an additional attribute, <b>PseudoWireClass</b>, appears in the GUI. Click the <b>Select</b> button of PseudoWireClass attribute to choose a pseudowire class previously created in Prime Provisioning. The pseudowire class name is used for provisioning pw-class commands on IOS and IOS XR devices. See <a href="#">Creating and Modifying Pseudowire Classes, page E-15</a>, for additional information on pseudowire class support.</p>
L2VPN Group Name	<p>Choose a name from the drop-down list. The choices are:</p> <ul style="list-style-type: none"> <li>• <b>ISC</b></li> <li>• <b>VPNSC</b></li> </ul> <p>This attribute is used for provisioning the L2VPN group name on IOS XR devices.</p> <p>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 <a href="#">Defining L2VPN Group Names for IOS XR Devices, page E-18</a>.</p>
E-Line Name	<p>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 Provisioning 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.</p>

**Table E-4** ATM Interface Type Attributes (continued)

Attribute	Description
PW Tunnel Selection	<p>Check the 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</p> <p>Subsequently, when you create a service request based on this policy, you must specify the TE tunnel ID in a field provided. Prime Provisioning 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 Provisioning does not check the validity of the value. That is, Prime Provisioning does not verify the existence of the tunnel.</p>

## VPLS Service Attributes

This section describes attributes available in the VPLS policy workflow:

- [Table E-5, “Interface Type Attributes \(for VPLS\),” on page 58](#)

**Table E-5** Interface Type Attributes (for VPLS)

Attribute	Description
Interface Type	<p>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 depending on how you have set up the policy and based on the service provider’s POP design. The interfaces are:</p> <ul style="list-style-type: none"> <li>• <b>ANY</b> (Any interface can be chosen.)</li> <li>• <b>Port-Channel</b> (A bundle of ports that share the same characteristics—this gives the service provider the ability to aggregate bandwidth and protection.)</li> <li>• <b>Ethernet</b></li> <li>• <b>FastEthernet</b></li> <li>• <b>GE-WAN</b></li> <li>• <b>GigabitEthernet</b></li> <li>• <b>TenGigabitEthernet</b></li> <li>• <b>TenGigE</b></li> </ul> <p>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.</p>
Interface Format	<p>Enter the slot number/port number for the interface (for example, <b>1/0</b> 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.</p>

Table E-5 Interface Type Attributes (for VPLS) (continued)

Attribute	Description
Encapsulation	Choose a type. The choices are: <ul style="list-style-type: none"> <li>• <b>DOT1Q</b></li> <li>• <b>DEFAULT</b></li> </ul> If <b>DEFAULT</b> is the encapsulation type, Prime Provisioning shows another field for the UNI port type.
Standard UNI Port	Check the 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.
UNI Shutdown	Check the 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.
Keep Alive	Check the box to configure keepalives on the UNI port. By default, this check box is unchecked, which causes the command <b>no keepalive</b> 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.
ANY	Check the 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.
UNI	Check the 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.
UNI MAC addresses	Enter one or more Ethernet MAC addresses. This selection is present only if you uncheck the <b>Use Existing ACL Name</b> check box. Click the <b>Edit</b> 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.
Port Type	Choose a type. The choices are: <ul style="list-style-type: none"> <li>• <b>Access Port</b></li> <li>• <b>Trunk with Native VLAN</b></li> </ul>
Link Speed (optional)	Enter None, 10, 100, 1000, Auto, or nonegotiate.
Link Duplex (optional)	Enter None, Full, Half, or Auto.
PE/UNI Interface Description	Enter an optional description, for example <i>Customer-B ERMS (EVP-LAN) Service</i> .
VLAN ID AutoPick	Check the box if you want Prime Provisioning 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
VLAN NAME (optional)	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.


Table E-5 Interface Type Attributes (for VPLS) (continued)

Attribute	Description
System MTU	<p>Enter the size in bytes. The maximum transmission unit (MTU) size is configurable and optional. Prime Provisioning does not perform an integrity check for this customized value. If a service request goes to the <b>Failed Deploy</b> state because this size is not accepted, you must adjust the size until the service request is deployed. Prime Provisioning supports, ranges for different platforms, as specified below. The range is 1500 to 9216.</p> <ul style="list-style-type: none"> <li>For the 3750 and 3550 platforms, the MTU range is 1500-1546.</li> <li>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 Provisioning uses 9216 in both cases.</li> <li>For the 7600 SVI (interface VLAN), the MTU size is 1500-9216.</li> </ul>
Use Existing ACL Name	<p>Check the box if you want assign your own named access list to the port. By default, this check box is not checked and Prime Provisioning automatically assigns a MAC-based ACL on the customer facing UNI port, based on values you enter in <b>UNI MAC addresses</b> (below).</p>
Port-Based ACL Name	<p>Enter a Port-Based ACL Name (if you checked the <b>Use Existing ACL Name</b> check box, as mentioned in the previous step). Prime Provisioning 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.</p>
Disable CDP	<p>Check the box if you want to disable the Cisco Discover Protocol (CDP) on the UNI port.</p>
Filter BPDU	<p>Check the box to specify that the UNI port should not process Layer 2 Bridge Protocol Data Units (BPDUs).</p>
UNI Port Security	<p>Check the 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.</p> <ul style="list-style-type: none"> <li>For <b>Maximum Number of MAC address</b>, enter the number of MAC addresses allowed for port security.</li> <li>For <b>Aging</b>, enter the length of time the MAC address can stay on the port security table.</li> <li>For <b>Violation Action</b>, choose what action will occur when a port security violation is detected: <ul style="list-style-type: none"> <li><b>PROTECT</b>—Drops packets with unknown source addresses until a sufficient number of secure MAC addresses are removed to drop below the maximum value.</li> <li><b>RESTRICT</b>—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.</li> <li><b>SHUTDOWN</b>—Puts the interface into the error-disabled state immediately and sends an SNMP trap notification.</li> </ul> </li> <li>In the <b>Secure MAC Addresses</b> field, enter one or more Ethernet MAC addresses. Click the <b>Edit</b> button to enter the addresses.</li> </ul>
Enable Storm Control	<p>Check the 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.</p>

Table E-5 Interface Type Attributes (for VPLS) (continued)

Attribute	Description
Protocol Tunnelling	<p>Check the 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:</p> <ul style="list-style-type: none"> <li>• <b>Tunnel CDP</b>—Enable Layer 2 tunnelling on Cisco Discover Protocol (CDP).</li> <li>• <b>CDP Threshold</b>—Enter the number of packets per second to be received before the interface is shut down.</li> <li>• <b>cdp drop threshold</b>—Enter the number of packets per second to be received at which point the interface will start dropping CDP packets.</li> <li>• <b>Tunnel VTP</b>—Enable Layer 2 tunnelling on VLAN Trunk Protocol (VTP).</li> <li>• <b>VTP threshold</b>—Enter the number of packets per second to be received before the interface is shut down.</li> <li>• <b>vtp drop threshold</b>—Enter the number of packets per second to be received at which point the interface will start dropping VTP packets.</li> <li>• <b>Tunnel STP</b>—Enable Layer 2 tunnelling on Spanning Tree Protocol (STP).</li> <li>• <b>STP Threshold</b>—Enter the number of packets per second to be received before the interface is shut down.</li> <li>• <b>stp drop threshold</b>—Enter the number of packets per second to be received at which point the interface will start dropping STP packets.</li> <li>• <b>Recovery Interval</b>—Enter the amount of time, in seconds, to wait before recovering a UNI port.</li> </ul>

Table E-5 Interface Type Attributes (for VPLS) (continued)

Attribute	Description
Bridge Domain ID	<div data-bbox="435 317 483 359"></div> <div data-bbox="435 359 1487 430"> <p><b>Note</b> This attribute only appears in the Link Attributes window of some VPLS service request scenarios, as mentioned below.</p> </div> <hr/> <p>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:</p> <ul style="list-style-type: none"> <li>• The Bridge Domain ID attribute is only available for the following service request scenarios: <ul style="list-style-type: none"> <li>– Ethernet/ERMS with a CE</li> <li>– Ethernet/ERMS without a CE</li> <li>– Ethernet/EMS with a CE</li> <li>– Ethernet/EMS without a CE</li> </ul> </li> <li>• 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.</li> <li>• The following points apply to service requests based on this policy: <ul style="list-style-type: none"> <li>– 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.</li> <li>– 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.</li> <li>– 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.</li> </ul> </li> </ul>

# Sample Configlets

This section provides sample configlets for L2VPN and Metro Ethernet service provisioning in Prime Provisioning. It contains the following subsections:

- [Overview, page E-63](#)
- [ERS \(EVPL\) \(Point-to-Point\), page E-65](#)
- [ERS \(EVPL\) \(Point-to-Point, UNI Port Security\), page E-66](#)
- [ERS \(EVPL\) \(1:1 VLAN Translation\), page E-67](#)
- [ERS \(EVPL\) \(2:1 VLAN Translation\), page E-68](#)
- [ERS \(Pseudowire Class, E-Line, L2VPN Group Name, IOS XR Device\), page E-69](#)
- [ERS \(EVPL\) \(NBI Enhancements for L2VPN, IOS Device\), page E-70](#)
- [ERS \(EVPL\) and EWS \(EPL\) \(Local Connect on E-Line\), page E-71](#)
- [ERS \(EVPL\), EWS \(EPL\), ATM, or Frame Relay \(Additional Template Variables for L2VPN, IOS and IOS XR Device\), page E-72](#)
- [EWS \(EPL\) \(Point-to-Point\), page E-73](#)
- [EWS \(EPL\) \(Point-to-Point, UNI Port Security, BPDU Tunneling\), page E-74](#)
- [EWS \(EPL\) \(Hybrid\), page E-76](#)
- [EWS \(EPL\) \(Pseudowire Class, E-Line, L2VPN Group Name, IOS XR Device\), page E-79](#)
- [EWS \(EPL\) \(NBI Enhancements for L2VPN, IOS Device\), page E-80](#)
- [ATM over MPLS \(VC Mode\), page E-81](#)
- [ATM over MPLS \(VP Mode\), page E-82](#)
- [ATM \(Port Mode, Pseudowire Class, E-Line, L2VPN Group Name, IOS XR Device\), page E-83](#)
- [Frame Relay over MPLS, page E-84](#)
- [Frame Relay \(DLCI Mode\), page E-85](#)
- [VPLS \(Multipoint, ERMS/EVP-LAN\), page E-86](#)
- [VPLS \(Multipoint, EMS/EP-LAN\), BPDU Tunneling\), page E-87](#)

## Overview

The configlets provided in this section show the CLIs generated by Prime Provisioning 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 Provisioning 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
<pre>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 bpdupfilter 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</pre>	<pre>vlan 772 exit ! interface FastEthernet8/17 switchport trunk allowed vlan 1,451,653,659,766-768,772,878 ! <b>interface Vlan772</b> <b>no ip address</b> <b>description L2VPN ERS</b> <b>xconnect 99.99.8.99 89027 encapsulation</b> <b>mpls</b> <b>no shutdown</b></pre>

### 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	N-PE
<pre> 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 <b>switchport port-security</b> <b>switchport nonegotiate</b> <b>switchport port-security maximum 45</b> <b>switchport port-security aging time 34</b> <b>switchport port-security violation shutdown</b> <b>switchport port-security mac-address</b> <b>3456.3456.5678</b> spanning-tree bpdupfilter 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 <b>deny any host 1234.3234.3432</b> permit any any           </pre>	<pre> 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           </pre>

### 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	N-PE
<pre> ! vlan 123 exit ! interface FastEthernet1/0/8 no cdp enable no keepalive no ip address <b>switchport trunk allowed vlan 123</b> switchport nonegotiate switchport port-security maximum 34 switchport port-security aging time 23 switchport port-security violation protect switchport port-security spanning-tree bpdudfilter enable mac access-group ISC-FastEthernet1/0/8 in ! interface GigabitEthernet1/1/1 no ip address switchport mode trunk <b>switchport trunk allowed vlan 1,123</b> <b>switchport vlan mapping 123 778</b>           </pre>	<pre> 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           </pre>

### 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
<pre> vlan 567 exit ! interface FastEthernet1/0/5 no cdp enable no keepalive no ip address switchport <b>switchport access vlan 567</b> <b>switchport mode dot1q-tunnel</b> switchport trunk allowed vlan none switchport nonegotiate spanning-tree bpdupfilter enable mac access-group ISC-FastEthernet1/0/5 in ! interface GigabitEthernet1/1/1 no ip address switchport trunk allowed vlan 1,123,567 <b>switchport vlan mapping dot1q-tunnel 567</b> <b>234 779</b> ! 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 </pre>	<pre> 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 </pre>

### 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	N-PE
<pre> ! 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 bpdupfilter 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 ! </pre>	<pre> ! 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 ! ! </pre>

### 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	N-PE
<pre> ! 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 <b>storm-control unicast level 1.0</b> <b>storm-control broadcast level 50.0</b> <b>storm-control multicast level 50.0</b> shutdown <b>keepalive</b> spanning-tree bpdudfilter enable  ! interface GigabitEthernet1/0/1 no ip address switchport switchport trunk encapsulation dot1q switchport mode trunk switchport trunk allowed vlan 3200 ! </pre>	<pre> ! 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 <b>storm-control unicast level 1.0</b> <b>storm-control broadcast level 50.0</b> <b>storm-control multicast level 50.0</b> shutdown keepalive spanning-tree bpdudfilter enable  ! <b>interface Vlan3300</b> no ip address xconnect 192.169.105.40 7502 encapsulation mpls no shutdown ! </pre>

### Comments

None.

## 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
	<pre>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 <b>c1-test-12-crs1-1--0--559</b>     interface GigabitEthernet0/0/0/2.559     interface GigabitEthernet0/0/0/4.559   !   !   !</pre>

### 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)	<p>Template Content:</p> <pre>interface Loopback0 description LocalLoopbackAddress=\$L2VPNLocalLoopback LocalHostName=\$L2VPNLocalHostName RemoteLoopbackAddress=\$L2VPNRemoteLoopback RemoteHostName=\$L2VPNRemoteHostName</pre> <p>Configlets:</p> <pre>interface Loopback0 description LocalLoopbackAddress= 192.169.105.40 LocalHostName=c1-test-12-7600-2 RemoteLoopbackAddress=192.169.105.80 RemoteHostName= c1-test-12-7600-4</pre>

### 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	N-PE
<pre> 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 bpdupfilter enable ! interface FastEthernet1/0/23 no ip address switchport trunk allowed vlan 774,787-788           </pre>	<pre> 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           </pre>

### 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
<pre> 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 <b>switchport port-security maximum 34</b> <b>switchport port-security aging time 32</b> <b>switchport port-security violation shutdown</b> <b>switchport port-security</b> <b>l2protocol-tunnel cdp</b> <b>l2protocol-tunnel stp</b> <b>l2protocol-tunnel vtp</b> <b>l2protocol-tunnel shutdown-threshold cdp 88</b> <b>l2protocol-tunnel shutdown-threshold stp 99</b> <b>l2protocol-tunnel shutdown-threshold vtp 56</b> <b>l2protocol-tunnel drop-threshold cdp 56</b> <b>l2protocol-tunnel drop-threshold stp 64</b> <b>l2protocol-tunnel drop-threshold vtp 34</b> <b>storm-control unicast level 34.0</b> <b>storm-control broadcast level 23.0</b> <b>storm-control multicast level 12.0</b> spanning-tree portfast spanning-tree bpdupfilter 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 <b>deny any host 3456.3456.1234</b> permit any any </pre>	<pre> 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 </pre>

## 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
<pre> 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 <b>switchport port-security maximum 34</b> <b>switchport port-security aging time 32</b> <b>switchport port-security violation shutdown</b> <b>switchport port-security</b> <b>l2protocol-tunnel cdp</b> <b>l2protocol-tunnel stp</b> <b>l2protocol-tunnel vtp</b> <b>l2protocol-tunnel shutdown-threshold cdp 88</b> <b>l2protocol-tunnel shutdown-threshold stp 99</b> <b>l2protocol-tunnel shutdown-threshold vtp 56</b> <b>l2protocol-tunnel drop-threshold cdp 56</b> <b>l2protocol-tunnel drop-threshold stp 64</b> <b>l2protocol-tunnel drop-threshold vtp 34</b> <b>storm-control unicast level 34.0</b> <b>storm-control broadcast level 23.0</b> <b>storm-control multicast level 12.0</b> spanning-tree portfast spanning-tree bpdupfilter 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 <b>deny any host 3456.3456.1234</b> permit any any </pre>	<pre> 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 </pre>

## 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
<pre> 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 </pre>	<pre> 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 </pre>

## 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	N-PE
<pre> ! 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 bpdupfilter enable ! interface GigabitEthernet1/0/1 no ip address switchport switchport trunk encapsulation dot1q switchport trunk allowed vlan 700 switchport mode trunk ! </pre>	<pre> ! 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-12-12404-2--1000 interface GigabitEthernet0/3/1/1.700 neighbor 192.169.105.30 pw-id 1000 pw-class PW_AD7-AD3_Cutsomer2 ! </pre>

### 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 Provisioning-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	N-PE
<pre> ! 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 <b>storm-control unicast level 1.0</b> <b>storm-control broadcast level 50.0</b> <b>storm-control multicast level 50.0</b> shutdown <b>keepalive</b> spanning-tree bpdufilter enable  ! interface GigabitEthernet1/0/1 no ip address switchport switchport trunk encapsulation dot1q switchport mode trunk switchport trunk allowed vlan 3201 ! </pre>	<pre> ! 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 <b>storm-control unicast level 1.0</b> <b>storm-control broadcast level 50.0</b> <b>storm-control multicast level 50.0</b> shutdown <b>keepalive</b> spanning-tree bpdufilter enable  ! <b>interface Vlan3301</b> no ip address xconnect 192.169.105.40 7502 encapsulation mpls no shutdown ! </pre>

### 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)	<pre>interface ATM2/0.34234 point-to-point pvc 213/423 l2transport encapsulation aal5 xconnect 99.99.4.99 89025 encapsulation mpls</pre>

### 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)	<pre>pseudowire-class ISC-pw-tunnel-123 encapsulation mpls preferred-path interface tunnel123 disable-fallback ! interface ATM2/0 atm pvp 131 12transport xconnect 99.99.4.99 89024 pw-class ISC-pw-tunnel-123</pre>

### 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)	<pre>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 !</pre>

### 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)	<pre>interface Serial1/1 exit ! connect C1_89001 Serial1/1 135 12transport xconnect 99.99.4.99 89001 encapsulation mpls</pre>

### 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)	<pre>pseudowire-class ISC-pw-dynamic-default encapsulation l2tpv3 ip local interface Loopback10 ip dfbit set ! interface Serial13/2 encapsulation frame-relay exit ! connect ISC_1054 Serial13/2 86 l2transport xconnect 10.9.1.1 1054 encapsulation l2tpv3 pw-class ISC-pw-dynamic-default</pre>

### 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	N-PE
<pre>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 bpdupfilter 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</pre>	<pre>12 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</pre>

### 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 BPDU 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
<pre> 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 bpdupfilter enable           </pre>	<pre> 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           </pre>

### Comments

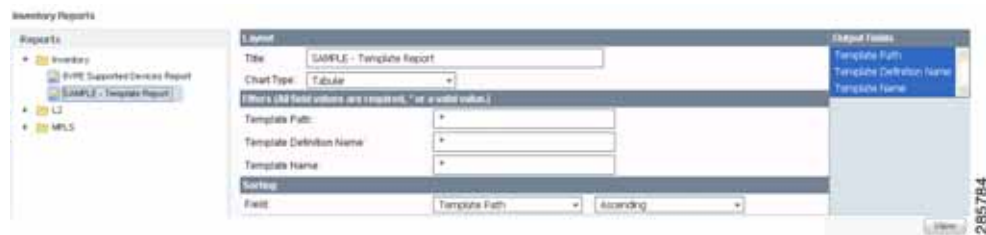
- The N-PE is a 7600 with an OSM or SIP-600 module.
- The VFI contains all the N-PEs (neighbors) that this N-PE talks to.
- The VPLS EMS (EP-LAN) UNI is the same as L2VPN (point-to-point) EWS (EPL) UNI.
- The SVI is the same as VPLS ERS (EVP-LAN) SVI.

# 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 elsewhere in this guide.

Click on any of the specific reports and you can define how to set up the report. [Figure E-1](#) shows the sample file under the folder **Inventory**.

**Figure E-1** *Inventory > SAMPLE - Template Report - Report Window*



This section explains the Reports feature and how to use it in the following areas:

- [Introducing Reports, page E-88](#)
- [Accessing Reports, page E-89](#)
- [Using Reports GUI, page E-89](#)
- [Running Reports, page E-90](#)
- [Creating Custom Reports, page E-92](#)

## 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 Provisioning 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 this guide, Prime Provisioning provides additional sample reports. Sample reports are provided for informational purposes only and are untested and unsupported.

The data structures that Prime Provisioning 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 Provisioning 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 E-1](#).
- 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 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 E-92](#) 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 E-1](#).

The window is divided into several areas:

- [Layout, page E-89](#)
- [Filters, page E-89](#)
- [Output Fields, page E-90](#)
- [Sorting, page E-90](#)

### 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 Provisioning 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 Provisioning. 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 E-2](#)).

## 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 E-2](#).

**Figure E-2** Report Output

#	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	Act2000
18	Examples	AccessList1	Protocol-IP
19	Examples	AccessList1	Protocol-TCP
20	Examples	ATM	ATMData

The reports GUI supports output in tabular format. The output is listed in columns, which are derived from 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 E-91
- Print explained in the “[Printing Reports](#)” section on page E-91
- E-mail explained in the “[E-mailing Reports](#)” section on page E-91

## Exporting Reports

Click on the **Export** icon in [Figure E-2](#) 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 Provisioning 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 E-2](#).

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 E-2](#) 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 Provisioning 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:

**\$PRIMEP\_HOME/resources/nbi/reports/PrimeProvisioning/<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 Provisioning server using the **./prime.sh stopall** command.  
See [Cisco Prime Provisioning 7.2 Administration Guide](#) for information on starting and stopping Prime Provisioning.
- Step 2** Open the **\$PRIMEP\_HOME/resources/nbi/reports/PrimeProvisioning/<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 **\$PRIMEP\_HOME/resources/nbi/reports/PrimeProvisioning/<folder\_name>\_report.xml** file.
- Step 5** Restart the Prime Provisioning server using the **./prime.sh startwd** command.  
See [Cisco Prime Provisioning 7.2 Administration Guide](#) for information on starting and stopping Prime Provisioning.
- 

After restarting Prime Provisioning, the modifications take effect, based on changes you made to the **\$PRIMEP\_HOME/resources/nbi/reports/PrimeProvisioning/<folder\_name>\_report.xml** file.

## Generating L2 and VPLS Reports

The Prime Provisioning reporting GUI is used across multiple Prime Provisioning 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 E-88](#). The rest of this section provides information about the L2 and VPLS reports available in Prime Provisioning.

This section provides information on generating L2 and VPLS reports. It contains the following sections:

- [Accessing L2 and VPLS Reports, page E-93](#)
- [L2 and VPLS Reports, page E-93](#)
- [Creating Custom L2 and VPLS Reports, page E-100](#)

## 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 Provisioning 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 E-93](#).

## L2 and VPLS Reports

This section provides details on the following L2 and VPLS reports:

- [L2 End-to-End Wire Report, page E-94](#)
- [L2 PE Service Report, page E-96](#)
- [L2 VPN Report, page E-97](#)
- [VPLS Attachment Circuit Report, page E-97](#)
- [VPLS PE Service Report, page E-99](#)
- [VPLS VPN Report, page E-100](#)



### 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 E-100](#).

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 Provisioning 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 E-88](#) for details on how to modify report configuration files to create custom reports.

## Generating MPLS Reports

The Prime Provisioning reporting GUI is used across multiple Prime Provisioning 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:

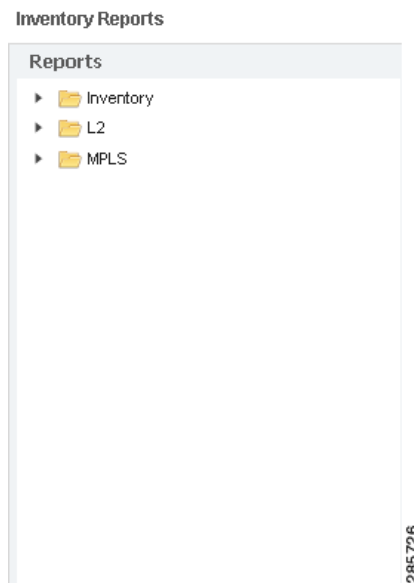
- [Accessing Reports, page E-89](#)
- [Running Reports, page E-90](#)
- [MPLS PE Service Report, page E-102](#)
- [MPLS Service Request Report, page E-102](#)
- [MPLS Service Request Report - 6VPE, page E-103](#)
- [6VPE Supported Devices Report, page E-104](#)
- [Creating Custom Reports, page E-92](#)

## Accessing MPLS Reports

To access MPLS reports, perform the following steps:

- 
- Step 1** Log into Prime Provisioning.
  - Step 2** Go to: **Inventory > Reports > Inventory Reports**.
  - Step 3** Click on the MPLS folder to display the available MPLS reports.  
The Reports window appears, as shown in [Figure E-3](#).

**Figure E-3** 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 E-105](#), 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 from 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 ascending and descending 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 E-4](#).

**Figure E-4** 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 10-14](#)).



**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 E-5](#).

Figure E-5 MPLS Service Request Report

Layout	
Title:	MPLS SR Report (PE,CE,VPN,SR ID,SR STATE)
Chart Type:	Tabular
Filters (All field values are required, * or a valid value.)	
PE_ROUTER:	* <input type="text"/> <input type="button" value="Select"/>
CE_ROUTER:	* <input type="text"/> <input type="button" value="Select"/>
Job_ID:	* <input type="text"/>
SR_STATE:	* <input type="text"/>
VPN_ID:	* <input type="text"/> <input type="button" value="Select"/>
Output Fields	
<div style="border: 1px solid black; padding: 5px;">           PE_ROUTER            CE_ROUTER            Job_ID            SR_STATE            VPN_ID            CREATION_DATE_TIME         </div>	
Sorting	
N/A	

#### 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 10-14](#)).
- **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 10-14](#)).



**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 E-6](#).

Figure E-6 MPLS Service Request Report - 6VPE

### Filter Values

- **Job ID**—Service request job IDs.
- **SR STATE**—Service request states (see [Service Request States, page 10-14](#)).
- **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 10-14](#)).



**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 Provisioning GUI, this report is located under **Inventory > Reports > Inventory Reports**.

Click the 6VPE Supported Devices Report icon to bring up the window for this report, as shown in [Figure E-7](#).



Figure E-7 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 Provisioning 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 section includes:

- [TE Task Logs, page E-105](#)
  - [SR Deployment Logs, page E-106](#)
  - [Logs Created from Task Manager, page E-106](#)
  - [Viewing a Task Log, page E-106](#)
- [TE Performance Reports, page E-107.](#)

## 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 9-17](#).

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 9-74](#).

## 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 9-76](#).

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 **Inventory > 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.
- 

## EMAIL

Email feature has been deprecated and will be removed in a subsequent release.



## Removed Features: Cisco Configuration Engine Server

---

This appendix describes about the support for Cisco Networking Services (CNS) Layer 2 services that has been removed and are no longer accessible from the product. See your Cisco representative for further details.



Note

The Cisco Configuration Engine Server is referred to as IE2100 throughout the Prime Provisioning user interface. The IE2100 appliance referenced within Prime Provisioning 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.

Prime Provisioning 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 Provisioning also supports CNS Plug-and-Play.

To use the Cisco CNS IE2100 functionality on Prime Provisioning, you must first set up the Cisco CNS IE2100 appliance and the Prime Provisioning workstation as explained in an appendix in the [Cisco Prime Provisioning 7.2 Installation Guide](#).

This appendix includes the following sections. Implement these sections in sequence:



Note

The [“Using Plug-and-Play” section on page F-4](#) is optional.

1. [Creating a Cisco CNS IE2100 Appliance, page F-1](#)
2. [Creating a Cisco IOS Device Using the Cisco CNS Device Access Protocol, page F-2](#)
3. [Using Plug-and-Play, page F-4](#)

## Creating a Cisco CNS IE2100 Appliance

Prime Provisioning 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 Provisioning.”](#)

- 
- 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 Provisioning.”](#)

- 
- 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 Provisioning.
- 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 Provisioning sends e-mail notifications upon receipt of device configuration changes originated outside Prime Provisioning 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 *Cisco Prime Provisioning 7.2 Administration Guide*. 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**. Email feature has been deprecated and will be removed in a subsequent release



**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 Provisioning.

**Step 7** Specify the **Device State**, as follows:

- Choose **ACTIVE** (the default)—if 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 Provisioning 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 Provisioning supports the Plug-and-Play device configuration through a Cisco CNS IE2100 appliance. Prime Provisioning 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 F-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 Provisioning 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 Provisioning 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 Provisioning 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 Provisioning 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 Provisioning 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 Provisioning performs a collect config of the IOS device and stores it in the Prime Provisioning repository.
- Verifies whether any Prime Provisioning 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.





## 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 [Cisco Prime Provisioning 7.2 Administration Guide](#).

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 G-1** DCPL Properties

Property	Default Value	Range/Rules	Explanation
<b>AutoDiscovery Properties:</b>			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 <b>true</b> and <b>false</b> .	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 <b>true</b> and <b>false</b> .	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.
<b>Cleanup Properties:</b>			Cleans up various system resources such as log files and temporary files.
/Cleanup/TaskLogs/			This component cleans up old TaskLogs.

Table G-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.
<b>/Cleanup/Tasks/</b>			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.
<b>/Cleanup/TempFiles/</b>			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.
<b>/Cleanup/logLevel</b>	<b>CONFIG</b>	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).
<b>CNS Properties:</b>			
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 G-1 DCPL Properties (continued)

deprecatedReboot	0	The valid values are <b>0</b> and <b>1</b> .	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.
<b>DCS Properties:</b>			Device Configuration Service. This component corresponds to a library that is used by Prime Provisioning to communicate with network devices using protocols such as telnet, ssh, tftp, and so forth.
<b>/DCS/FTP/</b>			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 <b>true</b> and <b>false</b> .	If <b>true</b> , DCS uses only the primary warning expression list, specified in DCS/IOSWarningExpressions. If <b>false</b> , 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 G-1 DCPL Properties (continued)

/DCS/IOSWarningExpressions		string	<p>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.</p> <p>All regular expressions except the last one should have a \$ at the end of the regular expression.</p> <p>% 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\$</p>
----------------------------	--	--------	----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

Table G-1 DCPL Properties (continued)

/DCS/IOSWarningExpressions (Continued)		string	<p>% IP routing table V.* does not exist. Create first\$</p> <p>% IP routing table g.*does not exist. Create first\$</p> <p>% No CEF interface information\$</p> <p>%No matching route to delete\$</p> <p>% Translation not found\$</p> <p>.*Not all config may be removed and may reappear after reactivating\$</p> <p>^%.?NOTE:\$</p> <p>OSPF: Unrecognized virtual interface .* Treat it as loopback stub route\$</p> <p>outside interface address added\$</p> <p>% Profile already contains this keyring\$</p> <p>%PVC is already defined\$</p> <p>Restarting RADIUS authentication service on port .*</p> <p>\$ Restarting RADIUS accounting service on port .*\$</p> <p>Redundant .* statement\$</p> <p>security level for .* changed to\$</p> <p>.*Service policy .* is already attached\$</p> <p>% Signature RSA Keys not found in configuration.\$</p> <p>.*success\$</p> <p>The .*command will also show the fingerprint\$ %The static routes in .* with outgoing interface .* will be removed\$</p> <p>Unable to disable parser cache\$</p> <p>% Unknown VPN\$ .*</p> <p>Unknown VRF specified\$</p> <p>% VRF .* does not exist or does not have a RD\$</p> <p>.?warning.*</p>
/DCS/IOSWarningExpressionsExitCfgMode		string	<p>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.</p>

Table G-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.
<b>/DCS/RCP/</b>			
rcpDirectory	/tmp	string	Directory to use for uploaded/downloaded config files.
<b>/DCS/SSH/</b>			
overWriteSSHKeys	true	The valid values are <b>true</b> and <b>false</b> .	Overwrite SSH Keys: If <b>true</b> , will allow new keys to overwrite existing keys in the key file for a given host. If <b>false</b> , 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.
<b>/DCS/SSHv2/</b>			
overWriteSSHv2Keys	true	The valid values are <b>true</b> and <b>false</b> .	Overwrite SSHv2 Keys: If <b>true</b> , will allow new keys to overwrite existing keys in the key file for a given host. If <b>false</b> , an error will be displayed if host sent key does not match the server sent key.
<b>/DCS/TFTP/</b>			
tftpCreateFileOnServerBeforeUpload	true	The valid values are <b>true</b> and <b>false</b> .	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 <b>true</b> , 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 Provisioning server.
tftpSubDirectory		string	TFTP Sub Directory used by DCS and GTL.
<b>/DCS/XR</b>			
IOS XR properties.			

Table G-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 <b>true</b> and <b>false</b> .	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 <b>true</b> and <b>false</b> .	Retrieve the committed CLI configuration after an XML configuration download. If the default of <b>true</b> 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 <b>false</b> , 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 G-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.
<b>/DCS/misc</b>			Miscellaneous settings.
ConfigForMergeXML		string, file name	Configuration file to be used for the merging of two XMLs.
allowPromptCharsInBanner	false	The valid values are <b>true</b> and <b>false</b> .	Controls if prompt characters, such as # and >, are allowed in banners. If <b>true</b> , 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 <b>true</b> . 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.
<b>DeploymentFlow Property:</b>			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 G-1 DCPL Properties (continued)

Discovery Properties:			Prime Provisioning auto discovery framework.
/Discovery/DeviceDiscovery			
continueOnError	false	The valid values are <b>true</b> and <b>false</b> .	A Boolean flag indicating whether device discovery should try to continue on an error. When the value is <b>true</b> , 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. <b>WARNING:</b> If this property is set to <b>true</b> , 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 G-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.
<b>/Discovery/DataCollection</b>			
continueOnError	false	The valid values are <b>true</b> and <b>false</b> .	A Boolean flag indicating whether data collection should try to continue on an error. When the value is <b>true</b> , 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 <b>true</b> , 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 G-1 DCPL Properties (continued)

reuseConfigsIfAvailable	false	The valid values are <b>true</b> and <b>false</b> .	If the Boolean flag is <b>true</b> , 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.
<b>/Discovery/MPLSService</b>			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).
<b>/Discovery/MetroEService</b>			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 <b>true</b> and <b>false</b> .	Set this to <b>true</b> 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 <b>false</b> for performance reasons.
<b>/Discovery/NPCDiscovery</b>			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).
<b>/Discovery/RoleAssignment</b>			

Table G-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).
<b>/Discovery/Workflow</b>			Prime Provisioning 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 <b>true and false.</b>	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 <b>true.</b>
/Discovery/tmpdir	<vpnsc_tmp> /Discovery	string	A directory to store the temporary results of the discovery process.
<b>DistributionFramework Properties:</b>			Distribution Framework. This component handles the distribution of work (jobs) between different servers in a Prime Provisioning distributed installation.
<b>/DistributionFramework/Dispatcher/</b>			Service that dispatches jobs to workers.
DefaultUnitDuration	1000	integer	The unit duration (in milliseconds) used to estimate jobs without a profile.

Table G-1 DCPL Properties (continued)

PingInterval	1000	integer	The interval (in ms) dispatcher pings the workers to get the load.
ProcessorEpsilon	10	integer	If two processors 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.
<b>/DistributionFramework/RemoteUtil/</b>			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).
<b>/DistributionFramework/ServiceLauncher/</b>			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).
<b>/DistributionFramework/ThreadPool/</b>			Thread pool component used by the worker to execute jobs.

Table G-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).
<b>/DistributionFramework/Worker/</b>			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).
<b>GSAM Property:</b>			Generic Service Access Model to get an XML dump from the repository for the provisioning driver.
<b>/GSAM/logLevel</b>	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).
<b>GTL Properties:</b>			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.
<b>/GTL/CSL/</b>			Configuration Services Layer
ios/			IOS related properties.

Table G-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 G-1 DCPL Properties (continued)

usePAM	false	The valid values are <b>true</b> and <b>false</b> .	When the value is <b>true</b> , the selected PAM is used for device authentication. When the value is <b>false</b> , the standard authentication credentials are used in the Prime Provisioning 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 <b>true</b> and <b>false</b> .	Flag indicating whether to run GTL in <b>ECHO</b> mode or <b>DCS</b> mode. Setting Prime Provisioning to run in echo mode allows Prime Provisioning 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 <b>true</b> ), 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 Provisioning downloads the configurations.
/GTL/ios/			IOS related GTL properties.
copy-running-to-startup	true	The valid values are <b>true</b> and <b>false</b> .	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).
<b>GUI Properties:</b>			The component for GUI-based properties.



Table G-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 G-1 DCPL Properties (continued)

<b>/GUI/MPLSOAM/</b>			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).
<b>/GUI/MplsVPN/</b>			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 <b>true</b> and <b>false</b> .	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.
<b>/GUI/Performance/</b>			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).
<b>GUI/Ping</b>			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).
<b>/GUI/Topology/</b>			Component related to the web start topology application.

Table G-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 <b>true</b> and <b>false</b> .	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.
<b>JavaWebStart Properties:</b>			Java Web Start components.
<b>/JavaWebStart/InventoryManager/</b>			Component to create and manage Devices.
MaxDevicesPerSaveTransaction	25	integer, 1-500	Specifies the maximum number of devices per transaction when performing save operation.
<b>/JavaWebStart/TaskManager/</b>			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 one Collect Config task created and executed in the repository.

Table G-1 DCPL Properties (continued)

<b>LDAP Properties</b>			LDAP properties is used to create the users, set password, and roles for the user in the LDAP server.
<b>/LDAP/LdapAuthentication/</b>	false	The valid values are <b>true</b> and <b>false</b> .	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.
<b>/LDAP/HostName/</b>		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.
<b>/LDAP/DistinguishedName/</b>		string	This property is used for Authentication of user Credentials. The Distinguished Name is same for multiple LDAP servers.
<b>/LDAP/UserDefinedException/</b>		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.
<b>Logging Properties:</b>			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.
<b>/Logging/Defaults/</b>			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 <b>logFileSize</b> . 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 <b>logFileNumber</b> 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 G-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.
<b>/Logging/TaskLogs/</b>			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.
<b>Provisioning Properties:</b>			Contains properties and components for service provisioning like MPLS VPNs.
<b>/Provisioning/Engine/</b>			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.
<b>/Provisioning/NOM/</b>			Network Object Model for parsing and delta generation of configs.
<b>DocumentBuilderFactory/</b>			This contains the properties for the DOM builder factory.
ignoreComments	true	The valid values are <b>true</b> and <b>false</b> .	Flag.
ignoreWhiteSpace	false	The valid values are <b>true</b> and <b>false</b> .	Flag for DOM builder factory.
validation	false	The valid values are <b>true</b> and <b>false</b> .	Flag for validation of xml files.

Table G-1 DCPL Properties (continued)

catSyntaxFile	catSyntax.xml	string	Contains the XML for Catalyst command syntax.
explicitlyRemoveRouteTargets	false	The valid values are <b>true</b> and <b>false</b> .	Normally ( <b>false</b> ), 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 <b>true</b> , Prime Provisioning 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).
<b>/Provisioning/PasswordManagement/</b>			User generated Password generation
<b>PasswordFormula/</b>			User generated Password formula generation class
class		string	User generated class file
<b>/Provisioning/ProvDrv/</b>			Contains properties for the XML driven provisioning ProvDrv.
AuditJITUpload	true	The valid values are <b>true</b> and <b>false</b> .	If the value of this property is set to <b>false</b> , 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 <b>true</b> , 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 <b>true</b> .
CleanStagedConfigletWhenForceDeploy	false	The valid values are <b>true</b> and <b>false</b> .	If this value is <b>true</b> , when a service request is force deployed, the staged configlet is removed before provisioning. If this value is the default of <b>false</b> , the staged configlet is considered as part of the base configuration during provisioning.

Table G-1 DCPL Properties (continued)

DownloadTemplateToUnmanagedDevice	false	The valid values are <b>true</b> and <b>false</b> .	If this value is <b>true</b> , for an unmanaged device, Prime Provisioning attempts to download just the template. The configlet generated by the provision is not part of the download. By default, this value is <b>false</b> and then there is no attempt to download to an unmanaged device.
ForceTemplateDeploy	false	The valid values are <b>true</b> and <b>false</b> .	<p>Templates are downloaded in the first/initial service request (SR) deployment.</p> <p>During <b>edit/modification of SRs</b>, in case templates are attached to the SR, the following is true:</p> <p>a) Templates are updated if this property is set to <b>true</b>, in case the physical interface or VLAN ID (or other Prime Provisioning repository variables) is modified or changed.</p> <p>b) Templates attached to a service policy are not downloaded to new LINKS added to an existing SR if this property is set to <b>true</b> or <b>false</b>.</p> <p>Templates are not updated if this property is set to <b>false</b>, if the physical interface or VLAN ID (or other Prime Provisioning repository variables) is modified.</p> <p>If the templates do not include any Prime Provisioning repository variables, the recommendation is to set this property to <b>false</b>. If the templates include any Prime Provisioning repository variables, the recommendation is to set this property to <b>true</b>.</p>
MaxNumberOfDevicesPerDownload	100	integer	Prime Provisioning 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 G-1 DCPL Properties (continued)

NegateTemplateDeploy	true	The valid values are <b>true</b> and <b>false</b> .	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 <b>true</b> and <b>false</b> .	If the value of this property is set to <b>false</b> , 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 <b>true</b> , 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 <b>true</b> and <b>false</b> .	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 G-1 DCPL Properties (continued)

enableLogging	true	The valid values are <b>true</b> and <b>false</b> .	When the value is the default of <b>true</b> , debugging of logging is enabled for this service.  When the value is <b>false</b> , debugging of logging is not enabled for this service.
<b>platform/</b>			Used by ProvDrv
<b>CISCO_ROUTER/</b>			Used by ProvDrv
serviceBladeClass	com.cisco.vpnsc. prov.te. ServiceBlade. TeServiceBlade	string	Identifies ServiceBlade class name for ProvDrv.
sendAuditEvent	true	The valid values are <b>true</b> and <b>false</b> .	Set <b>true</b> to enable sending audit event for this service.
<b>Uds/</b>			User defined services.
<b>platform/</b>			Service platform
<b>CISCO_ROUTER/</b>			Cisco router
serviceBladeClass	com.cisco.vpnsc. prov.uds.Uds ServiceBlade	string	Uds Service Blade.
<b>deviceConfig/</b>			
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 <b>true</b> and <b>false</b> .	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 <b>true</b> and <b>false</b> .	If this property is set to <b>true</b> , 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 <b>true</b> and <b>false</b> .	Set true to enable sending audit event for this service.
serviceFile	l2vpnService.xml	string	Layer 2 VPN Service definition file.
<b>platform/</b>			Contains properties for L2VPN for different platforms.

Table G-1 DCPL Properties (continued)

<b>CATOS/</b>			Service blade parameters for CATOS.
serviceBladeClass	com.cisco.vpnsc. prov.l2vpn.L2VP NServiceBlade	string	ServiceBladeClass location.
<b>CISCO_ROUTER/</b>			
iosXRConfigType	XML		Config type for IOS XR devices for MPLS service blade
serviceBladeClass	com.cisco.vpnsc. prov.l2vpn.L2VP NServiceBlade	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).
<b>fsm/</b>			MPLS Finite State Machine (FSM) Provisioning.
<b>DownloadWeights/</b>			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.
<b>platform/</b>			Contains properties for L2VPN for different platforms.
<b>CATOS/</b>			Service blade parameters for CATOS.
serviceBladeClass	com.cisco.vpnsc. prov.fsm. FSMService Blade	string	ServiceBladeClass location.
<b>CISCO_ROUTER/</b>			
IosXRConfigType	XML		Config type for IOS XR devices for MPLS service blade
serviceBladeClass	com.cisco.vpnsc. prov.fsm. FSMService Blade	string	ServiceBladeClass location.

Table G-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 <b>true</b> and <b>false</b> .	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 <b>true</b> and <b>false</b> .	If this property is set to <b>true</b> , 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 <b>true</b> and <b>false</b> .	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.
<b>l2vpn/</b>			MPLS Layer 2 VPN Provisioning.
<b>DownloadWeights/</b>			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.
<b>platform/</b>			Contains properties for L2VPN for different platforms.
<b>CATOS/</b>			Service blade parameters for CATOS.
serviceBladeClass	com.cisco.vpnsc. prov.l2vpn.L2VPNServiceBlade	string	ServiceBladeClass location.
<b>CISCO_ROUTER/</b>			

Table G-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 <b>true</b> and <b>false</b> .	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 <b>true</b> and <b>false</b> .	If this property is set to <b>true</b> , 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 <b>true</b> and <b>false</b> .	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).
<b>mpls/</b>			Contains properties for MPLS/BGP Layer 3 VPN service.
<b>DownloadWeights/</b>			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 G-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.
<b>platform/</b>			Platform related classes.
<b>CATOS/</b>			Service blade parameters for CATOS.
serviceBladeClass	com.cisco.vpnsc. prov.mpls.MplsS erviceBlade	string	ServiceBladeClass location.
<b>CISCO_ROUTER/</b>			IOS.
iosXRConfigType	XML		Config type for IOS XR devices for MPLS service blade
serviceBladeClass	com.cisco.vpnsc. prov.mpls.MplsS erviceBlade	string	ServiceBladeClass location
allowDuplicateIpAddressForPPPo ATM	false	The valid values are <b>true</b> and <b>false</b> .	Provision PPPoATM by allowing duplicate IP addresses for MPLS Service Requests. Ignore duplicate IP address on Loopback and Multilink interfaces.
allowOverwriteManualAssigned Address	false	The valid values are <b>true</b> and <b>false</b> .	Allow manually-assigned IP address in Service Request overwrite the pre-existing interface IP address. <b>False</b> 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 Provisioning detects that and reports the error. <b>True</b> means Prime Provisioning allows the new IP address to overwrite the existing IP address.
allowShared VLAN Modification	false	The valid values are <b>true</b> and <b>false</b> .	For residential services, if the flag is on, <b>true</b> , shared VLAN attributes are available for modify in edit mode. If the flag is off, <b>false</b> , attributes are in read only mode.
auditIpAddressViaUnnumbered	false	The valid values are <b>true</b> and <b>false</b> .	When the value is the default of <b>false</b> , the auditor only looks for the IP address of a provisioned interface. When the value is <b>true</b> , the auditor tries to match the IP address of the unnumbered interface, if one exists.

Table G-1 DCPL Properties (continued)

auditMaxrouteThreshold	true	The valid values are <b>true</b> and <b>false</b> .	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 <b>true</b> and <b>false</b> .	This property is set for the autodiscovered systems containing a superset of the commands that Prime Provisioning 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 <b>true</b> and <b>false</b> .	Exclude the no keepalive command on the port channel trunk port.
forceRemoveNonBroadcastStatic RouteOnPE	false	The valid values are <b>true</b> and <b>false</b> .	The default value is <b>false</b> . When the value is set to <b>true</b> , Prime Provisioning removes the non-broadcast type static route command that has a pre-existing long syntax, even if the command was not provisioned by Prime Provisioning. 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 <b>true</b> and <b>false</b> .	Remove a VRF, even when some Loopback interfaces are still pointing to it.
ignoreMajorInterfaceCheck	false	The valid values are <b>true</b> and <b>false</b> .	This property controls the check for a proper major interface name in an unmanaged CE. If set to <b>true</b> , Prime Provisioning 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 <b>true</b> and <b>false</b> .	If set to <b>true</b> , 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 G-1 DCPL Properties (continued)

ospfProcessLimit	2	integer	If the number of OSPF processes reaches or exceeds this limit, a warning will be produced.
parseConfigAfterProvisioning	false	The valid values are <b>true</b> and <b>false</b> .	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 <b>true</b> and <b>false</b> .	When this property is set to <b>true</b> , the Prime Provisioning 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 <b>true</b> and <b>false</b> .	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 <b>true</b> and <b>false</b> .	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 <b>true</b> and <b>false</b> .	Removing the Prime Provisioning generated subinterface commands in decommission service requests.
routeMapDeletedAfterLastLinkDeletion	true	The valid values are <b>true</b> and <b>false</b> .	If this property is set to <b>true</b> , the route map configuration is automatically removed from the device after the last link is deleted. If <b>false</b> , the route map configuration is left as it is in the device.
saveDebugData	true	The valid values are <b>true</b> and <b>false</b> .	If this property is set to <b>true</b> , 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 <b>true</b> and <b>false</b> .	Set true to enable sending audit event for this service.

Table G-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
skipIpAddressValidationOnUnmanagedCE	false	The valid values are <b>true</b> and <b>false</b> .	When the value is <b>false</b> , 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 <b>true</b> , this validation is bypassed.
useNextHopAddressForStaticRoutes	false	The valid values are <b>true</b> and <b>false</b> .	For Static Routes, use local router outbound interface or IP address of the next hop to reach the destination network.
useOnlyExtraCEloopbackForGreyAccessList	false	The valid values are <b>true</b> and <b>false</b> .	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.
<b>shared/</b>			Properties shared by MPLS VPN, L2VPN and VPLS.
<b>FeatureQuery/</b>			Prime Provisioning 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 <b>true</b> and <b>false</b> .	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 <b>allowed vlan</b> cmd is absent for ERS service.
leaveSystemMTUUnset	false	The valid values are <b>true</b> and <b>false</b> .	If this property is set as <b>true</b> : 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 <b>false</b> : 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 G-1 DCPL Properties (continued)

overwriteInterfaceDescription	true	The valid values are <b>true</b> and <b>false</b> .	By default, Prime Provisioning generates a <b>description</b> 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 <b>description</b> subcommand (Example: crypto map entries, gre Interfaces, and so on).
transferUNIDescToVlanName	false	The valid values are <b>true</b> and <b>false</b> .	Controls provisioning of the VLAN name on the PE-POP. If set to <b>true</b> , the VLAN name is assigned from the description for the UNI. If set to the default of <b>false</b> , no VLAN name is assigned.
useSRDescriptionToGenerateDebug Data	false	The valid values are <b>true</b> and <b>false</b> .	This property is used to generate more intuitive debug data for easy fixing of issues.
<b>staging/</b>			
<b>platform/</b>			Platform related classes.
<b>CATOS/</b>			Service blade parameters for CATOS.
serviceBladeClass	com.cisco.vpnsc. prov.staging. StagingService Blade	string	ServiceBladeClass location.
<b>CISCO_ROUTER/</b>			IOS.
serviceBladeClass	com.cisco.vpnsc. prov.staging. StagingService Blade	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 <b>true</b> and <b>false</b> .	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 <b>true</b> and <b>false</b> .	If this property is set to <b>true</b> , whenever an SR is provisioned, the uploaded config files and input XML data are saved to a temporary directory for debugging purposes.

Table G-1 DCPL Properties (continued)

sendAuditEvent	true	The valid values are <b>true</b> and <b>false</b> .	Set <b>true</b> 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.
<b>vpls/</b>			Contains properties for Virtual Private LAN Service.
<b>DownloadWeights/</b>			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 <b>true</b> and <b>false</b> .	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.
<b>platform/</b>			Platform related classes.
<b>CATOS/</b>			Service blade parameters for CATOS.
serviceBladeClass	com.cisco.vpnsc. prov.vpls. VplsService Blade	string	ServiceBladeClass location.
<b>CISCO_ROUTER/</b>			IOS.
serviceBladeClass	com.cisco.vpnsc. prov.vpls. VplsService Blade	string	ServiceBladeClass location.

Table G-1 DCPL Properties (continued)

saveDebugData	true	The valid values are <b>true</b> and <b>false</b> .	If this property is set to <b>true</b> , 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 <b>true</b> and <b>false</b> .	Set <b>true</b> 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.
<b>SLA Properties:</b>			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 <b>true</b> and <b>false</b> .	If <b>true</b> and if showInRunningConfig is <b>true</b> - 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 <b>true</b> and <b>false</b> .	If true, the configured SLAs appear in the router's running configuration.

Table G-1 DCPL Properties (continued)

<b>SYSTEM Properties:</b>			The properties common to all sub-systems in Prime Provisioning 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 Provisioning 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 Provisioning.
from	<mailfrom>	string	The <b>from</b> field in the e-mail header of the mails sent out by Prime Provisioning.
smtpHost	<mailhost>	string	The server using which e-mail messages from Prime Provisioning should be sent out.
/SYSTEM/fullyManaged/			Properties related to e-mails sent out by Prime Provisioning 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 Provisioning.
/SYSTEM/license/			Properties related to Prime Provisioning 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 G-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 Provisioning.
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.
<b>Scheduler Properties:</b>			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.
<b>Services Properties:</b>			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 G-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 <b>true</b> and <b>false</b> .	With the default value of <b>true</b> , you can force purge a Service Request. If the value is <b>false</b> , you cannot force purge a Service Request.
disableFallBack	false	The valid values are <b>true</b> and <b>false</b> .	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 <b>true</b> and <b>false</b> .	This prevents allocating VLAN ID 1 for services configured by Prime Provisioning. This is applicable for both auto allocation of VLAN from VLAN resource pool and manual allocation. Set this property to <b>true</b> to block Prime Provisioning from deploying services with VLAN ID 1
interfaceDescRegEx		string	Interface Configuration Regular Expression.
interfaceNameRegEx		string	Interface Name Regular Expression.
l2vpnGroupNameOptions	Prime Provisioning, 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 <b>true</b> and <b>false</b> .	This property is effective only for IOS XR L2VPN services. The default is <b>false</b> . When set to <b>true</b> , this configures pseudowire transport mode to VLANs.
<b>SnmpService Properties:</b>			The Snmp Service package provides APIs to perform SNMP get() and set() operations.
/SnmpService/misc			Advanced settings.

Table G-1 DCPL Properties (continued)

enableDebug	false	The valid values are <b>true</b> and <b>false</b> .	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 Provisioning 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 G-1 DCPL Properties (continued)

partialConfigAudit	false	The valid values are <b>true</b> and <b>false</b> .	When the value is the default of <b>false</b> , the config audit is not limited. When the value is set to <b>true</b> , 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 <b>true</b> and <b>false</b> .	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 <b>false</b> , RBAC permission checking is not enabled. When the values is set to <b>true</b> , 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.
<b>TaskManager Properties:</b>			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 G-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).
<b>TemplateManager Properties:</b>			
userTemplateAttrFile	<vpnsc_home>/UserTemplateAttr.xml	string	User template attribute file path and name.
<b>VpnInvServer Properties:</b>			
/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).
<b>aagent Properties:</b>			
/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 G-1 DCPL Properties (continued)

<b>cfr Properties:</b>			The Command Flow Runner component. This currently runs within the Tomcat server (in the Prime Provisioning web application) and is responsible for running MPLSOAM troubleshooting workflows.
/cfr/Diagnostics/			
disableTunnelDiagnostics	false	The valid values are <b>true</b> and <b>false</b> .	Set to <b>true</b> 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 <code>java.util.logging.SimpleFormatter</code> instead of the Prime Provisioning 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 <code>java.util.logging</code> package.
<b>lockmanager Properties:</b>			Component that handles device locking. When different jobs (such as provisioning) try to update the config on the device, they obtain <b>software</b> 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 <b>true</b> and <b>false</b> .	Determines if a software lock is to be applied to the devices in the CollectConfig task. If <b>true</b> , 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 G-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.
<b>nbi Properties:</b>			Northbound API (Nbi) component related defines.
/nbi/BackwardCompatible			Path for execQuery requests.
RecordNumber	false	The valid values are <b>true</b> and <b>false</b> .	For execQuery requests, the number embedded in the output class name include <b>Record</b> for the default, <b>false</b> , or <b>Record#1</b> for <b>true</b> .
/nbi/CompositeDir	<vpncsc_home>/resources/java/xml/com/cisco/vpncsc/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	<vpncsc_home>/resources/java/xml/com/cisco/vpncsc/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.vpncsc.nbi.io.NbiSimpleFormatter	string	File path and name.
/nbi/Logger	com.cisco.vpncsc.nbi.util.NbiVpncscLogger	string	File path and name.
/nbi/MetaCheckInterval	300000	string	Set the time for next meta check to happen.
/nbi/MetaDir	<vpncsc_home>/resources/java/xml/com/cisco/vpncsc/repository/meta/xml	string	Path to meta XML files. Do not change it or the meta XML will not be backed up.

Table G-1 DCPL Properties (continued)

/nbi/ProvidedReportMetaDir	<vpncsc_home>/resources/java/xml/com/cisco/vpncsc/repository/meta/xml	string	Path to Prime Provisioning provided report meta XML files. Do not change it or the report meta xml files will not be backed up.
/nbi/Reader	com.cisco.vpncsc.nbi.io.NbiSoapReader	string	File path and name.
/nbi/RequestParserMgr	com.cisco.vpncsc.nbi.parser.NbiRequestParserMgr	string	File path and name.
/nbi/SSLfilepath	<vpncsc_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.vpncsc.nbi.parser.NbiWsdlParser	string	File path and name.
/nbi/Validation	true	The valid values are <b>true</b> and <b>false</b> .	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 <b>true</b> and <b>false</b> .	SoapEncapsulation.
/nbi/Writer	com.cisco.vpncsc.nbi.io.NbiSoapWriter	string	File path and name.
/nbi/logHandler	com.cisco.vpncsc.nbi.util.VpncscLogHandler	string	Custom log handler for nbi. This handler allows NBI to use alternate formatter from default one used by rest of Prime Provisioning. In this case, NBI defaults to using SimpleFormatter which dumps simple output as opposed to XML output.

Table G-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).
<b>notification Properties:</b>			Event notification related defines.
/notification/Logger	com.cisco.vpnsc.nbi.util.NbiVpnscLogger	string	File path and name.
/notification/clientEnabled	false	The valid values are <b>true</b> and <b>false</b> .	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.VpnscLoggingHandler	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 G-1 DCPL Properties (continued)

/notification/username	admin	string	Both username and password are same as the ones used for GUI login.
<b>pal Properties:</b>			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 Provisioning 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.
<b>repository Properties:</b>			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 G-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 <b>true</b> and <b>false</b> .	The default value is <b>false</b> . When the value is set to <b>true</b> , the user wants a manual allocation of the address in the aged address to succeed. When the value is set to <b>true</b> , 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 <b>true</b> and <b>false</b> .	This property set at <b>true</b> 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 <b>true</b> and <b>false</b> .	The default value is <b>false</b> . When the value is set to <b>true</b> , 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 <b>true</b> and <b>false</b> .	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 G-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 <b>true</b> and <b>false</b> .	Specifies whether the elements in the RBAC cache are eternal, never expire. The value <b>true</b> indicates the elements in the cache are eternal and never expire. The default value <b>false</b> 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 <b>true</b> and <b>false</b> .	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/checkCreatorPermission Enabled	true	The valid values are <b>true</b> and <b>false</b> .	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 <b>true</b> and <b>false</b> .	The creator of objects can give the permissions of Modify or Delete to others. If this flag is false, enable RBAC permission checkin.



Table G-1 DCPL Properties (continued)

/repository/rbac/enableAutologin	true	The valid values are <b>true</b> and <b>false</b> .	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 <b>true</b> and <b>false</b> .	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 <b>false</b> .
/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 <b>true</b> and <b>false</b> .	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.
<b>watchdog Properties:</b>			All the servers in Prime Provisioning 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 G-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 cnserver 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 G-1 DCPL Properties (continued)

lowWatermark	<lowwater>	string	Low 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 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 <b>true</b> and <b>false</b> .	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 G-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 <b>watchdog.heartbeat.wds.masterReconnectAttemptDelay</b> 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 <b>java</b> .
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 G-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/ flags			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/ class			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/ discovery/			Handles various Prime Provisioning Discovery workflow related tasks.

Table G-1 DCPL Properties (continued)

class	com.cisco.vpnsc. discovery.engine. server.Discovery Server	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.Vpnsc DispatcherImpl	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 G-1 DCPL Properties (continued)

cmd	java com.cisco.vpnsc. watchdog.ext.Ser viceLauncherImp l	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 G-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 .WDLockManag er	string	Class that keeps track of lockmanager heartbeats.
cmd	java com.cisco.vpnsc. lockmanager.Loc kManagerImpl	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 G-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 .WDDNameServer	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 G-1 DCPL Properties (continued)

cmd	java com.cisco.vpnsc. scheduler.Schedu ler	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 G-1 DCPL Properties (continued)

app_args	Worker com.cisco.vpnsc. dist.WorkerImpl, com.cisco.vpnsc. sla.sql.SlaMainte nanceService, com.cisco.vpnsc. repository.ual.U ALCleanupServi ceImpl, com.cisco.vpnsc. license.LicenseS ynchronize, com.cisco.vpnsc. cleanup.TaskLog CleanupService, com.cisco.vpnsc. cleanup.TempFil eCleanupService, com.cisco.vpnsc. cleanup.Runtime TaskCleanupServ ice”	string	Arguments to the class specified in the <b>cmd</b> 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.Ser viceLauncherImp l	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 G-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	<mailto:Restart>	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.
<b>xml Properties:</b>			The component for XML-based properties.

Table G-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 <b>true</b> and <b>false</b> .	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.

