



CHAPTER 6

MPLS Provisioning

The service provider's backbone is comprised of the core provider edge (PE) device and its provider routers. An MPLS VPN consists of a set of customer sites that are interconnected through an MPLS provider core network. At each site, there are one or more customer edge (CE) devices, which attach to one or more PEs.

Cisco Prime Fulfillment provisioning engine for MPLS accesses the configuration files on both the CE and PE to compute the necessary changes to the configuration files to support the service on the PE to CE link (or PE to CLE, PE to MVRFCE to CE, or PE to MVRFCE to CLE).

This chapter describes MPLS VPN service concepts and the steps required to provision MPLS VPN services using the Prime Fulfillment API. The provisioning example includes the process flow from creating the inventory to auditing the service deployment.

For information on MPLS provisioning using the Prime Fulfillment GUI, see the [Cisco Prime Fulfillment User Guide 6.1](#).

This chapter contains the following sections:

- [MPLS Service Definitions, page 6-2](#)
- [MPLS Service Requests, page 6-8](#)
- [End-to-End Provisioning Process, page 6-35](#).

MPLS Service Definitions

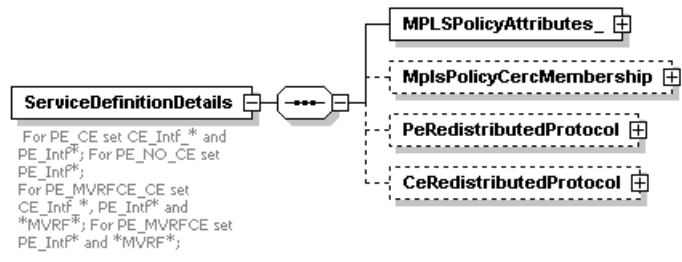
An MPLS service definition defines attributes for the policy type (**MPLSPolicyAttributes**), and can include the CE routing community (CERC) membership and redistributed protocol information.

CERC membership defines the CE routing community for this policy and is represented by the VPN routing and forwarding tables (VRFs), and the redistributed protocols define the metric attributes. MPLS policy attributes are described in the [MPLS Policy Attributes](#) section.

Schema Diagram

Figure 6-1 shows the schema diagram for an MPLS service definition.

Figure 6-1 MPLS Service Definition Schema Diagram



For each service definition property, you can set an additional attribute, **editable=true**, to allow the network operator to override these attributes when creating the service request. If an attribute is set to **editable=false**, these attributes cannot be changed in the service request.



Note

When a property has the additional attribute **editable=true**, all the related and child attributes are also editable.

See the following example:

```
<objectPath xsi:type="ns1:CIMObjectPath">
  <className xsi:type="xsd:string">ServiceDefinitionDetails</className>
  <properties xsi:type="ns1:CIMPropertyList"
    soapenc:arrayType="ns1:CIMProperty[]">
    <item xsi:type="ns1:CIMProperty">
      <name xsi:type="xsd:string">SubType</name>
      <value xsi:type="xsd:string">PE_CE</value>
    </item>
    <item xsi:type="ns1:CIMProperty">
      <name xsi:type="xsd:string">PE_Intf_Type</name>
      <value xsi:type="xsd:string">GigabitEthernet</value>
      <qualifier xsi:type="ns1:CIMQualifier">
        <name xsi:type="xsd:string">editable</name>
        <value xsi:type="xsd:string">true</value>
      </qualifier>
    </item>
    <item xsi:type="ns1:CIMProperty">
      <name xsi:type="xsd:string">PE_Intf_Desc</name>
      <value xsi:type="xsd:string">Interface Description</value>
    </item>
  </properties>
</objectPath>
```

```

<qualifier xsi:type="ns1:CIMQualifier">
  <name xsi:type="xsd:string">editable</name>
  <value xsi:type="xsd:string">true</value>
</qualifier>
</item>
<item xsi:type="ns1:CIMProperty">
  <name xsi:type="xsd:string">PE_Intf_Shutdown</name>
  <value xsi:type="xsd:string">TRUE</value>
  <qualifier xsi:type="ns1:CIMQualifier">
    <name xsi:type="xsd:string">editable</name>
    <value xsi:type="xsd:string">true</value>
  </qualifier>
</item>

```

Supported MPLS Operations

The following basic API operations are supported in this release:

Policies

- PE_CE
 - Create PE_CE Policy
 - Modify PE_CE Policy
 - Delete the PE_CE Policy
- PE_NOCE
 - Create PE_NOCE Policy
 - Modify the PE_NOCE Policy
 - Delete the PE_NOCE Policy
- MVRF_CE
 - Create MVRF_CE Policy
 - Modify MVRF_CE Policy
 - Delete MVRF_CE Policy
- MVRF_NO_CE
 - Create MVRF_NO_CE Policy
 - Modify MVRF_NOCE Policy
 - Delete MVRF_NOCE Policy

Service Requests

- PE_CE SR
 - Create PE_CE SR
 - Modify PE_CE SR
 - Delete the PE_CE SR
- PE_NOCE SR
 - Create PE_NOCE SR
 - Modify the PE_NOCE SR

- Delete the PE_NOCE SR
- MVRF_CE SR
 - Create MVRF_CE SR
 - Modify MVRF_CE SR
 - Delete MVRF_CE SR
- MVRF_NO_CE SR
 - Create MVRF_NO_CE SR
 - Modify MVRF_NOCE SR
 - Delete MVRF_NOCE SR

Besides the above, you can also perform the following operations:

- View MPLS Policy
- View MPLS SR Response
- View MPLS SR

The End-to-End Provisioning section of this document includes examples of inventory and resource pool operations.

MPLS Policy Attributes

The MPLS policy attributes include; the policy subtype, device interfaces and encapsulation types, IP addressing schemes, routing protocols, and VPN membership information. These values are set based on the policy subtype.

The following MPLS policy subtypes are supported:

- PE_CE—A standard MPLS policy for the PE_CE link. This is the default MPLS policy for Prime Fulfillment.
- PE_NO_CE—In this policy type, you specify only the PE interface information. The CE device at the customer location is not managed by Prime Fulfillment.
- PE_MVRFCE_CE—A Multi-VPN routing and forwarding CE (MVRFCE) network has multiple CE devices that connect to one MVRFCE device. The MVRFCE stores the VRFs for all VPNs in the customer network and connects directly with the PE device at the edge of the provider network. Prime Fulfillment manages the PE to MVRFCE to CE link.
- PE_MVRFCE_NO_CE—Prime Fulfillment manages the PE to MVRFCE link. The CE device at the customer site is not managed by Prime Fulfillment.

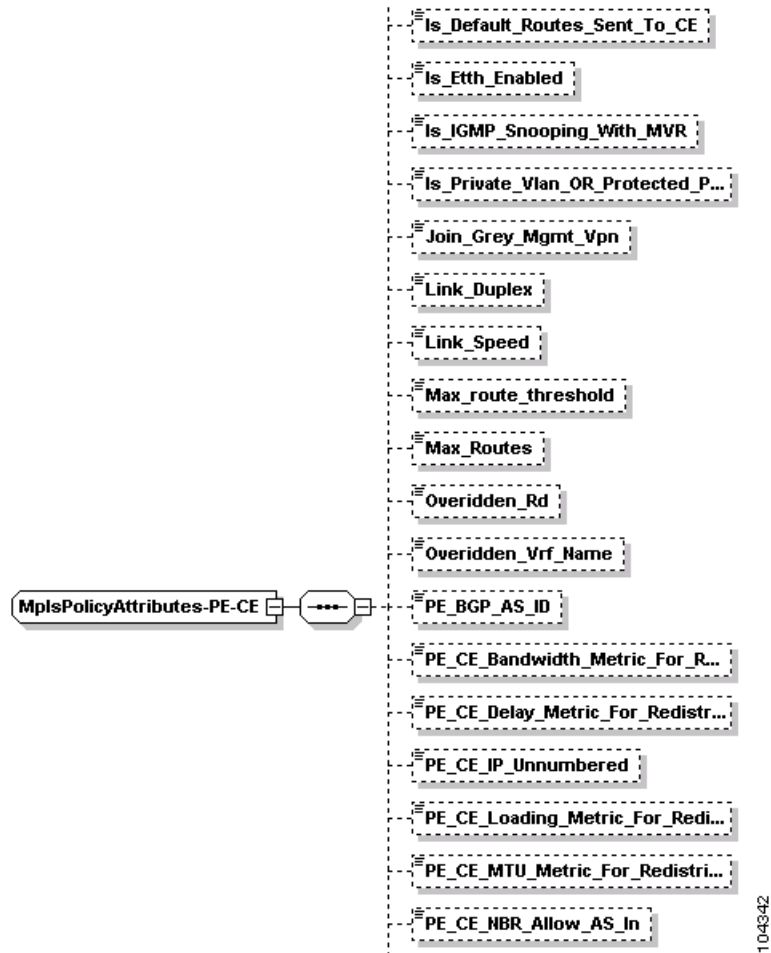


Note

For all policy subtypes with no CE present, you do not declare the CE devices to be CPEs, and you do not set policy attributes for the CE devices in the service definition.

There are numerous properties that can be set for an MPLS policy. [Figure 6-2](#) shows a partial schema diagram for the **MPLSPolicAttributes** with **SubType=PE_CE**.

Figure 6-2 MPLS Policy Attributes Schema Diagram



Policy Examples

The following XML gives an example of how policy attributes might be used.

Create Bundled/Physical Interface

This example describes a service definition for a bundle interface configuration. A policy is created with the interface type **Bundle-Ether** interface.

Example: CreateMPLSServiceDefn_Bundle_PE_CE_IPV4_IPV6.xml

```
<?xml version="1.0" encoding="UTF-8"?>
<soapenv:Envelope
  xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
  xmlns:soapenc="http://schemas.xmlsoap.org/soap/encoding/"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xmlns:ns0="http://www.cisco.com/cim-cx/2.0"
```

```

xmlns:ns1="urn:CIM">
<soapenv:Header>
  <ns0:message id="87855" timestamp="2002-12-13T14:55:38.885Z"
    sessiontoken="p36bttjwy1" />
</soapenv:Header>
<soapenv:Body>
  <ns1:createInstance>
    <objectPath xsi:type="ns1:CIMObjectPath">
      <className xsi:type="xsd:string">ServiceDefinition</className>
      <properties xsi:type="ns1:CIMPropertyList"
        soapenc:arrayType="ns1:CIMProperty[]">
        <item xsi:type="ns1:CIMProperty">
          <name xsi:type="xsd:string">Name</name>
          <value xsi:type="xsd:string">Bundle_PE_CE_IPV4_IPV6</value>
        </item>
        <item xsi:type="ns1:CIMProperty">
          <name xsi:type="xsd:string">Type</name>
          <value xsi:type="xsd:string">Mpls</value>
        </item>
        <item xsi:type="ns1:CIMProperty">
          <name xsi:type="xsd:string">Remarks</name>
          <value xsi:type="xsd:string">data remarks</value>
        </item>
      </properties>
      <objectPath xsi:type="ns1:CIMObjectPath">
        <className xsi:type="xsd:string">ServiceDefinitionDetails</className>
        <properties xsi:type="ns1:CIMPropertyList"
          soapenc:arrayType="ns1:CIMProperty[]">
          <item xsi:type="ns1:CIMProperty">
            <name xsi:type="xsd:string">SubType</name>
            <value xsi:type="xsd:string">PE_CE</value>
          </item>
          <item xsi:type="ns1:CIMProperty">
            <name xsi:type="xsd:string">PE_CE_IP_Unnumbered</name>
            <value xsi:type="xsd:string">>false</value>
            <qualifier xsi:type="ns1:CIMQualifier">
              <name xsi:type="xsd:string">editable</name>
              <value xsi:type="xsd:string">>true</value>
            </qualifier>
          </item>
        </properties>
      </objectPath>
    </createInstance>
  </body>
</soapenv:Body>
<!-- PE Intf Info -->
  <item xsi:type="ns1:CIMProperty">
    <name xsi:type="xsd:string">PE_Intf_Type</name>
    <value xsi:type="xsd:string">Bundle-Ether</value>
    <qualifier xsi:type="ns1:CIMQualifier">
      <name xsi:type="xsd:string">editable</name>
      <value xsi:type="xsd:string">>true</value>
    </qualifier>
  </item>
  <item xsi:type="ns1:CIMProperty">
    <name xsi:type="xsd:string">PE_Intf_Desc</name>
    <value xsi:type="xsd:string" />
    <qualifier xsi:type="ns1:CIMQualifier">
      <name xsi:type="xsd:string">editable</name>
      <value xsi:type="xsd:string">>true</value>
    </qualifier>
  </item>
  <item xsi:type="ns1:CIMProperty">
    <name xsi:type="xsd:string">PE_Intf_Format</name>
    <value xsi:type="xsd:string" />
    <qualifier xsi:type="ns1:CIMQualifier">
      <name xsi:type="xsd:string">editable</name>
      <value xsi:type="xsd:string">>true</value>
    </qualifier>
  </item>

```

```

    </qualifier>
  </item>
  <item xsi:type="ns1:CIMProperty">
    <name xsi:type="xsd:string">PE_Intf_Shutdown</name>
    <value xsi:type="xsd:string">>false</value>
    <qualifier xsi:type="ns1:CIMQualifier">
      <name xsi:type="xsd:string">editable</name>
      <value xsi:type="xsd:string">>true</value>
    </qualifier>
  </item>

  <item xsi:type="ns1:CIMProperty">
    <name xsi:type="xsd:string">Autopick_Vlan_ID</name>
    <value xsi:type="xsd:string">>false</value>
    <qualifier xsi:type="ns1:CIMQualifier">
      <name xsi:type="xsd:string">editable</name>
      <value xsi:type="xsd:string">>true</value>
    </qualifier>
  </item>

<!-- CE Intf Info -->
  <item xsi:type="ns1:CIMProperty">
    <name xsi:type="xsd:string">CE_Intf_Type</name>
    <value xsi:type="xsd:string">ANY</value>
    <qualifier xsi:type="ns1:CIMQualifier">
      <name xsi:type="xsd:string">editable</name>
      <value xsi:type="xsd:string">>true</value>
    </qualifier>
  </item>
  <item xsi:type="ns1:CIMProperty">
    <name xsi:type="xsd:string">CE_Intf_Desc</name>
    <value xsi:type="xsd:string" />
    <qualifier xsi:type="ns1:CIMQualifier">
      <name xsi:type="xsd:string">editable</name>
      <value xsi:type="xsd:string">>true</value>
    </qualifier>
  </item>
  <item xsi:type="ns1:CIMProperty">
    <name xsi:type="xsd:string">CE_Intf_Format</name>
    <value xsi:type="xsd:string" />
    <qualifier xsi:type="ns1:CIMQualifier">
      <name xsi:type="xsd:string">editable</name>
      <value xsi:type="xsd:string">>true</value>
    </qualifier>
  </item>

<!-- IP Address -->
  <item xsi:type="ns1:CIMProperty">
    <name xsi:type="xsd:string">Auto_Assign_IP_Address</name>
    <value xsi:type="xsd:string">>false</value>
    <qualifier xsi:type="ns1:CIMQualifier">
      <name xsi:type="xsd:string">editable</name>
      <value xsi:type="xsd:string">>true</value>
    </qualifier>
  </item>

<!-- Routing Protocols IPv4 & IPV6 -->
  <item xsi:type="ns1:CIMProperty">
    <name xsi:type="xsd:string">PE_CE_Routing_Protocol</name>
    <value xsi:type="xsd:string">BGP</value>
    <qualifier xsi:type="ns1:CIMQualifier">
      <name xsi:type="xsd:string">editable</name>
      <value xsi:type="xsd:string">>true</value>

```

```

        </qualifier>
    </item>

    <item xsi:type="ns1:CIMProperty">
        <name xsi:type="xsd:string">PE_CE_IPV6_Routing_Protocol</name>
        <value xsi:type="xsd:string">BGP</value>
        <qualifier xsi:type="ns1:CIMQualifier">
            <name xsi:type="xsd:string">editable</name>
            <value xsi:type="xsd:string">true</value>
        </qualifier>
    </item>

    <item xsi:type="ns1:CIMProperty">
        <name xsi:type="xsd:string">Template_Enabled</name>
        <value xsi:type="xsd:string">true</value>
        <qualifier xsi:type="ns1:CIMQualifier">
            <name xsi:type="xsd:string">editable</name>
            <value xsi:type="xsd:string">true</value>
        </qualifier>
    </item>

</properties>

<!-- CERC -->
<objectPath xsi:type="ns1:CIMObjectPath">
<className xsi:type="xsd:string">MplsPolicyCercMembership</className>
<properties xsi:type="ns1:CIMPropertyList"
    soapenc:arrayType="ns1:CIMProperty[]">
    <item xsi:type="ns1:CIMProperty">
        <name xsi:type="xsd:string">CERC</name>
        <value xsi:type="xsd:string">cerc1</value>
    </item>
    <item xsi:type="ns1:CIMProperty">
        <name xsi:type="xsd:string">IsHub</name>
        <value xsi:type="xsd:string">true</value>
    </item>
</properties>
</objectPath>

<!-- Templates -->

</objectPath>
</objectPath>
</ns1:createInstance>
</soapenv:Body>
</soapenv:Envelope>

```

Related APIs

- [CreateMPLSServiceDefn_Bundle_PE_CE_IPV4_IPV6.xml](#)
- [CreateMPLSServiceDefn_Bundle_PE_NO_CE_IPV4_IPV6.xml](#)

MPLS Service Requests

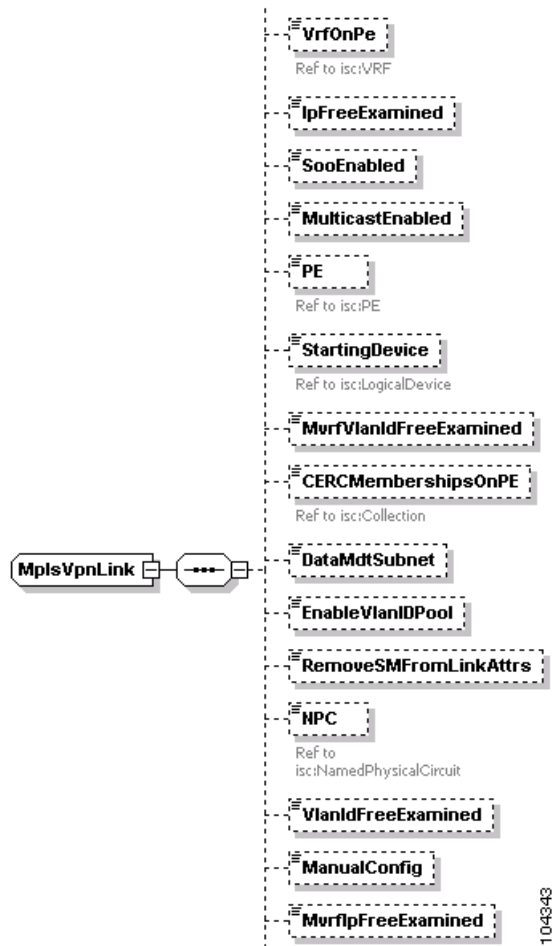
An MPLS service request specifies the MPLS policy (service definition) to use, the list of links, referred to as MPLS VPN links, and the link attributes. Use the link attribute settings in the service request to override any policy settings defined as editable in the service definition.

**Note**

You can also integrate an Prime Fulfillment template with an MPLS service request and associate one or more templates to the CPE and PE devices. See [Chapter 4, “Using Templates,”](#) for more information.

Prime Fulfillment supports an extensive list of properties that can be set for MPLS VPN links. [Figure 6-3](#) shows a partial schema diagram for the **MplsVpnLink** in a service request.

Figure 6-3 MPLS VPN Link Schema Diagram



Service Request Examples

The following XML examples illustrate how MPLS APIs can be used. They also demonstrate the kinds of properties that need to be specified for each request.

In the following examples, significant attributes are highlighted in bold.

Examples of these three features are included below:

- [Create Independent Route Target, page 6-10](#)
- [Create Static Route Configuration, page 6-16](#)

- [Create BGP Max Prefix with Second VLAN ID and BGP DIO, page 6-20](#)
- [Create EIGRP Key Chain, page 6-26](#)
- [Create OSPF Default Info Originate, page 6-32](#)

Create Independent Route Target

In this example, a service order is created with a unique route target for IPv4 and IPv6.

The attribute relevant for the unique route target example is **CERC_RTTYPE** (in **bold** below). The preferred values for **CERC_RTTYPE** tag are IPv4, IPv6 and dual. If the RT Type mentioned does not match with IPv4 or IPv6 or dual, it will throw an error.

Note the following:

- The unique RT feature is supported only for IOS XR devices and only when the corresponding DCPL property is set to true.
- When a service request is created, the RT Type of the CERCs mentioned in the CERC_MEMBERSHIP should be indicated.
- When modifying a service request, a new CERC of RT Type can be added according to the address family only. Otherwise, it will throw error.
- If the **CERC_RTTYPE** tag is not indicated when modifying the script, the newly added CERC will take RT Type as its address family.
- When modifying a service request, the **CERC_RTTYPE** of an existing CERC can be modified.
- When modifying a CERC of RT Type dual to IPv6, pass the CERC name with the RT Type to be modified.

Example:

cerc1=dual is the RT Type at the time the service request is created. This now needs to be changed to IPv6 and when modifying the script, pass cerc1=IPv6 in the **CERC_RTTYPE** tag. Then the cerc1 RT configuration will be changed to IPv6.

If a CERC needs to be deleted, specify the CERC name in deleteInstance. The specified CERC will be deleted.



Note

When creating a service request for dual address family, the RT configuration can be IPv4 and IPv6, or dual. There should be two CERCs, one with IPv4 and another with IPv6, or one CERC with the RT configuration dual. If a service request is created for IPV4 address family, the RT configuration should be only IPv4, and if it is created for IPv6 address family, it should be only IPv6.

Example: CreateMPLSServiceOrder_PE_NO_CE_IPV4_IPV6_Unique_RT.xml

```
<?xml version="1.0" encoding="UTF-8"?>
<soapenv:Envelope
  xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
  xmlns:soapenc="http://schemas.xmlsoap.org/soap/encoding/"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xmlns:ns0="http://www.cisco.com/cim-cx/2.0"
  xmlns:ns1="urn:CIM">
  <soapenv:Header>
    <!-- WaitTimeout has a default set in system properties.-->
    <ns0:message id="87855" timestamp="2002-12-13T14:55:38.885Z"
      Wait="true" WaitTimeout="90" sessiontoken="p36bttjwy1"/>
  </soapenv:Header>
</soapenv:Envelope>
```

```

</soapenv:Header>
<soapenv:Body>
  <ns1:performBatchOperation>
    <actions xsi:type="ns1:CIMActionList"
      soapenc:arrayType="ns1:CIMAction[]">
      <action>
        <actionName xsi:type="xsd:string">createInstance</actionName>
        <objectPath xsi:type="ns1:CIMObjectPath">
          <className xsi:type="xsd:string">ServiceOrder</className>
          <properties xsi:type="ns1:CIMPropertyList"
            soapenc:arrayType="ns1:CIMProperty[]">
            <item xsi:type="ns1:CIMProperty">
              <name xsi:type="xsd:string">ServiceName</name>
              <value xsi:type="xsd:string">ServiceOrder253</value>
            </item>
            <item xsi:type="ns1:CIMProperty">
              <name xsi:type="xsd:string">CarrierId</name>
              <value xsi:type="xsd:string">322</value>
            </item>
            <item xsi:type="ns1:CIMProperty">
              <name xsi:type="xsd:string">DesiredDueDate</name>
              <value xsi:type="xsd:dateTime">2002-12-13T14:55:38.000Z</value>
            </item>
            <item xsi:type="ns1:CIMProperty">
              <name xsi:type="xsd:string">NumberOfRequests</name>
              <value xsi:type="xsd:string">1</value>
            </item>
            <!--item xsi:type="ns1:CIMProperty">
              <name xsi:type="xsd:string">Organization</name>
              <value xsi:type="xsd:string">NbiCustomer</value>
            </item-->
          </properties>
        </objectPath>
      </action>
      <action>
        <actionName xsi:type="xsd:string">createInstance</actionName>
        <objectPath xsi:type="ns1:CIMObjectPath">
          <className xsi:type="xsd:string">ServiceRequest</className>
          <properties xsi:type="ns1:CIMPropertyList"
            soapenc:arrayType="ns1:CIMProperty[]">
            <item xsi:type="ns1:CIMProperty">
              <name xsi:type="xsd:string">RequestName</name>
              <value xsi:type="xsd:string">MYSR-1</value>
            </item>
            <item xsi:type="ns1:CIMProperty">
              <name xsi:type="xsd:string">Type</name>
              <value xsi:type="xsd:string">Mpls</value>
            </item>
          </properties>
        <objectPath xsi:type="ns1:CIMObjectPath">
          <className xsi:type="xsd:string">ServiceRequestDetails</className>
          <properties xsi:type="ns1:CIMPropertyList"
            soapenc:arrayType="ns1:CIMProperty[]">
            <!-- Policy Info -->
            <item xsi:type="ns1:CIMProperty">
              <name xsi:type="xsd:string">ServiceDefinition</name>
              <value xsi:type="xsd:string">PE_NO_CE</value>
              <qualifier xsi:type="xsd:string">
                <name xsi:type="xsd:string">ServiceDefinitionType</name>
                <value xsi:type="xsd:string">Mpls</value>
              </qualifier>
            </item>
          </properties>
        </objectPath xsi:type="ns1:CIMObjectPath">

```

```

<className xsi:type="xsd:string">MplsVpnLink</className>
<properties xsi:type="ns1:CIMPropertyList"
  soapenc:arrayType="ns1:CIMProperty[]">

<!-- ManualConfig is needed when specifying PE-CE link and not using NPC -->
<!-- Device Info -->
  <item xsi:type="ns1:CIMProperty">
    <name xsi:type="xsd:string">ManualConfig</name>
    <value xsi:type="xsd:string">>true</value>
  </item>
  <item xsi:type="ns1:CIMProperty">
    <name xsi:type="xsd:string">PE</name>
    <value xsi:type="xsd:string">CRS</value>
  </item>

</properties>
<objectPath xsi:type="ns1:CIMObjectPath">
  <className xsi:type="xsd:string">LinkAttrs</className>
  <properties xsi:type="ns1:CIMPropertyList"
    soapenc:arrayType="ns1:CIMProperty[]">

<!-- PE Intf Info -->
  <item xsi:type="ns1:CIMProperty">
    <name xsi:type="xsd:string">PE_Intf_Name</name>
    <value xsi:type="xsd:string">Bundle-Ether123</value>
  </item>

  <item xsi:type="ns1:CIMProperty">
    <name xsi:type="xsd:string">PE_Intf_Encap</name>
    <value xsi:type="xsd:string">DOT1Q</value>
  </item>

  <item xsi:type="ns1:CIMProperty">
    <name xsi:type="xsd:string">PE_Vlan_ID</name>
    <value xsi:type="xsd:string">1600</value>
  </item>
  <item xsi:type="ns1:CIMProperty">
    <name xsi:type="xsd:string">Autopick_Vlan_ID</name>
    <value xsi:type="xsd:string">>false</value>
  </item>

<!-- interface Addresses for IPV4 -->

  <item xsi:type="ns1:CIMProperty">
    <name xsi:type="xsd:string">PE_Intf_Address</name>
    <value xsi:type="xsd:string">12.20.19.47/24</value>
  </item>

<!-- interface Addresses for IPV6 -->

  <item xsi:type="ns1:CIMProperty">
    <name xsi:type="xsd:string">PE_Intf_Address_IPV6</name>
    <value xsi:type="xsd:string">2009::47/32</value>
  </item>

  <item xsi:type="ns1:CIMProperty">
    <name xsi:type="xsd:string">Auto_Assign_IP_Address</name>
    <value xsi:type="xsd:string">>false</value>
  </item>

<!-- IPV4 Routing Protocol -->
  <item xsi:type="ns1:CIMProperty">
    <name xsi:type="xsd:string">PE_CE_Routing_Protocol</name>
    <value xsi:type="xsd:string">STATIC</value>
  </item>

```

```

        </item>

<!-- IPv6 Routing Protocol -->
    <item xsi:type="ns1:CIMProperty">
        <name xsi:type="xsd:string">PE_CE_IPv6_Routing_Protocol</name>
        <value xsi:type="xsd:string">NONE</value>
    </item>

<!-- CERC Type -->
    <item xsi:type="ns1:CIMProperty">
        <name xsi:type="xsd:string">CERC_RTTYPE</name>
        <value xsi:type="xsd:string">cerc3=IPv4,cerc5=IPv6</value>
    </item>

<!-- Enable RD Format -->

    <!--item xsi:type="ns1:CIMProperty">
        <name xsi:type="xsd:string">RD_FORMAT</name>
        <value xsi:type="xsd:string">RD_AS</value>
    </item-->

    </properties>
</objectPath>

<!-- CERC's -->

<objectPath xsi:type="ns1:CIMObjectPath">
    <className xsi:type="xsd:string">CERCMembership</className>
    <properties xsi:type="ns1:CIMPropertyList"
        soapenc:arrayType="ns1:CIMProperty[]">
        <item xsi:type="ns1:CIMProperty">
            <name xsi:type="xsd:string">CERC</name>
            <value xsi:type="xsd:string">cerc3</value>
        </item>
        <item xsi:type="ns1:CIMProperty">
            <name xsi:type="xsd:string">IsHub</name>
            <value xsi:type="xsd:string">>true</value>
        </item>
    </properties>
</objectPath>

<objectPath xsi:type="ns1:CIMObjectPath">
    <className xsi:type="xsd:string">CERCMembership</className>
    <properties xsi:type="ns1:CIMPropertyList"
        soapenc:arrayType="ns1:CIMProperty[]">
        <item xsi:type="ns1:CIMProperty">
            <name xsi:type="xsd:string">CERC</name>
            <value xsi:type="xsd:string">cerc5</value>
        </item>
        <item xsi:type="ns1:CIMProperty">
            <name xsi:type="xsd:string">IsHub</name>
            <value xsi:type="xsd:string">>true</value>
        </item>
    </properties>
</objectPath>

<!-- LinkTemplate example using template definition and direct input of template data
(buffer method). The template data
        is entered in the DataBuffer objectDef section -->
<!-- Template start here-->

<!-- Template Ends Here -->

```

```

</objectPath>

<objectPath xsi:type="ns1:CIMObjectPath">
  <className xsi:type="xsd:string">MplsVpnLink</className>
  <properties xsi:type="ns1:CIMPropertyList"
    soapenc:arrayType="ns1:CIMProperty[]">

<!-- ManualConfig is needed when specifying PE-CE link and not using NPC -->
<!-- Device Info -->
  <item xsi:type="ns1:CIMProperty">
    <name xsi:type="xsd:string">ManualConfig</name>
    <value xsi:type="xsd:string">true</value>
  </item>
  <item xsi:type="ns1:CIMProperty">
    <name xsi:type="xsd:string">PE</name>
    <value xsi:type="xsd:string">CRS</value>
  </item>

</properties>
<objectPath xsi:type="ns1:CIMObjectPath">
  <className xsi:type="xsd:string">LinkAttrs</className>
  <properties xsi:type="ns1:CIMPropertyList"
soapenc:arrayType="ns1:CIMProperty[]">

<!-- PE Intf Info -->
  <item xsi:type="ns1:CIMProperty">
    <name xsi:type="xsd:string">PE_Intf_Name</name>
    <value xsi:type="xsd:string">Bundle-Ether123</value>
  </item>

  <item xsi:type="ns1:CIMProperty">
    <name xsi:type="xsd:string">PE_Intf_Encap</name>
    <value xsi:type="xsd:string">DOT1Q</value>
  </item>

  <item xsi:type="ns1:CIMProperty">
    <name xsi:type="xsd:string">PE_Vlan_ID</name>
    <value xsi:type="xsd:string">1600</value>
  </item>
  <item xsi:type="ns1:CIMProperty">
    <name xsi:type="xsd:string">Autopick_Vlan_ID</name>
    <value xsi:type="xsd:string">>false</value>
  </item>

<!-- interface Addresses for IPV4 -->

  <item xsi:type="ns1:CIMProperty">
    <name xsi:type="xsd:string">PE_Intf_Address</name>
    <value xsi:type="xsd:string">12.20.19.47/24</value>
  </item>

<!-- interface Addresses for IPV6 -->

  <item xsi:type="ns1:CIMProperty">
    <name xsi:type="xsd:string">Auto_Assign_IP_Address</name>
    <value xsi:type="xsd:string">>false</value>
  </item>

<!-- IPV4 Routing Protocol -->
  <item xsi:type="ns1:CIMProperty">
    <name xsi:type="xsd:string">PE_CE_Routing_Protocol</name>
    <value xsi:type="xsd:string">STATIC</value>
  </item>

```

```

<!-- IPv6 Routing Protocol -->
<!-- <item xsi:type="ns1:CIMProperty">
    <name xsi:type="xsd:string">PE_CE_IPV6_Routing_Protocol</name>
    <value xsi:type="xsd:string">EIGRP</value>
</item>
<item xsi:type="ns1:CIMProperty">
    <name xsi:type="xsd:string">PE_EIGRP_AS_ID</name>
    <value xsi:type="xsd:string">2369</value>
</item>
<item xsi:type="ns1:CIMProperty">
    <name xsi:type="xsd:string">CE_EIGRP_AS_ID</name>
    <value xsi:type="xsd:string">1526</value>
</item>
<item xsi:type="ns1:CIMProperty">
    <name
xsi:type="xsd:string">PE_CE_Bandwidth_Metric_For_Redistribution</name>
    <value xsi:type="xsd:string">6311</value>
</item>
<item xsi:type="ns1:CIMProperty">
    <name
xsi:type="xsd:string">PE_CE_Delay_Metric_For_Redistribution</name>
    <value xsi:type="xsd:string">5312</value>
</item>
<item xsi:type="ns1:CIMProperty">
    <name
xsi:type="xsd:string">PE_CE_Reliability_Metric_For_Redistribution</name>
    <value xsi:type="xsd:string">169</value>
</item>
<item xsi:type="ns1:CIMProperty">
    <name
xsi:type="xsd:string">PE_CE>Loading_Metric_For_Redistribution</name>
    <value xsi:type="xsd:string">147</value>
</item>
<item xsi:type="ns1:CIMProperty">
    <name xsi:type="xsd:string">PE_CE_MTU_Metric_For_Redistribution</name>
    <value xsi:type="xsd:string">864</value>
</item>

<!-- CERC Type -->

<!-- Enable Multipath Loadsharing -->

<!-- Enable RD Format -->

    </properties>
</objectPath>

<!-- CERC's -->

<objectPath xsi:type="ns1:CIMObjectPath">
    <className xsi:type="xsd:string">CERCMembership</className>
    <properties xsi:type="ns1:CIMPropertyList"
        soapenc:arrayType="ns1:CIMProperty[]">
        <item xsi:type="ns1:CIMProperty">
            <name xsi:type="xsd:string">CERC</name>
            <value xsi:type="xsd:string">multi-cerc3</value>
        </item>
        <item xsi:type="ns1:CIMProperty">
            <name xsi:type="xsd:string">IsHub</name>
            <value xsi:type="xsd:string">>true</value>
        </item>
    </properties>

```

```

        </objectPath>

<!-- LinkTemplate example using template definition and direct input of template data
(buffer method). The template data
                                is entered in the DataBuffer objectDef section -->
<!-- TEmplate start here-->

<!-- Template Ends Here -->
        </objectPath>

        </objectPath>
    </objectPath>
</action>
</actions>
</ns1:performBatchOperation>
</soapenv:Body>
</soapenv:Envelope>

```

Create Static Route Configuration

In this example XML script, a static route configuration is created. Both the IPv4 attribute **Static_Next_Hop_Option** and the IPv6 attribute **Static_Next_Hop_Option_IPV6** are given the new value **OUTGOING_INTF_NAME_NEXT_HOP_IPADDR** when the service request is created.

The preferred values for **Static_Next_Hop_Option** are **OUT_GOING_INTF_NAME**, **NEXT_HOP_IPADDR**, and **OUTGOING_INTF_NAME_NEXT_HOP_IPADDR**.

Other relevant attributes are the IPv4 attribute **STATIC_NEXT_HOP_IP_ADDR**, which is set to 2.22.29.28 and the IPv6 attribute **STATIC_NEXT_HOP_IPV6_ADDR**, which is set to 89::68.



Note

In the PE_CE case, no values have been specified for Next Hop Ip Address. For PE_NO_CE, the Next Hop Ip Address value should be specified when selecting **NEXT_HOP_IPADDR** or **OUTGOING_INTF_NAME_NEXT_HOP_IPADDR** for **Static_Next_Hop_Option**.

File: CreateMPLSServiceOrder_StaticRoute_PE_NO_CE.xml

```

<?xml version="1.0" encoding="UTF-8"?>
<soapenv:Envelope
  xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
  xmlns:soapenc="http://schemas.xmlsoap.org/soap/encoding/"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xmlns:ns0="http://www.cisco.com/cim-cx/2.0"
  xmlns:ns1="urn:CIM">
  <soapenv:Header>
    <!-- WaitTimeout has a default set in system properties.-->
    <ns0:message id="87855" timestamp="2002-12-13T14:55:38.885Z"
      Wait="true" WaitTimeout="90" sessiontoken="p36bttjwy1"/>
  </soapenv:Header>
  <soapenv:Body>
    <ns1:performBatchOperation>
      <actions xsi:type="ns1:CIMActionList"
        soapenc:arrayType="ns1:CIMAction[]">
        <action>
          <actionName xsi:type="xsd:string">createInstance</actionName>
          <objectPath xsi:type="ns1:CIMObjectPath">
            <className xsi:type="xsd:string">ServiceOrder</className>

```



```

<properties xsi:type="ns1:CIMPropertyList"
    soapenc:arrayType="ns1:CIMProperty[]">
  <item xsi:type="ns1:CIMProperty">
    <name xsi:type="xsd:string">ServiceName</name>
    <value xsi:type="xsd:string">ServiceOrder253</value>
  </item>
  <item xsi:type="ns1:CIMProperty">
    <name xsi:type="xsd:string">CarrierId</name>
    <value xsi:type="xsd:string">322</value>
  </item>
  <item xsi:type="ns1:CIMProperty">
    <name xsi:type="xsd:string">DesiredDueDate</name>
    <value xsi:type="xsd:dateTime">2002-12-13T14:55:38.000Z</value>
  </item>
  <item xsi:type="ns1:CIMProperty">
    <name xsi:type="xsd:string">NumberOfRequests</name>
    <value xsi:type="xsd:string">1</value>
  </item>
  <!--item xsi:type="ns1:CIMProperty">
    <name xsi:type="xsd:string">Organization</name>
    <value xsi:type="xsd:string">NbiCustomer</value>
  </item-->
</properties>
</objectPath>
</action>
<action>
  <actionName xsi:type="xsd:string">createInstance</actionName>
  <objectPath xsi:type="ns1:CIMObjectPath">
    <className xsi:type="xsd:string">ServiceRequest</className>
    <properties xsi:type="ns1:CIMPropertyList"
        soapenc:arrayType="ns1:CIMProperty[]">
      <item xsi:type="ns1:CIMProperty">
        <name xsi:type="xsd:string">RequestName</name>
        <value xsi:type="xsd:string">MYSR-1</value>
      </item>
      <item xsi:type="ns1:CIMProperty">
        <name xsi:type="xsd:string">Type</name>
        <value xsi:type="xsd:string">Mpls</value>
      </item>
    </properties>
    <objectPath xsi:type="ns1:CIMObjectPath">
      <className xsi:type="xsd:string">ServiceRequestDetails</className>
      <properties xsi:type="ns1:CIMPropertyList"
          soapenc:arrayType="ns1:CIMProperty[]">
        <!-- Policy Info -->
        <item xsi:type="ns1:CIMProperty">
          <name xsi:type="xsd:string">ServiceDefinition</name>
          <value xsi:type="xsd:string">PE_NO_CE</value>
          <qualifier xsi:type="xsd:string">
            <name xsi:type="xsd:string">ServiceDefinitionType</name>
            <value xsi:type="xsd:string">Mpls</value>
          </qualifier>
        </item>
      </properties>
      <objectPath xsi:type="ns1:CIMObjectPath">
        <className xsi:type="xsd:string">MplsVpnLink</className>
        <properties xsi:type="ns1:CIMPropertyList"
            soapenc:arrayType="ns1:CIMProperty[]">
          <!-- ManualConfig is needed when specifying PE-CE link and not using NPC -->
          <!-- Device Info -->
          <item xsi:type="ns1:CIMProperty">
            <name xsi:type="xsd:string">ManualConfig</name>
            <value xsi:type="xsd:string">true</value>
          </item>
        </properties>
      </objectPath>
    </properties>
  </objectPath>
</action>

```

```

    </item>
    <item xsi:type="ns1:CIMProperty">
      <name xsi:type="xsd:string">PE</name>
      <value xsi:type="xsd:string">Cisco CRS</value>
    </item>

  </properties>
  <objectPath xsi:type="ns1:CIMObjectPath">
    <className xsi:type="xsd:string">LinkAttrs</className>
    <properties xsi:type="ns1:CIMPropertyList"
soapenc:arrayType="ns1:CIMProperty[]">

<!-- PE Intf Info -->
  <item xsi:type="ns1:CIMProperty">
    <name xsi:type="xsd:string">PE_Intf_Name</name>
    <value xsi:type="xsd:string">TenGigE0/0/0/0</value>
  </item>
  <item xsi:type="ns1:CIMProperty">
    <name xsi:type="xsd:string">PE_Intf_Encap</name>
    <value xsi:type="xsd:string">DOT1Q</value>
  </item>

  <item xsi:type="ns1:CIMProperty">
    <name xsi:type="xsd:string">PE_Vlan_ID</name>
    <value xsi:type="xsd:string">1709</value>
  </item>
  <item xsi:type="ns1:CIMProperty">
    <name xsi:type="xsd:string">Autopick_Vlan_ID</name>
    <value xsi:type="xsd:string">>false</value>
  </item>

<!-- interface Addresses for IPV4 -->

  <item xsi:type="ns1:CIMProperty">
    <name xsi:type="xsd:string">PE_Intf_Address</name>
    <value xsi:type="xsd:string">78.69.19.37/24</value>
  </item>

<!-- interface Addresses for IPV6 -->
  <item xsi:type="ns1:CIMProperty">
    <name xsi:type="xsd:string">PE_Intf_Address_IPV6</name>
    <value xsi:type="xsd:string">3::32/64</value>
  </item>

  <item xsi:type="ns1:CIMProperty">
    <name xsi:type="xsd:string">Auto_Assign_IP_Address</name>
    <value xsi:type="xsd:string">>false</value>
  </item>

<!-- IPV4 Routing Protocol -->
  <item xsi:type="ns1:CIMProperty">
    <name xsi:type="xsd:string">PE_CE_Routing_Protocol</name>
    <value xsi:type="xsd:string">STATIC</value>
  </item>
  <item xsi:type="ns1:CIMProperty">
    <name xsi:type="xsd:string">PE_CE_IPV6_Routing_Protocol</name>
    <value xsi:type="xsd:string">STATIC</value>
  </item>
  <item xsi:type="ns1:CIMProperty">
    <name xsi:type="xsd:string">Is_Default_Routes_Sent_To_CE</name>
    <value xsi:type="xsd:string">TRUE</value>
  </item>
  <item xsi:type="ns1:CIMProperty">
    <name xsi:type="xsd:string">Is_Default_Routes_Sent_To_CE_IPV6</name>

```

```

        <value xsi:type="xsd:string">TRUE</value>
    </item>
    <item xsi:type="ns1:CIMProperty">
        <name xsi:type="xsd:string">Is_Default_Info_Originate</name>
        <value xsi:type="xsd:string">TRUE</value>
    </item>
    <item xsi:type="ns1:CIMProperty">
        <name xsi:type="xsd:string">Static_Next_Hop_Option</name>
        <value
xsi:type="xsd:string">OUTGOING_INTF_NAME_NEXT_HOP_IPADDR</value>
        </item>
    <item xsi:type="ns1:CIMProperty">
        <name xsi:type="xsd:string">STATIC_NEXT_HOP_IP_ADDR</name>
        <value xsi:type="xsd:string">2.22.29.28</value>
    </item>
    <item xsi:type="ns1:CIMProperty">
        <name xsi:type="xsd:string">Static_Next_Hop_Option_IPV6</name>
        <value
xsi:type="xsd:string">OUTGOING_INTF_NAME_NEXT_HOP_IPADDR</value>
        </item>
    <item xsi:type="ns1:CIMProperty">
        <name xsi:type="xsd:string">STATIC_NEXT_HOP_IPV6_ADDR</name>
        <value xsi:type="xsd:string">89::68</value>
    </item>
</properties>
</objectPath>
<objectPath xsi:type="ns1:CIMObjectPath">
    <className xsi:type="xsd:string">AdvertisedRoutesForCe</className>
    <properties xsi:type="ns1:CIMPropertyList"
        soapenc:arrayType="ns1:CIMProperty[]">
        <item xsi:type="ns1:CIMProperty">
            <name xsi:type="xsd:string">Advr_Routes_IP_Address</name>
            <value xsi:type="xsd:string">2.5.16.34/32</value>
        </item>
        <item xsi:type="ns1:CIMProperty">
            <name xsi:type="xsd:string">Advr_Routes_Metric</name>
            <value xsi:type="xsd:string">18</value>
        </item>
    </properties>
</objectPath>
<objectPath xsi:type="ns1:CIMObjectPath">
    <className xsi:type="xsd:string">AdvertisedRoutesForCeIPV6</className>
    <properties xsi:type="ns1:CIMPropertyList"
        soapenc:arrayType="ns1:CIMProperty[]">
        <item xsi:type="ns1:CIMProperty">
            <name xsi:type="xsd:string">Advr_Routes_IPV6_Address</name>
            <value xsi:type="xsd:string">29::24/128</value>
        </item>
        <item xsi:type="ns1:CIMProperty">
            <name xsi:type="xsd:string">Advr_Routes_Metric_IPV6</name>
            <value xsi:type="xsd:string">40</value>
        </item>
    </properties>
</objectPath>
<!-- CERC's -->

<objectPath xsi:type="ns1:CIMObjectPath">
    <className xsi:type="xsd:string">CERCMembership</className>
    <properties xsi:type="ns1:CIMPropertyList"
        soapenc:arrayType="ns1:CIMProperty[]">
        <item xsi:type="ns1:CIMProperty">
            <name xsi:type="xsd:string">CERC</name>
            <value xsi:type="xsd:string">cerc4</value>
        </item>
    </properties>
</objectPath>

```

```

        <item xsi:type="ns1:CIMProperty">
            <name xsi:type="xsd:string">IsHub</name>
            <value xsi:type="xsd:string">>true</value>
        </item>
    </properties>
</objectPath>
</objectPath>
</objectPath>
</objectPath>
</action>
</actions>
</ns1:performBatchOperation>
</soapenv:Body>
</soapenv:Envelope>

```

Create BGP Max Prefix with Second VLAN ID and BGP DIO

This example shows how to create a service request for BGP max prefix, BGP Default Info Originate and second VLAN ID with a PE no CE case (IPv4 and IPv6).

The significant attributes for this specific example are highlighted in bold>.

Example:

CreateMPLSServiceOrder_BGP_MaxPrefix_BGP_DIO_SecondVlanId_PE_NO_CE_IPV4_IPV6.xml

```

<!-- Example for BPG MaxPrefix Attributes on IPv4/IPv6,BGP Default Info Originate for
IPv4/IPv6 and DOT1Q on second VLAN ID -->
<?xml version="1.0" encoding="UTF-8"?>
<soapenv:Envelope
  xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
  xmlns:soapenc="http://schemas.xmlsoap.org/soap/encoding/"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xmlns:ns0="http://www.cisco.com/cim-cx/2.0"
  xmlns:ns1="urn:CIM">
  <soapenv:Header>
    <ns0:message id="87855" timestamp="2002-12-13T14:55:38.885Z" Wait="true"
WaitTimeout="90" sessiontoken="p36bttjwy1"/>
  </soapenv:Header>
  <soapenv:Body>
    <ns1:performBatchOperation>
      <actions xsi:type="ns1:CIMActionList" soapenc:arrayType="ns1:CIMAction[]">
        <action>
          <actionName xsi:type="xsd:string">createInstance</actionName>
          <objectPath xsi:type="ns1:CIMObjectPath">
            <className xsi:type="xsd:string">ServiceOrder</className>
            <properties xsi:type="ns1:CIMPropertyList"
soapenc:arrayType="ns1:CIMProperty[]">
              <item xsi:type="ns1:CIMProperty">
                <name xsi:type="xsd:string">ServiceName</name>
                <value xsi:type="xsd:string">ServiceOrder253</value>
              </item>
              <item xsi:type="ns1:CIMProperty">
                <name xsi:type="xsd:string">CarrierId</name>
                <value xsi:type="xsd:string">322</value>
              </item>
              <item xsi:type="ns1:CIMProperty">
                <name xsi:type="xsd:string">DesiredDueDate</name>
                <value
xsi:type="xsd:dateTime">2002-12-13T14:55:38.000Z</value>
              </item>
              <item xsi:type="ns1:CIMProperty">

```

```

        <name xsi:type="xsd:string">NumberOfRequests</name>
        <value xsi:type="xsd:string">1</value>
    </item>
    <!--
    <item xsi:type="ns1:CIMProperty">
        <name xsi:type="xsd:string">Organization</name>
        <value xsi:type="xsd:string">NbiCustomer</value>
    </item>
    -->
</properties>
</objectPath>
</action>

<action>
<actionName xsi:type="xsd:string">createInstance</actionName>
<objectPath xsi:type="ns1:CIMObjectPath">
    <className xsi:type="xsd:string">ServiceRequest</className>
    <properties xsi:type="ns1:CIMPropertyList"
soapenc:arrayType="ns1:CIMProperty[]">
        <item xsi:type="ns1:CIMProperty">
            <name xsi:type="xsd:string">RequestName</name>
            <value xsi:type="xsd:string">MYSR-1</value>
        </item>
        <item xsi:type="ns1:CIMProperty">
            <name xsi:type="xsd:string">Type</name>
            <value xsi:type="xsd:string">Mpls</value>
        </item>
    </properties>
    <objectPath xsi:type="ns1:CIMObjectPath">
    <className xsi:type="xsd:string">ServiceRequestDetails</className>
    <properties xsi:type="ns1:CIMPropertyList"
soapenc:arrayType="ns1:CIMProperty[]">
    <!-- Policy Info -->
        <item xsi:type="ns1:CIMProperty">
            <name xsi:type="xsd:string">ServiceDefinition</name>
            <value
xsi:type="xsd:string">PE_NoCE_BGPMaxPrefix_Dual</value>
            <qualifier xsi:type="xsd:string">
                <name xsi:type="xsd:string">ServiceDefinitionType</name>
                <value xsi:type="xsd:string">Mpls</value>
            </qualifier>
        </item>
    </properties>

    <!-- ##### ( MPLS VPN LINK Start ) #####
-->
    <objectPath xsi:type="ns1:CIMObjectPath">
    <className xsi:type="xsd:string">MplsVpnLink</className>
    <properties xsi:type="ns1:CIMPropertyList"
soapenc:arrayType="ns1:CIMProperty[]">
    <!-- Device Info -->
        <item xsi:type="ns1:CIMProperty">
            <name xsi:type="xsd:string">ManualConfig</name>
            <value xsi:type="xsd:string">true</value>
        </item>
        <item xsi:type="ns1:CIMProperty">
            <name xsi:type="xsd:string">PE</name>
            <value xsi:type="xsd:string">enpe-25</value>
        </item>

    </properties>

    <!-- ##### ( LINKATTRS Start ) #####
-->

```

```

        <objectPath xsi:type="ns1:CIMObjectPath">
          <className xsi:type="xsd:string">LinkAttrs</className>
          <properties xsi:type="ns1:CIMPropertyList"
soapenc:arrayType="ns1:CIMProperty[]">
<!-- PE Intf info -->
          <item xsi:type="ns1:CIMProperty">
            <name xsi:type="xsd:string">PE_Intf_Name</name>
            <value
xsi:type="xsd:string">GigabitEthernet0/3/0/1</value>
            </item>

            <item xsi:type="ns1:CIMProperty">
              <name xsi:type="xsd:string">PE_Vlan_ID</name>
              <value xsi:type="xsd:string">457</value>
            </item>

            <item xsi:type="ns1:CIMProperty">
              <name xsi:type="xsd:string">Second_PE_Vlan_ID</name>
              <value xsi:type="xsd:string">63</value>
            </item>

          </item>

          <item xsi:type="ns1:CIMProperty">
            <name
xsi:type="xsd:string">Autopick_Vlan_ID</name>
            <value xsi:type="xsd:string">>false</value>
          </item>

<!-- PE interface Addresses for IPV4 -->

          <item xsi:type="ns1:CIMProperty">
            <name
xsi:type="xsd:string">PE_Intf_Address</name>
            <value
xsi:type="xsd:string">29.71.2.5/24</value>
          </item>
          <item xsi:type="ns1:CIMProperty">
            <name
xsi:type="xsd:string">Auto_Assign_IP_Address</name>
            <value xsi:type="xsd:string">>false</value>
          </item>

<!-- PE interface Addresses for IPV6 -->

          <item xsi:type="ns1:CIMProperty">
            <name
xsi:type="xsd:string">PE_Intf_Address_IPV6</name>
            <value xsi:type="xsd:string">27::19/25</value>
          </item>

<!-- PE IPV4 Routing Protocol -->

          <item xsi:type="ns1:CIMProperty">
            <name xsi:type="xsd:string">PE_CE_Routing_Protocol</name>
            <value xsi:type="xsd:string">BGP</value>
          </item>
          <item xsi:type="ns1:CIMProperty">
            <name xsi:type="xsd:string">CE_BGP_AS_ID</name>
            <value xsi:type="xsd:string">100</value>
          </item>

          <item xsi:type="ns1:CIMProperty">
            <name xsi:type="xsd:string">BGP_NBR_IP_ADDR</name>
            <value xsi:type="xsd:string">1.3.4.6</value>
          </item>

```

```

        <item xsi:type="ns1:CIMProperty">
            <name xsi:type="xsd:string">PE_CE_NBR_DEFAULT_INFO_ORIGINATE</name>
            <value xsi:type="xsd:string">Enable</value>
        </item>
    <item xsi:type="ns1:CIMProperty">
        <name
xsi:type="xsd:string">PE_CE_NBR_DEFAULT_INFO_ORIGINATE_ROUTE_POLICY</name>
        <value xsi:type="xsd:string">cisco</value>
    </item>
    <item xsi:type="ns1:CIMProperty">
        <name xsi:type="xsd:string">PE_CE_BGP_ADVERTISE_INTERVAL</name>
        <value xsi:type="xsd:string">6</value>
    </item>
    <item xsi:type="ns1:CIMProperty">
        <name xsi:type="xsd:string">PE_CE_BGP_MAX_PREFIX_NUMBER</name>
        <value xsi:type="xsd:string">10</value>
    </item>
    <item xsi:type="ns1:CIMProperty">
        <name xsi:type="xsd:string">PE_CE_BGP_MAX_PREFIX_THRESHOLD</name>
        <value xsi:type="xsd:string">55</value>
    </item>
    <item xsi:type="ns1:CIMProperty">
        <name xsi:type="xsd:string">PE_CE_BGP_MAX_PREFIX_RESTART</name>
        <value xsi:type="xsd:string">451</value>
    </item>
    <item xsi:type="ns1:CIMProperty">
        <name xsi:type="xsd:string">PE_CE_BGP_MAX_PREFIX_WARNING_ONLY</name>
        <value xsi:type="xsd:string">>true</value>
    </item>

<!-- PE IPV6 Routing Protocol -->

    <item xsi:type="ns1:CIMProperty">
        <name xsi:type="xsd:string">PE_CE_IPV6_Routing_Protocol</name>
        <value xsi:type="xsd:string">BGP</value>
    </item>
    <item xsi:type="ns1:CIMProperty">
        <name xsi:type="xsd:string">CE_BGP_AS_ID_IPV6</name>
        <value xsi:type="xsd:string">100</value>
    </item>
    <item xsi:type="ns1:CIMProperty">
        <name xsi:type="xsd:string">BGP_NBR_IPV6_ADDR</name>
        <value xsi:type="xsd:string">1::32</value>
    </item>

    <item xsi:type="ns1:CIMProperty">
        <name xsi:type="xsd:string">PE_CE_NBR_DEFAULT_INFO_ORIGINATE_IPV6</name>
        <value xsi:type="xsd:string">Enable</value>
    </item>

    <item xsi:type="ns1:CIMProperty">
        <name
xsi:type="xsd:string">PE_CE_NBR_DEFAULT_INFO_ORIGINATE_ROUTE_POLICY_IPV6</name>
        <value xsi:type="xsd:string">cisco</value>
    </item>

    <item xsi:type="ns1:CIMProperty">
        <name xsi:type="xsd:string">PE_CE_BGP_ADVERTISE_INTERVAL_IPV6</name>
        <value xsi:type="xsd:string">5</value>
    </item>
    <item xsi:type="ns1:CIMProperty">
        <name xsi:type="xsd:string">PE_CE_BGP_MAX_PREFIX_NUMBER_IPV6</name>
        <value xsi:type="xsd:string">18</value>
    </item>

```

```

        <item xsi:type="ns1:CIMProperty">
          <name
xsi:type="xsd:string">PE_CE_BGP_MAX_PREFIX_THRESHOLD_IPV6</name>
          <value xsi:type="xsd:string">84</value>
        </item>
        <item xsi:type="ns1:CIMProperty">
          <name xsi:type="xsd:string">PE_CE_BGP_MAX_PREFIX_RESTART_IPV6</name>
          <value xsi:type="xsd:string">35</value>
        </item>
        <item xsi:type="ns1:CIMProperty">
          <name
xsi:type="xsd:string">PE_CE_BGP_MAX_PREFIX_WARNING_ONLY_IPV6</name>
          <value xsi:type="xsd:string">>true</value>
        </item>

<!-- CERC Type -->
                                <!--
                                <item xsi:type="ns1:CIMProperty">
                                  <name xsi:type="xsd:string">CERC_RTTYPE</name>
                                  <value
xsi:type="xsd:string">cerc005=dual,cerc006=IPv6</value>
                                  </item>
                                -->

<!-- BGP Route Map In / Out -->
                                <!--
                                <item xsi:type="ns1:CIMProperty">
                                  <name
xsi:type="xsd:string">PE_CE_NBR_ROUTE_MAP_IN_NAME</name>
                                  <value xsi:type="xsd:string">verve2</value>
                                </item>
                                <item xsi:type="ns1:CIMProperty">
                                  <name
xsi:type="xsd:string">PE_CE_NBR_ROUTE_MAP_OUT_NAME</name>
                                  <value xsi:type="xsd:string">verve2</value>
                                </item>
                                -->

<!-- Enable Multipath Loadsharing -->
                                <!--
                                <item xsi:type="ns1:CIMProperty">
                                  <name
xsi:type="xsd:string">Bgp_Multipath_Enabled</name>
                                  <value xsi:type="xsd:string">>true</value>
                                </item>
                                -->

<!-- Enable RD Format -->
                                <!--
                                <item xsi:type="ns1:CIMProperty">
                                  <name xsi:type="xsd:string">RD_FORMAT</name>
                                  <value xsi:type="xsd:string">RD_AS</value>
                                </item>

                                <item xsi:type="ns1:CIMProperty">
                                  <name xsi:type="xsd:string">RD_FORMAT</name>
                                  <value xsi:type="xsd:string">RD_IPADDR</value>
                                </item>

                                <item xsi:type="ns1:CIMProperty">
                                  <name xsi:type="xsd:string">RD_IPADDRESS</name>
                                  <value
xsi:type="xsd:string">11.21.30.24</value>
                                </item>

```



```

        <item xsi:type="ns1:CIMProperty">
            <name xsi:type="xsd:string">Enable_CSC</name>
            <value xsi:type="xsd:string">>false</value>
        </item>
        <item xsi:type="ns1:CIMProperty">
            <name
xsi:type="xsd:string">MulticastEnabled</name>
            <value xsi:type="xsd:string">>false</value>
        </item>
        <item xsi:type="ns1:CIMProperty">
            <name
xsi:type="xsd:string">Vrf_Rd_Overwrite_Enabled</name>
            <value xsi:type="xsd:string">>true</value>
        </item>
        <item xsi:type="ns1:CIMProperty">
            <name
xsi:type="xsd:string">Overridden_Vrf_Name</name>
            <value xsi:type="xsd:string">test3</value>
        </item>
        <item xsi:type="ns1:CIMProperty">
            <name xsi:type="xsd:string">Overridden_Rd</name>
            <value xsi:type="xsd:string">158:143</value>
        </item>
        -->
    </properties>
</objectPath>
<!-- ##### ( LINKATTRS End ) ##### -->

<!-- CERC's -->

    <objectPath xsi:type="ns1:CIMObjectPath">
        <className xsi:type="xsd:string">CERCMembership</className>
        <properties xsi:type="ns1:CIMPropertyList"
soapenc:arrayType="ns1:CIMProperty[]">
            <item xsi:type="ns1:CIMProperty">
                <name xsi:type="xsd:string">CERC</name>
                <value xsi:type="xsd:string">27Mycerc1</value>
            </item>
            <item xsi:type="ns1:CIMProperty">
                <name xsi:type="xsd:string">IsHub</name>
                <value xsi:type="xsd:string">>true</value>
            </item>
        </properties>
    </objectPath>

<!-- Template Info -->
    <!--
        <objectPath xsi:type="ns1:CIMObjectPath">
            <className xsi:type="xsd:string">LinkTemplate</className>
            <properties xsi:type="ns1:CIMPropertyList"
soapenc:arrayType="ns1:CIMProperty[]">
                <item xsi:type="ns1:CIMProperty">
                    <name xsi:type="xsd:string">LogicalDevice</name>
                    <value xsi:type="xsd:string">CRS</value>
                </item>
                <item xsi:type="ns1:CIMProperty">
                    <name xsi:type="xsd:string">DatafilePath</name>
                    <value xsi:type="xsd:string">/mpls/sample3</value>
                </item>
                <item xsi:type="ns1:CIMProperty">
                    <name xsi:type="xsd:string">DatafileName</name>
                    <value xsi:type="xsd:string">data3</value>
                </item>
            </properties>
        </objectPath>
    </!--

```

```

        <item xsi:type="ns1:CIMProperty">
          <name xsi:type="xsd:string">TemplateActive</name>
          <value xsi:type="xsd:string">true</value>
        </item>
        <item xsi:type="ns1:CIMProperty">
          <name xsi:type="xsd:string">TemplateAction</name>
          <value xsi:type="xsd:string">APPEND</value>
        </item>
      </properties>

    </objectPath>
  -->
      <!-- End of Template here -->

    </objectPath>
    <!-- ##### ( MPLS VPN LINK End
##### -->

    <!-- ##### ( If you need Add one more link here
##### -->

    </objectPath>
  </objectPath>
</action>
</actions>
</ns1:performBatchOperation>
</soapenv:Body>
</soapenv:Envelope>

```

Create EIGRP Key Chain

This example shows how to create a service request for the EIGRP key chain feature with a PE and a CE (IPv4 or IPv6).

Example: CreateMPLSServiceOrder_EIGRP_KeyChain_PE_CE_IPV4_IPV6.xml

```

<!-- Example for EIGRP Key Chain Feature for IPv4/IPv6 -->
<?xml version="1.0" encoding="UTF-8"?>
<soapenv:Envelope
  xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
  xmlns:soapenc="http://schemas.xmlsoap.org/soap/encoding/"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xmlns:ns0="http://www.cisco.com/cim-cx/2.0"
  xmlns:ns1="urn:CIM">
  <soapenv:Header>
    <!-- WaitTimeout has a default set in system properties.-->
    <ns0:message id="87855" timestamp="2002-12-13T14:55:38.885Z"
      Wait="true" WaitTimeout="90" sessiontoken="p36bttjwy1"/>
  </soapenv:Header>
  <soapenv:Body>
    <ns1:performBatchOperation>
      <actions xsi:type="ns1:CIMActionList"
        soapenc:arrayType="ns1:CIMAction[]">
        <action>
          <actionName xsi:type="xsd:string">createInstance</actionName>
          <objectPath xsi:type="ns1:CIMObjectPath">
            <className xsi:type="xsd:string">ServiceOrder</className>
            <properties xsi:type="ns1:CIMPropertyList"

```

```

        soapenc:arrayType="ns1:CIMProperty[]">
<item xsi:type="ns1:CIMProperty">
  <name xsi:type="xsd:string">ServiceName</name>
  <value xsi:type="xsd:string">ServiceOrder253</value>
</item>
<item xsi:type="ns1:CIMProperty">
  <name xsi:type="xsd:string">CarrierId</name>
  <value xsi:type="xsd:string">322</value>
</item>
<item xsi:type="ns1:CIMProperty">
  <name xsi:type="xsd:string">DesiredDueDate</name>
  <value xsi:type="xsd:dateTime">2002-12-13T14:55:38.000Z</value>
</item>
<item xsi:type="ns1:CIMProperty">
  <name xsi:type="xsd:string">NumberOfRequests</name>
  <value xsi:type="xsd:string">1</value>
</item>
<!--item xsi:type="ns1:CIMProperty">
  <name xsi:type="xsd:string">Organization</name>
  <value xsi:type="xsd:string">NbiCustomer</value>
</item-->
</properties>
</objectPath>
</action>
<action>
  <actionName xsi:type="xsd:string">createInstance</actionName>
  <objectPath xsi:type="ns1:CIMObjectPath">
    <className xsi:type="xsd:string">ServiceRequest</className>
    <properties xsi:type="ns1:CIMPropertyList"
      soapenc:arrayType="ns1:CIMProperty[]">
      <item xsi:type="ns1:CIMProperty">
        <name xsi:type="xsd:string">RequestName</name>
        <value xsi:type="xsd:string">MYSR-1</value>
      </item>
      <item xsi:type="ns1:CIMProperty">
        <name xsi:type="xsd:string">Type</name>
        <value xsi:type="xsd:string">Mpls</value>
      </item>
    </properties>
    <objectPath xsi:type="ns1:CIMObjectPath">
      <className xsi:type="xsd:string">ServiceRequestDetails</className>
      <properties xsi:type="ns1:CIMPropertyList"
        soapenc:arrayType="ns1:CIMProperty[]">
      <item xsi:type="ns1:CIMProperty">
        <name xsi:type="xsd:string">ServiceDefinition</name>
        <value xsi:type="xsd:string">PE_CE_IPV4_NBI</value>
        <qualifier xsi:type="xsd:string">
          <name xsi:type="xsd:string">ServiceDefinitionType</name>
          <value xsi:type="xsd:string">Mpls</value>
        </qualifier>
      </item>
    </properties>
    <objectPath xsi:type="ns1:CIMObjectPath">
      <className xsi:type="xsd:string">MplsVpnLink</className>
      <properties xsi:type="ns1:CIMPropertyList"
        soapenc:arrayType="ns1:CIMProperty[]">
      <!-- ManualConfig is needed when specifying PE-CE link and not using NPC -->
      <item xsi:type="ns1:CIMProperty">
        <name xsi:type="xsd:string">ManualConfig</name>
        <value xsi:type="xsd:string">true</value>
      </item>
      <item xsi:type="ns1:CIMProperty">

```

```

        <name xsi:type="xsd:string">PE</name>
        <value xsi:type="xsd:string">CRS</value>
    </item>
    <item xsi:type="ns1:CIMProperty">
        <name xsi:type="xsd:string">Cpe</name>
        <value xsi:type="xsd:string">iscind-7600-2</value>
    </item>
</properties>
<objectPath xsi:type="ns1:CIMObjectPath">
    <className xsi:type="xsd:string">LinkAttrs</className>
    <properties xsi:type="ns1:CIMPropertyList"
soapenc:arrayType="ns1:CIMProperty[]">

<!-- PE Intf info -->
    <item xsi:type="ns1:CIMProperty">
        <name xsi:type="xsd:string">PE_Intf_Name</name>
        <value xsi:type="xsd:string">GigabitEthernet0/15/1/4</value>
    </item>

    <!-- <item xsi:type="ns1:CIMProperty">
        <name xsi:type="xsd:string">PE_Intf_Shutdown</name>
        <value xsi:type="xsd:string">true</value>
    </item>

    <item xsi:type="ns1:CIMProperty">
        <name xsi:type="xsd:string">PE_Intf_Encap</name>
        <value xsi:type="xsd:string">DOT1Q</value>
    </item> -->

    <item xsi:type="ns1:CIMProperty">
        <name xsi:type="xsd:string">PE_Vlan_ID</name>
        <value xsi:type="xsd:string">700</value>
    </item>

    <item xsi:type="ns1:CIMProperty">
        <name xsi:type="xsd:string">Autopick_Vlan_ID</name>
        <value xsi:type="xsd:string">>false</value>
    </item>
<!-- CE Intf Info -->
    <item xsi:type="ns1:CIMProperty">
        <name xsi:type="xsd:string">CE_Intf_Name</name>
        <value xsi:type="xsd:string">FastEthernet2/9</value>
    </item>
    <!-- <item xsi:type="ns1:CIMProperty">
        <name xsi:type="xsd:string">CE_Intf_Encap</name>
        <value xsi:type="xsd:string">DOT1Q</value>
    </item> -->
    <item xsi:type="ns1:CIMProperty">
        <name xsi:type="xsd:string">CE_Vlan_ID</name>
        <value xsi:type="xsd:string">702</value>
    </item>
<!-- interface Addresses for IPV4 -->
    <item xsi:type="ns1:CIMProperty">
        <name xsi:type="xsd:string">CE_Intf_Address</name>
        <value xsi:type="xsd:string">15.16.17.2/24</value>
    </item>
    <item xsi:type="ns1:CIMProperty">
        <name xsi:type="xsd:string">PE_Intf_Address</name>
        <value xsi:type="xsd:string">15.16.17.3/24</value>
    </item>
    <item xsi:type="ns1:CIMProperty">
        <name xsi:type="xsd:string">Auto_Assign_IP_Address</name>

```

```

        <value xsi:type="xsd:string">>false</value>
    </item>
<!-- interface Addresses for IPV6 -->

    <item xsi:type="ns1:CIMProperty">
        <name xsi:type="xsd:string">PE_Intf_Address_IPV6</name>
        <value xsi:type="xsd:string">2010::16/28</value>
    </item>
    <item xsi:type="ns1:CIMProperty">
        <name xsi:type="xsd:string">CE_Intf_Address_IPV6</name>
        <value xsi:type="xsd:string">2010::17/16</value>
    </item>

<!-- IPV4 Routing Protocol -->
    <item xsi:type="ns1:CIMProperty">
        <name xsi:type="xsd:string">PE_CE_Routing_Protocol</name>
        <value xsi:type="xsd:string">EIGRP</value>
    </item>
    <item xsi:type="ns1:CIMProperty">
        <name
xsi:type="xsd:string">PE_CE_EIGRP_AUTHENTICATION_KEY_CHAIN_NAME</name>
        <value xsi:type="xsd:string">nbi-key-v4</value>
    </item>
    <item xsi:type="ns1:CIMProperty">
        <name xsi:type="xsd:string">CE_EIGRP_AS_ID</name>
        <value xsi:type="xsd:string">2136</value>
    </item>
    <item xsi:type="ns1:CIMProperty">
        <name xsi:type="xsd:string">PE_EIGRP_AS_ID</name>
        <value xsi:type="xsd:string">100</value>
    </item>
    <item xsi:type="ns1:CIMProperty">
        <name
xsi:type="xsd:string">PE_CE_Bandwidth_Metric_For_Redistribution</name>
        <value xsi:type="xsd:string">1256</value>
    </item>
    <item xsi:type="ns1:CIMProperty">
        <name
xsi:type="xsd:string">PE_CE_Delay_Metric_For_Redistribution</name>
        <value xsi:type="xsd:string">4521</value>
    </item>
    <item xsi:type="ns1:CIMProperty">
        <name
xsi:type="xsd:string">PE_CE_Reliability_Metric_For_Redistribution</name>
        <value xsi:type="xsd:string">163</value>
    </item>
    <item xsi:type="ns1:CIMProperty">
        <name
xsi:type="xsd:string">PE_CE>Loading_Metric_For_Redistribution</name>
        <value xsi:type="xsd:string">145</value>
    </item>
    <item xsi:type="ns1:CIMProperty">
        <name xsi:type="xsd:string">PE_CE_MTU_Metric_For_Redistribution</name>
        <value xsi:type="xsd:string">2136</value>
    </item>

<!-- IPV6 Routing Protocol -->

    <item xsi:type="ns1:CIMProperty">
        <name xsi:type="xsd:string">PE_CE_IPV6_Routing_Protocol</name>
        <value xsi:type="xsd:string">EIGRP</value>
    </item>
    <item xsi:type="ns1:CIMProperty">

```

```

        <name
xsi:type="xsd:string">PE_CE_EIGRP_AUTHENTICATION_KEY_CHAIN_NAME_IPV6</name>
        <value xsi:type="xsd:string">nbi-key-v6</value>
    </item>
    <item xsi:type="ns1:CIMProperty">
        <name xsi:type="xsd:string">CE_EIGRP_AS_ID_IPV6</name>
        <value xsi:type="xsd:string">2136</value>
    </item>
        <item xsi:type="ns1:CIMProperty">
            <name xsi:type="xsd:string">PE_EIGRP_AS_ID_IPV6</name>
            <value xsi:type="xsd:string">100</value>
        </item>
        <item xsi:type="ns1:CIMProperty">
            <name
xsi:type="xsd:string">PE_CE_Bandwidth_Metric_For_Redistribution_IPV6</name>
            <value xsi:type="xsd:string">1256</value>
        </item>
        <item xsi:type="ns1:CIMProperty">
            <name
xsi:type="xsd:string">PE_CE_Delay_Metric_For_Redistribution_IPV6</name>
            <value xsi:type="xsd:string">4521</value>
        </item>
        <item xsi:type="ns1:CIMProperty">
            <name
xsi:type="xsd:string">PE_CE_Reliability_Metric_For_Redistribution_IPV6</name>
            <value xsi:type="xsd:string">163</value>
        </item>
        <item xsi:type="ns1:CIMProperty">
            <name
xsi:type="xsd:string">PE_CE>Loading_Metric_For_Redistribution_IPV6</name>
            <value xsi:type="xsd:string">145</value>
        </item>
        <item xsi:type="ns1:CIMProperty">
            <name
xsi:type="xsd:string">PE_CE_MTU_Metric_For_Redistribution_IPV6</name>
            <value xsi:type="xsd:string">2136</value>
        </item>

    <!-- Vrf-RD- Overridden-->

        <!-- <item xsi:type="ns1:CIMProperty">
            <name xsi:type="xsd:string">Vrf_Rd_Over_Write_Enabled</name>
            <value xsi:type="xsd:string">>true</value>
        </item>
    <item xsi:type="ns1:CIMProperty">
        <name xsi:type="xsd:string">Overridden_Vrf_Name</name>
        <value xsi:type="xsd:string">vrf1</value>
    </item>
    <item xsi:type="ns1:CIMProperty">
        <name xsi:type="xsd:string">Overridden_Rd</name>
        <value xsi:type="xsd:string">100:45</value>
    </item> -->

    </properties>
</objectPath>

<!-- CERC's -->

<objectPath xsi:type="ns1:CIMObjectPath">
    <className xsi:type="xsd:string">CERCMembership</className>
    <properties xsi:type="ns1:CIMPropertyList"
        soapenc:arrayType="ns1:CIMProperty[]">

```

```

        <item xsi:type="ns1:CIMProperty">
          <name xsi:type="xsd:string">CERC</name>
          <value xsi:type="xsd:string">cercl</value>
        </item>
        <item xsi:type="ns1:CIMProperty">
          <name xsi:type="xsd:string">IsHub</name>
          <value xsi:type="xsd:string">>true</value>
        </item>
      </properties>
    </objectPath>

<!-- LinkTemplate example using template definition and direct input of template data
(buffer method). The template data
        is entered in the DataBuffer objectDef section -->
    <!--
      <objectPath xsi:type="ns1:CIMObjectPath">
        <className xsi:type="xsd:string">LinkTemplate</className>
        <properties xsi:type="ns1:CIMPropertyList"
soapenc:arrayType="ns1:CIMProperty[]">
          <item xsi:type="ns1:CIMProperty">
            <name xsi:type="xsd:string">LogicalDevice</name>
            <value xsi:type="xsd:string">mlpe7</value>
          </item>
          <item xsi:type="ns1:CIMProperty">
            <name xsi:type="xsd:string">DatafilePath</name>
            <value xsi:type="xsd:string">/sampl8/data23</value>
          </item>
          <item xsi:type="ns1:CIMProperty">
            <name xsi:type="xsd:string">TemplateActive</name>
            <value xsi:type="xsd:string">>true</value>
          </item>
          <item xsi:type="ns1:CIMProperty">
            <name xsi:type="xsd:string">TemplateAction</name>
            <value xsi:type="xsd:string">APPEND</value>
          </item>
        </properties>
      <objectPath xsi:type="ns1:CIMObjectPath">
        <className xsi:type="xsd:string">DataBuffer</className>
        <properties xsi:type="ns1:CIMPropertyList"
          soapenc:arrayType="ns1:CIMProperty[]">
          <item xsi:type="ns1:CIMProperty">
            <name xsi:type="xsd:string">protocol</name>
            <value xsi:type="xsd:string">ip</value>
          </item>
          <item xsi:type="ns1:CIMProperty">
            <name xsi:type="xsd:string">Application</name>
            <value xsi:type="xsd:string">bootps</value>
          </item>
        </properties>
      </objectPath>
    </objectPath>
  </objectPath>
  </action>
</actions>
</ns1:performBatchOperation>
</soapenv:Body>
</soapenv:Envelope>

```

Create OSPF Default Info Originate

This example shows how to create a service request for generating a default route into an OSPF routing domain with a PE and no CE (IPv4 only).

Example: CreateMPLSServiceOrder_OSPF_DefaultInfoOriginate_PE_NO_CE_IPV4.xml

```
<?xml version="1.0" encoding="UTF-8"?>
<soapenv:Envelope
  xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
  xmlns:soapenc="http://schemas.xmlsoap.org/soap/encoding/"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xmlns:ns0="http://www.cisco.com/cim-cx/2.0"
  xmlns:ns1="urn:CIM">
  <soapenv:Header>
    <!-- WaitTimeout has a default set in system properties.-->
    <ns0:message id="87855" timestamp="2002-12-13T14:55:38.885Z"
      Wait="true" WaitTimeout="90" sessiontoken="p36bttjwy1"/>
  </soapenv:Header>
  <soapenv:Body>
    <ns1:performBatchOperation>
      <actions xsi:type="ns1:CIMActionList"
        soapenc:arrayType="ns1:CIMAction[]">
        <action>
          <actionName xsi:type="xsd:string">createInstance</actionName>
          <objectPath xsi:type="ns1:CIMObjectPath">
            <className xsi:type="xsd:string">ServiceOrder</className>
            <properties xsi:type="ns1:CIMPropertyList"
              soapenc:arrayType="ns1:CIMProperty[]">
              <item xsi:type="ns1:CIMProperty">
                <name xsi:type="xsd:string">ServiceName</name>
                <value xsi:type="xsd:string">ServiceOrder253</value>
              </item>
              <item xsi:type="ns1:CIMProperty">
                <name xsi:type="xsd:string">CarrierId</name>
                <value xsi:type="xsd:string">322</value>
              </item>
              <item xsi:type="ns1:CIMProperty">
                <name xsi:type="xsd:string">DesiredDueDate</name>
                <value xsi:type="xsd:dateTime">2002-12-13T14:55:38.000Z</value>
              </item>
              <item xsi:type="ns1:CIMProperty">
                <name xsi:type="xsd:string">NumberOfRequests</name>
                <value xsi:type="xsd:string">1</value>
              </item>
              <!--item xsi:type="ns1:CIMProperty">
                <name xsi:type="xsd:string">Organization</name>
                <value xsi:type="xsd:string">NbiCustomer</value>
              </item-->
            </properties>
          </objectPath>
        </action>
        <action>
          <actionName xsi:type="xsd:string">createInstance</actionName>
          <objectPath xsi:type="ns1:CIMObjectPath">
            <className xsi:type="xsd:string">ServiceRequest</className>
            <properties xsi:type="ns1:CIMPropertyList"
              soapenc:arrayType="ns1:CIMProperty[]">
              <item xsi:type="ns1:CIMProperty">
                <name xsi:type="xsd:string">RequestName</name>
                <value xsi:type="xsd:string">MYSR-1</value>
              </item>
            </properties>
          </objectPath>
        </action>
      </actions>
    </ns1:performBatchOperation>
  </soapenv:Body>
</soapenv:Envelope>
```



```

        <item xsi:type="ns1:CIMProperty">
          <name xsi:type="xsd:string">Type</name>
          <value xsi:type="xsd:string">Mpls</value>
        </item>
      </properties>
    <objectPath xsi:type="ns1:CIMObjectPath">
      <className xsi:type="xsd:string">ServiceRequestDetails</className>
      <properties xsi:type="ns1:CIMPropertyList"
soapenc:arrayType="ns1:CIMProperty[]" >

<!-- Policy Name -->
    <item xsi:type="ns1:CIMProperty">
      <name xsi:type="xsd:string">ServiceDefinition</name>
      <value xsi:type="xsd:string">PE_NO_CE</value>
      <qualifier xsi:type="xsd:string">
        <name xsi:type="xsd:string">ServiceDefinitionType</name>
        <value xsi:type="xsd:string">Mpls</value>
      </qualifier>
    </item>
  </properties>
  <objectPath xsi:type="ns1:CIMObjectPath">
    <className xsi:type="xsd:string">MplsVpnLink</className>
    <properties xsi:type="ns1:CIMPropertyList"
      soapenc:arrayType="ns1:CIMProperty[]" >

<!-- ManualConfig is needed when specifying PE-CE link and not using NPC -->

<!--Device Info -->
    <item xsi:type="ns1:CIMProperty">
      <name xsi:type="xsd:string">ManualConfig</name>
      <value xsi:type="xsd:string">>true</value>
    </item>
    <item xsi:type="ns1:CIMProperty">
      <name xsi:type="xsd:string">PE</name>
      <value xsi:type="xsd:string">Cisco CRS</value>
    </item>

  </properties>
  <objectPath xsi:type="ns1:CIMObjectPath">
    <className xsi:type="xsd:string">LinkAttrs</className>
    <properties xsi:type="ns1:CIMPropertyList"
soapenc:arrayType="ns1:CIMProperty[]" >

<!-- PE Intf Info -->
    <item xsi:type="ns1:CIMProperty">
      <name xsi:type="xsd:string">PE_Intf_Name</name>
      <value xsi:type="xsd:string">TenGigE0/0/0/2</value>
    </item>
    <item xsi:type="ns1:CIMProperty">
      <name xsi:type="xsd:string">PE_Vlan_ID</name>
      <value xsi:type="xsd:string">593</value>
    </item>
    <item xsi:type="ns1:CIMProperty">
      <name xsi:type="xsd:string">Autopick_Vlan_ID</name>
      <value xsi:type="xsd:string">>false</value>
    </item>

<!-- interface Addresses for IPV4 -->

    <item xsi:type="ns1:CIMProperty">
      <name xsi:type="xsd:string">PE_Intf_Address</name>
      <value xsi:type="xsd:string">36.25.4.6/24</value>
    </item>

```

```

<!-- IPv4 Routing Protocol -->
    <item xsi:type="ns1:CIMProperty">
        <name xsi:type="xsd:string">PE_CE_Routing_Protocol</name>
        <value xsi:type="xsd:string">OSPF</value>
    </item>
    <item xsi:type="ns1:CIMProperty">
        <name xsi:type="xsd:string">PE_OSPF_PROCESS_ID</name>
        <value xsi:type="xsd:string">123</value>
    </item>
    <item xsi:type="ns1:CIMProperty">
        <name xsi:type="xsd:string">PE_CE_OSPF_DEF_INFO_ORIGINATE</name>
        <value xsi:type="xsd:string">true</value>
    </item>
    <item xsi:type="ns1:CIMProperty">
        <name
xsi:type="xsd:string">PE_CE_OSPF_DEF_INFO_ORIGINATE_ALWAYS</name>
        <value xsi:type="xsd:string">true</value>
    </item>
    <item xsi:type="ns1:CIMProperty">
        <name xsi:type="xsd:string">PE_CE_OSPF_METRIC_VALUE</name>
        <value xsi:type="xsd:string">354</value>
    </item>
    <item xsi:type="ns1:CIMProperty">
        <name xsi:type="xsd:string">PE_CE_OSPF_METRIC_TYPE</name>
        <value xsi:type="xsd:string">None</value>
    </item>
    <item xsi:type="ns1:CIMProperty">
        <name xsi:type="xsd:string">PE_CE_OSPF_ROUTE_POLICY</name>
        <value xsi:type="xsd:string">dinesh</value>
    </item>
    <item xsi:type="ns1:CIMProperty">
        <name xsi:type="xsd:string">PE_CE_Ospf_Area_Number</name>
        <value xsi:type="xsd:string">345</value>
    </item>
</properties>
</objectPath>

<!-- CERC's -->

<objectPath xsi:type="ns1:CIMObjectPath">
    <className xsi:type="xsd:string">CERCMembership</className>
    <properties xsi:type="ns1:CIMPropertyList"
        soapenc:arrayType="ns1:CIMProperty[]">
        <item xsi:type="ns1:CIMProperty">
            <name xsi:type="xsd:string">CERC</name>
            <value xsi:type="xsd:string">Cerc-1-100</value>
        </item>
        <item xsi:type="ns1:CIMProperty">
            <name xsi:type="xsd:string">IsHub</name>
            <value xsi:type="xsd:string">true</value>
        </item>
    </properties>
</objectPath>

<!-- LinkTemplate example using template definition and direct input of template data
(buffer method). The template data
                                is entered in the DataBuffer objectDef section -->

<!-- TEmplate start here-->

    <!-- <objectPath xsi:type="ns1:CIMObjectPath">
        <className xsi:type="xsd:string">LinkTemplate</className>

```

```

        <properties xsi:type="ns1:CIMPropertyList"
soapenc:arrayType="ns1:CIMProperty[]" >
    <item xsi:type="ns1:CIMProperty">
        <name xsi:type="xsd:string">LogicalDevice</name>
        <value xsi:type="xsd:string">CRS</value>
    </item>
    <item xsi:type="ns1:CIMProperty">
        <name xsi:type="xsd:string">DatafilePath</name>
        <value xsi:type="xsd:string">/mpls/sample1</value>
    </item>
    <item xsi:type="ns1:CIMProperty">
        <name xsi:type="xsd:string">DatafileName</name>
        <value xsi:type="xsd:string">d21</value>
    </item>
    <item xsi:type="ns1:CIMProperty">
        <name xsi:type="xsd:string">TemplateActive</name>
        <value xsi:type="xsd:string">>true</value>
    </item>
    <item xsi:type="ns1:CIMProperty">
        <name xsi:type="xsd:string">TemplateAction</name>
        <value xsi:type="xsd:string">APPEND</value>
    </item>
</properties>

</objectPath>-->

<!-- Template Ends Here -->

        </objectPath>
    </objectPath>
</objectPath>
</action>
</actions>
</ns1:performBatchOperation>
</soapenv:Body>
</soapenv:Envelope>

```

Related APIs

- CreateMPLSServiceOrder_PE_CE_IPV4_IPV6_Unique_RT.xml
- CreateMPLSServiceOrder_PE_NO_CE_IPV4_IPV6_Unique_RT.xml
- ModifyMPLSServiceOrder_Unique_RT.xml
- CreateMPLSServiceOrder_Bundle_PE_CE_IPV4_IPV6.xml
- ModifyMPLSServiceOrder_Bundle_PE_CE.xml
- CreateMplsServiceOrder_StaticRoute_PE_CE.xml
- CreateMPLSServiceOrder_StaticRoute_PE_NO_CE.xml

End-to-End Provisioning Process

The following sections describe the required steps for using the API to provision MPLS VPNs, and include the operation, class, and required parameters for each step.

Process Summary

MPLS provisioning using the API includes the following end-to-end operations:

-
- Step 1** Prior to creating the inventory, you need to do the following:
- Send a login XML request to generate a session ID. This is used each time you access the system. (see [Session](#), page 3-7.)
 - View the status of the API by executing


```
runNbi x $PRIMEF_HOME/resources/nbi/xml/examples/Session/Login.xml
```

 If the API is running, you receive a session token. (see [Checking the API Status](#), page 2-1.)
- Step 2** Create Inventory.
- **CreateProvider.xml**—Create a provider.
 - **CreateRegion.xml**—Create a region.
 - **CreateOrganization.xml**—Create customer organization.
 - **CreateSite.xml**—Create customer site.
 - **CreateCiscoRouter.xml**—Create devices (Cisco IOS router).
 - **CreateCat.xml**—Create devices (Catalyst switch).
 - **CreatePE.xml**—Declare a device as **PE** and assign it to regions.
 - **CreateCpe.xml**—Declare a device as **CPE** and assign it to sites.
- For more information, see [Create Inventory](#), page 6-37.
- Step 3** Create Resource Pools.
- Create a **VPN** and select a **CERC**:
- **CreateRouteTarget.xml**—Inform PEs which routes should be inserted into the appropriate VRFs.
 - **CreateRouteDistinguisher.xml**—Help the CE router advertise IP subnets to the PE routers. The RD value must be a globally unique value to avoid conflict with other prefixes.
 - **CreateCERC.xml**—Create a CERC.
 - **CreateVPN.xml**—Create a VPN.
- For more information, see [Create Resource Pools](#), page 6-39.
- Step 4** Collect Device Configurations.
- **CreateTaskServiceOrderCollection.xml**—upload the current configuration from the device to the Prime Fulfillment database.
- For more information, see [Collect Device Configurations](#), page 6-40.
- Step 5** Create an MPLS Policy.
- **CreateMPLSServiceDefn_PE_CE.xml**—Define a policy template for a PE-CE link.
 - **CreateMPLSServiceDefn_PE_NO_CE.xml**—Define a policy template for a PE-no CE link.
 - **CreateMPLSServiceDefn_MVRF_CE.xml**—Define a policy template for a MVRF-CE link.
 - **CreateMPLSServiceDefn_MVRF_NO_CE.xml**—Define a policy template for a MVRF-no CE link.
- For more information, see [Create an MPLS Policy](#), page 6-42.

Step 6 Create an MPLS Service Request.



Note To modify a service request, see [Modifying a Service Request, page 1-12](#). To delete or purge a service request, you need the returned Locator ID. (See [Step 7](#).) For examples of modify and delete operations, go to `examples.tar` under the MPLS directory.

Define which service definition to use, the **MplsVpnLink**, and the link attributes:

- **CreateMPLSServiceOrder_PE_CE.xml**—Create service request using the PE-CE policy.
- **CreateMPLSServiceOrder_PE_NO_CE.xml**—Create service request using the PE-noCE policy.
- **CreateMPLSServiceOrder_MVFR_CE.xml**—Create service request using the MVRF-CE policy.
- **CreateMPLSServiceOrder_MVRF_NO_CE.xml**—Create service request using the MVRF-no CE policy.

For more information, see [Creating an MPLS Service Request, page 6-42](#).

Step 7 When you submit a service order XML request, Prime Fulfillment returns a **LocatorId** in the XML response.

Make a record of the Locator ID or service name for all service orders and service requests. The locator ID is required to view a service order, to perform a service order task (configuration audit or functional audit), and for all subsequent requests related to the service order or service request.

For more information, see [Service Order Response, page 2-12](#).

Step 8 Using the LocatorId, you can view the log output for the created and deployed service request. This includes all messages according to the level specified in the properties file.

For more information, see [Viewing Task Logs, page 5-24](#).

Detailed Process and Attributes

This MPLS provisioning example includes the following operations:

1. [Create Inventory](#)
2. [Create Resource Pools](#)
3. [Collect Device Configurations](#)
4. [Create an MPLS Policy](#)
5. [Creating an MPLS Service Request](#)
6. [Auditing Service Requests, page 6-44](#).

This section provides an example provisioning process using XML examples. The inventory of XML examples for the Prime Fulfillment API is available here: [Cisco Prime Fulfillment API Programmer Reference 6.1](#).

Create Inventory

Step 1 Create a **Provider** and **Region**.

The provider is the administrative domain of an Internet service provider (ISP), with one border gateway protocol (BGP) autonomous system (AS) number. The network owned by the provider is called the backbone network. If an ISP has two AS numbers, you must define it as two provider administrative domains (PADs). Each provider can contain multiple regions.

Table 6-1 Create Provider and Region

Operation	className	Required Keywords
createInstance	Provider	<ul style="list-style-type: none"> Name AsNumber
	Region	<ul style="list-style-type: none"> Name Provider

XML Examples:

- CreateProvider.xml
- CreateRegion.xml

Step 2 Create a Customer (**Organization**) and **Site**.

Table 6-2 Create Customer and Site

Operation	className	Required Keywords
createInstance	Organization	<ul style="list-style-type: none"> Name
	Site	<ul style="list-style-type: none"> Name Organization

XML Examples:

- CreateOrganization.xml
- CreateSite.xml

Step 3 Create devices.

In most cases, devices are Cisco IOS routers and Catalyst switches.

Table 6-3 Create Devices

Operation	className	Required Keywords
createInstance	<ul style="list-style-type: none"> CiscoRouter CatOS 	One or more of the following: <ul style="list-style-type: none"> ManagementIPAddress HostName DomainName

XML Examples:

- CreateCiscoRouter.xml

- CreateCat.xml

Step 4 Declare devices as **PEs** and assign them to regions.

Table 6-4 Create PEs

Operation	className	Required Keywords
createInstance	PE	<ul style="list-style-type: none"> • Provider • Region • Role= <ul style="list-style-type: none"> – N-PE – U-PE – P – PE-AGG • Device • Interface

XML Example:

- CreatePE.xml

Step 5 Declare devices as **Cpes** and assign them to sites.

When you declare a device as a **Cpe**, you also specify a **ManagementType** and interface information.

Table 6-5 Create CPEs

Operation	className	Required Keywords
createInstance	Cpe	<ul style="list-style-type: none"> • Site • Device • ManagementType

XML Example:

- CreateCpe.xml

Create Resource Pools

For MPLS provisioning, you define route targets, route distinguishers, CERCs, and VPNs.

A route target informs PEs 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 IP subnets advertised by the CE routers to the PE routers are augmented with route distinguishers (RDs). The RD value must be a globally unique value to avoid conflict with other prefixes.

Table 6-6 Create Resource Pools

Operation	classNames	Required Keywords
createInstance	<ul style="list-style-type: none"> RouteTarget RouteDistinguisher 	<ul style="list-style-type: none"> Start Size AssocClassType AssocClassId

Create a **VPN** and select a **CERC**.

Table 6-7 Create VPNs and CERCs

Operation	className	Required Parameters
createInstance	<ul style="list-style-type: none"> VPN 	<ul style="list-style-type: none"> Name CERC <p><i>or</i></p> <ul style="list-style-type: none"> CreateDefaultCERC
	<ul style="list-style-type: none"> CERC 	<ul style="list-style-type: none"> Name Provider SpokeRouteTarget HubRouteTarget

A CERC can either be full mesh or hub and spoke. If you specify hub and spoke, you must specify both the **SpokeRouteTarget** and **HubRouteTarget**. For a full mesh CERC, only **SpokeRouteTarget** is required.

XML Examples:

- CreateRouteTarget.xml
- CreateRouteDistinguisher.xml
- CreateCERC.xml
- CreateVPN.xml.

Collect Device Configurations

A device configuration collection is a task. This task uploads the current configuration from the device to the Prime Fulfillment database. The collection task is executed through a service request, and the service request is scheduled using a service order.

Table 6-8 Collect Device Configurations

Operation	className	Required Parameters
createInstance	ServiceOrder	<ul style="list-style-type: none"> • ServiceName • NumberofRequests • ServiceRequest
	ServiceRequest	<ul style="list-style-type: none"> • RequestName • Type=Task • ServiceRequestDetails
	ServiceRequestDetails	<ul style="list-style-type: none"> • SubType=Collection • Device (or DeviceGroup) <p>Note You must select at least one device or device group.</p> <ul style="list-style-type: none"> • RetrieveVersion=true • RetrieveDeviceInterfaces=true

The following example is a partial XML request used to collect the configuration from device ensw4000-1:

```
<className xsi:type="xsd:string">ServiceRequestDetails</className>
  <properties xsi:type="ns1:CIMPropertyList"
    soapenc:arrayType="ns1:CIMProperty[]" >
    <item xsi:type="ns1:CIMProperty">
      <name xsi:type="xsd:string">SubType</name>
      <value xsi:type="xsd:string">COLLECTION</value>
    </item>
    <item xsi:type="ns1:CIMProperty">
      <name xsi:type="xsd:string">Device</name>
      <value xsi:type="xsd:string">ensw4000-1</value>
    </item>
    <item xsi:type="ns1:CIMProperty">
      <name xsi:type="xsd:string">DeviceGroup</name>
      <value xsi:type="xsd:string">PE-Group</value>
    </item>
    <item xsi:type="ns1:CIMProperty">
      <name xsi:type="xsd:string">RetrieveDeviceAttributes</name>
      <value xsi:type="xsd:string">true</value> </item>
    <item xsi:type="ns1:CIMProperty">
      <name xsi:type="xsd:string">RetrieveDeviceInterfaces</name>
      <value xsi:type="xsd:string">true</value> </item>
  </properties>
</objectPath>
```

XML Example:

- CreateTaskServiceOrderCollection.xml

Create an MPLS Policy

An MPLS service policy (defined in a service definition) is a template of the parameters needed to define a service request. Once you have defined the policy template, it can be used by all MPLS service requests that share a common set of attributes.

An MPLS service definition consists of the **MplsPolicyAttributes**. The **MplsPolicyAttributes** define the properties specific to the policy subtype.

Table 6-9 Create an MPLS Policy

Operation	className	Required Parameters
createInstance	ServiceDefinition	<ul style="list-style-type: none"> Name Type=Mpls ServiceDefinitionDetails
	ServiceDefinitionDetails	<ul style="list-style-type: none"> MPLSPolicyAttributes Provider or Organization <p>Note If you do not specify a Provider or Organization, the service policy becomes a global policy.</p>
	MPLSPolicyAttributes	<ul style="list-style-type: none"> SubType= <ul style="list-style-type: none"> PE_CE PE_NO_CE PE_MVRFCE_CE PE_MVRFCE_NO_CE

XML Examples:

- CreateMPLSServiceDefn_PE_CE.xml
- CreateMPLSServiceDefn_PE_NO_CE.xml
- CreateMPLSServiceDefn_MVRF_CE.xml
- CreateMPLSServiceDefn_MVRF_NO_CE.xml.

Creating an MPLS Service Request

An MPLS service request defines the service definition to use, the **MplsVpnLink** and the link attributes. The **MplsVpnLink** specifies the device interfaces involved in this service request. The link attributes contain any policy setting overrides for properties set as editable in the service definition. A service request can specify one or more MPLS VPN links.



Note Use **performBatchOperation** to group the service order and service request into one XML request.



Note The service request name must be unique for each NBI API.

Table 6-10 Create an MPLS Service Request

Operation	className	Required Parameters
createInstance	ServiceOrder	<ul style="list-style-type: none"> • ServiceName • NumberOfRequests • ServiceRequest
	ServiceRequest	<ul style="list-style-type: none"> • RequestName • Type=Mpls • ServiceRequestDetails
	ServiceRequestDetails	<ul style="list-style-type: none"> • ServiceDefinition <ul style="list-style-type: none"> – ServiceDefinitionType=Mpls • MplsVpnLink
	MplsVpnLink	<ul style="list-style-type: none"> • NPC <i>or</i> • ManualConfig=true <ul style="list-style-type: none"> – PE – Cpe <p>Note You must specify either NPC or ManualConfig to define the interfaces for the MPLS VPN links. If you use ManualConfig, you must also specify the interfaces (for example, CE_Intf_Name and PE_Intf_Name).</p> <ul style="list-style-type: none"> • LinkAttrs • LinkTemplate (optional) <p>Note See the “Templates in a Service Request” section on page 4-18.</p>
	LinkAttrs	<ul style="list-style-type: none"> • PE_Intf_Name, CE_Intf_Name • PE_Vlan_ID • PE_Intf_Address, CE_Intf_Address • PE_CE_Routing_Protocol • CERCMembership

**Note**

The attributes **PE_Template**, **PE_Intf_Template**, **CE_Template**, and **CE_Intf_Template** allow NBI access to variables designed to hold template blobs (template blobs were used during MPLS provisioning in legacy versions of Prime Fulfillment).

XML Examples:

- CreateMPLSServiceOrder_PE_CE.xml
- CreateMPLSServiceOrder_PE_NO_CE.xml
- CreateMPLSServiceOrder_MVFR_CE.xml
- CreateMPLSServiceOrder_MVRF_NO_CE.xml.

Auditing Service Requests

A configuration audit occurs automatically each time you deploy a service request. During this configuration audit, Prime Fulfillment verifies that all Cisco IOS commands are present and that they have the correct syntax. An audit also verifies that there were no errors during deployment by examining the commands configured by the service request on the target devices. 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.

If you do not want the configuration audit to occur, change the value for the **Audit** parameter. The **Audit** parameter supports these values:

- **Audit**—This is the default. A successfully deployed service request is automatically audited unless this flag is changed.
- **NoAudit**—Do not perform a configuration audit when the service request is deployed.
- **ForceAudit**—Perform a configuration audit even if the service request deployment is not successful.

You can use the Audit parameter with a **Create**, **Modify**, or **Decommission** service request or a **Deployment** task. See the “[Service Decommission](#)” section on page 3-10 for more information. To perform a configuration audit as a separate task, or an MPLS functional audit, see the “[Tasks](#)” section on page 3-8.