Traffic Engineering Management Provisioning

This chapter describes the provisioning support for Traffic Engineering Management (TEM) provided in Cisco Prime Provisioning.

The TEM API solution provides bulk provisioning, updating, and deletion of traffic engineering (TE) objects.

More specifically, the chapter describes TEM service concepts and the steps required to provision TEM services using the Prime Provisioning API. The provisioning example includes all steps from creating the inventory to auditing the service deployment.

For information on TEM provisioning using the Prime Provisioning GUI, see the *Cisco Prime Provisioning 6.8 User Guide*.

This chapter contains the following sections:

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 6.8 Installation Guide* for general system recommendations and supported platforms.

General Limitations

Let issued service requests finish deployment before issuing other service requests to avoid conflicts.

Prime Provisioning manages a single IS-IS level and multiple OSPF areas. TEM supports one user per OSPF area for managed and backup tunnels and one user per head end device for unmanaged tunnels. However, it does not support 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.

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 the *Cisco Prime Provisioning 6.8 User Guide*.

A number of specific requirements are associated with the **TE Discovery** task. Helpful information is available in the *Cisco Prime Provisioning 6.8 User Guide*.

Concurrent use is supported in the Planning portion of the current implementation of 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.

Non-Cisco Devices and Prime Provisioning 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.

TEM Service Definitions

To provision TEM using the Prime Provisioning API, you need a TEM service definition and a TEM service request (SR). This section lists the supported service definitions, service orders, and policies and includes corresponding examples.

When you deploy a TEM service request using a service order, the attributes specified in the service definition are applied to the devices and interfaces listed in the service request, along with the attributes for each of the links.

Supported TEM Features

Prime Provisioning supports the following TEM features:

- TE Provider
  - Create TE Provider
  - Delete TE Provider
  - Modify TE Provider
  - View TE Provider
- Seed Router
  - Create Seed Router
- TE Discovery Task
  - Create Discovery Task (full and incremental)

  - Discovery Status.
- TE Policy
  - Create Managed and Unmanaged TE Policies
  - Create Managed TE Policy with MPLS enabled/disabled
  - Delete Policy
  - Modify Policy
  - View Policy
- TE Explicit Paths
  - Create Explicit Path
  - Delete Explicit Path
  - Modify Explicit Path
  - View Explicit Path
- TE SRLG (Shared-Risk Link Group)
  - Create SRLG
  - Delete SRLG
  - Modify SRLG
  - View SRLG
- TE Links (created during discovery)
  - Delete TE Links
  - View TE Links
- TE Traffic Admission (with Policy-Based Tunnel Selection [PBTS] or Class-Based Tunnel Selection [PBTS])
  - Create PBTS TE Traffic Admission SR
  - Modify PBTS TE Traffic Admission SR
  - Delete PBTS TE Traffic Admission SR
  - View PBTS TE Traffic Admission SR
  - Create CBTS TE Traffic Admission SR
  - Modify CBTS TE Traffic Admission SR
  - Delete CBTS TE Traffic Admission SR
  - View CBTS TE Traffic Admission SR
- TE Managed Primary Tunnels
  - Create Managed Tunnel
  - Modify Managed Tunnel
  - Delete Managed Tunnel
  - View Managed Tunnel
- TE Unmanaged Primary Tunnels
  - Create Unmanaged Tunnel
  - Modify Unmanaged Tunnel

  - Delete Unmanaged Tunnel
  - View Unmanaged Tunnel
- TE Backup Tunnels
  - Create Backup Tunnel
  - Delete Backup Tunnel
  - Modify Backup Tunnel
  - View TE Tunnel
- TE Routers
  - View TE Routers
- TE Protected Elements
  - Create TE Protected Element
  - View TE Protected Element
  - Create Compute Backup for TE Protected Element
  - View Compute Backup for TE Protected Element.

The above operations are supported for policies and service requests, which are created to provision a variety of network configurations.

The API XMLs for the above operations are contained in the *Cisco Prime Provisioning API 6.8 Programmer Reference.*

Policy Examples

The following constitute some common XML policy examples:

- Create TE Policy, page 10-4
- Delete TE Policy, page 10-6
- View TE Policy, page 10-6.

Create TE Policy

Managed policies have setup/hold priorities of 0/0 and can have additional routing constraints such as protection level and max delay.

Unmanaged policies cannot have a setup/hold priority of zero.

Example: TeManagedPolicyCreateRequest.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">nbiTest</value>
      </item>
      <item xsi:type="ns1:CIMProperty">
        <name xsi:type="xsd:string">Type</name>
        <value xsi:type="xsd:string">Tunnel</value>
      </item>
      <item xsi:type="ns1:CIMProperty">
        <name xsi:type="xsd:string">Provider</name>
        <value xsi:type="xsd:string">provider1</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">HoldPriority</name>
                    <value xsi:type="xsd:string">0</value>
                    </item>
                    <item xsi:type="ns1:CIMProperty">
                    <name xsi:type="xsd:string">FrrProtectionLevel</name>
                    <value xsi:type="xsd:string">None</value>
                    </item>
                    <item xsi:type="ns1:CIMProperty">
                    <name xsi:type="xsd:string">LinkAffinityMask</name>
                    <value xsi:type="xsd:string">0x0</value>
                    </item>
                    <item xsi:type="ns1:CIMProperty">
                    <name xsi:type="xsd:string">SetupPriority</name>
                    <value xsi:type="xsd:string">0</value>
                    </item>
                    <item xsi:type="ns1:CIMProperty">
                    <name xsi:type="xsd:string">TeProvider</name>
                    <value xsi:type="xsd:string">te_provider1</value>
                    </item>
                    <item xsi:type="ns1:CIMProperty">
                    <name xsi:type="xsd:string">Name</name>
                    <value xsi:type="xsd:string">TeMpls-10</value>
                    </item>
          <item xsi:type="ns1:CIMProperty">
                    <name xsi:type="xsd:string">LinkAffinity</name>
                    <value xsi:type="xsd:string">0x0</value>
                    </item>
                    <item xsi:type="ns1:CIMProperty">
                    <name xsi:type="xsd:string">BandwidthPoolType</name>
                    <value xsi:type="xsd:string">GLOBAL</value>
                    </item>
          </properties>
        </objectPath>
      </objectPath>
    </ns1:createInstance>
  </soapenv:Body>
</soapenv:Envelope>
```

Delete TE Policy

A policy can also be deleted based on its name.

Example: TeTunnelPolicyDeleteRequest.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:deleteInstance>
    <objectPath xsi:type="ns1:CIMObjectPath">
       <className xsi:type="xsd:string">ServiceDefinition</className>
       <keyProperties xsi:type="ns1:CIMKeyPropertyList"
                      soapenc:arrayType="ns1:CIMKeyProperty[]">
        <item xsi:type="ns1:CIMKeyProperty">
         <name xsi:type="xsd:string">Name</name>
         <value xsi:type="xsd:string">TeMpls-10</value>
        </item>
        <item xsi:type="ns1:CIMKeyProperty">
         <name xsi:type="xsd:string">Type</name>
         <value xsi:type="xsd:string">Tunnel</value>
        </item>
             </keyProperties>
      </objectPath>
    </ns1:deleteInstance>
  </soapenv:Body>
</soapenv:Envelope>
```

View TE Policy

A policy can be viewed based on its name.

Example: TeTunnelPolicyView.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="86E195A4C6E030C1F93442D46188F361"/>
  </soapenv:Header>
  <soapenv:Body>
   <ns1:enumerateInstances>
   <objectPath xsi:type="ns1:CIMObjectPath">
    <className xsi:type="xsd:string">ServiceDefinition</className>
```

```
    <keyProperties xsi:type="ns1:CIMKeyPropertyList"
                   soapenc:arrayType="ns1:CIMKeyProperty[]">
     <item xsi:type="ns1:CIMKeyProperty">
      <name xsi:type="xsd:string">Name</name>
         <value xsi:type="xsd:string">TeMpls-10</value>
     </item>
     <item xsi:type="ns1:CIMKeyProperty">
      <name xsi:type="xsd:string">Type</name>
      <value xsi:type="xsd:string">Tunnel</value>
     </item>
    </keyProperties>
    </objectPath>
   </ns1:enumerateInstances>
  </soapenv:Body>
</soapenv:Envelope>
```

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.

For more detailed information about TE Discovery, see the *Cisco Prime Provisioning 6.8 User Guide.*

TE Discovery Examples

Incremental TE Discovery includes two types of incremental discovery tasks:

- Create Incremental TE Device Discovery, page 10-7
- Create Incremental TE Link Discovery, page 10-8

Create Incremental TE Device Discovery

This example shows how a device that has been added to the network can be discovered. This device needs to be added in the inventory with the TE ID as management IP address. The relevant attributes are highlighted in bold below.

Example: TeIncrementalDiscoveryDevice.xml

```
<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">Discovery</value>
        </item>
        <item xsi:type="ns1:CIMProperty">
              <name xsi:type="xsd:string">Type</name>
               <value xsi:type="xsd:string">Task</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">SubType</name>
          <value xsi:type="xsd:string">INCREMENTAL_DISCOVERY</value> </item>
        <item xsi:type="ns1:CIMProperty">
         <name xsi:type="xsd:string">TeProvider</name>
         <value xsi:type="xsd:string">te_provider1</value> </item>
       <item xsi:type="ns1:CIMProperty">
         <name xsi:type="xsd:string">DiscoveryDevice</name>
         <value xsi:type="xsd:string"> IOU-Area0-R1 </value>
       </item>
      </properties>
    </objectPath>
  </objectPath>
  </action>
```

Create Incremental TE Link Discovery

This example shows how a link that has been added to the network can be discovered using Link Discovery. The devices EndDeviceA and EndDeviceB must have been discovered already. The relevant attributes are highlighted in bold below.

Example: TeIncrementalDiscoveryLink.xml

```
<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">Discovery</value>
       </item>
       <item xsi:type="ns1:CIMProperty">
        <name xsi:type="xsd:string">Type</name>
        <value xsi:type="xsd:string">Task</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">SubType</name>
           <value xsi:type="xsd:string">INCREMENTAL_DISCOVERY</value> </item>
         <item xsi:type="ns1:CIMProperty">
          <name xsi:type="xsd:string">TeProvider</name>
          <value xsi:type="xsd:string">te_provider1</value> </item>
         <item xsi:type="ns1:CIMProperty">
          <name xsi:type="xsd:string">EndDeviceA</name>
          <value xsi:type="xsd:string">cl-test-12-7600-4</value>
        </item>
        <item xsi:type="ns1:CIMProperty">
          <name xsi:type="xsd:string">InterfaceA</name>
          <value xsi:type="xsd:string">FastEthernet3/48</value>
        </item>
       <item xsi:type="ns1:CIMProperty">
          <name xsi:type="xsd:string">EndDeviceB</name>
          <value xsi:type="xsd:string">isc-cl-test-3925-1</value>
        </item>
        <item xsi:type="ns1:CIMProperty">
```

```
        <name xsi:type="xsd:string">InterfaceB</name>
        <value xsi:type="xsd:string">GigabitEthernet0/2</value>
      </item>
     </properties>
   </objectPath>
</objectPath>
</action>
```

TEM Service Requests

Before creating a service request, a service policy has to be defined. Use a predefined policy template as is or with modifications to create a service request, and deploy the service. For information on TEM policies, see the *Cisco Prime Provisioning 6.8 User Guide*.

A TEM service request defines the service definition to use and applies the needed policy information.

When you deploy a TEM service request using a service order, the attributes specified in the service definition are applied to the devices and interfaces defined in the service request.

This section contains the following:

- TE Topology Example, page 10-9
- Service Request Examples, page 10-10.

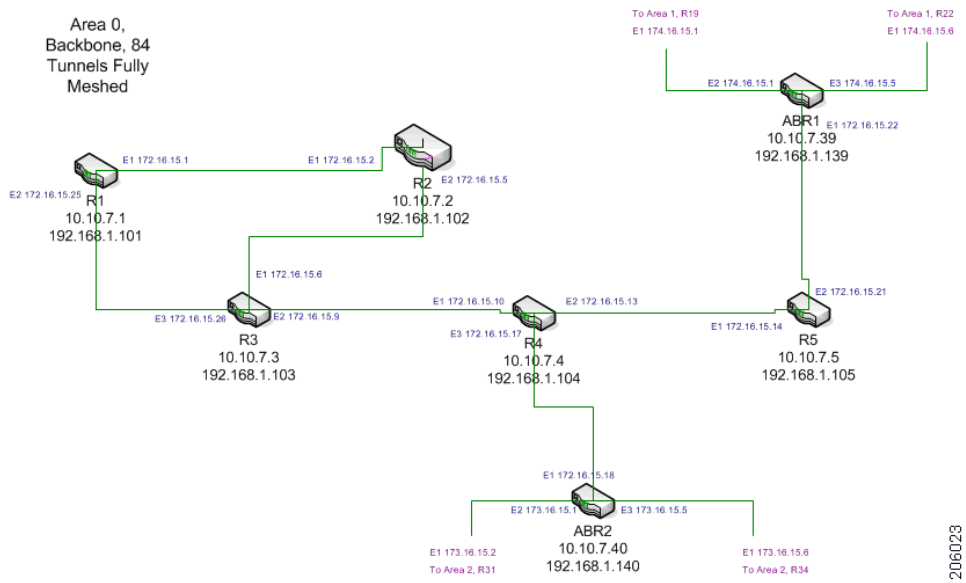
TE Topology Example

As an example of provisioning service requests, this section depicts a TE network in a number of OSPF areas.

A TE network is discovered by first gathering the TE topology information through a seed device. This device must be configured in Prime Provisioning TEM.

Below is a sample network topology with five IOS devices. To discover the network, isctmp1 is used as the seed router. Discovery, management, and provisioning of TE Tunnels is performed within multiple OSPF areas where each area belongs to a separate Prime Provisioning provider.

Figure 10-1 TE Discovery Through Seed Router



For more information about the discovery of multiple OSPF areas, see the *Cisco Prime Provisioning 6.8 User Guide.*

Service Request Examples

The following XML examples show a typical sequence of events from an API perspective, starting with the creation and deployment of a primary tunnel service request and followed by a short example of creating and deploying a backup tunnel service request.

They also demonstrate the kinds of properties that need to be specified for each request.

- Create and Deploy TE Managed Primary Tunnel SR, page 10-10.
- Create and Deploy Unmanaged Primary Tunnel SR, page 10-17.
- Create and Deploy TE Backup Tunnel SR, page 10-18.
- Create and Deploy TE Traffic Admission SR, page 10-19.

Create and Deploy TE Managed Primary Tunnel SR

In this sequence of events, we go through the following steps:

1. Create TE Managed Primary Tunnel SR.
2. Create TE Managed Primary Tunnel SR Response.
3. Request Performance Computation Status.
4. Receive Computation Response.
5. Save and Deploy Managed Primary Tunnel SR Request.
6. Save and Deploy Managed Primary Tunnel SR Response.

Create TE Managed Primary Tunnel SR

In this example, a managed primary tunnel is created. The request XML returns a response with a Computation ID. Using this Computation ID in the deployment XML, we can deploy the managed tunnel.

Example: CreateTeManagedTunnel.xml

```
<?xml version="1.0"?>
<soapenv:Envelope xmlns:ns0="http://www.cisco.com/cim-cx/2.0" xmlns:ns1="urn:CIM"
xmlns:soapenc="http://schemas.xmlsoap.org/soap/encoding/"
xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <soapenv:Header>
    <ns0:message id="87855" sessiontoken="p36bttjwy1"
timestamp="2002-12-13T14:55:38.885Z"/>
  </soapenv:Header>
  <soapenv:Body>
    <ns1:createInstance>
      <objectPath subAction="createInstance" xsi:type="ns1:CIMObjectPath">
        <className xsi:type="xsd:string">TeTunnelSr</className>
        <properties soapenc:arrayType="ns1:CIMProperty[]" xsi:type="ns1:CIMPropertyList">
          <item xsi:type="ns1:CIMProperty">
            <name xsi:type="xsd:string">RequestType</name>
            <value xsi:type="xsd:string">placement</value>
          </item>
          <item xsi:type="ns1:CIMProperty">
            <name xsi:type="xsd:string">TeProvider</name>
            <value xsi:type="xsd:string">te_provider1</value>
          </item>
        </properties>
        <objectPath subAction="createInstance" xsi:type="ns1:CIMObjectPath">
          <className xsi:type="xsd:string">TeTunnel</className>
          <properties soapenc:arrayType="ns1:CIMProperty[]"
xsi:type="ns1:CIMPropertyList">
            <item xsi:type="ns1:CIMProperty">
              <name xsi:type="xsd:string">HeadTeRouter</name>
              <value xsi:type="xsd:string">IOU-Area0-R1</value>
            </item>
            <item xsi:type="ns1:CIMProperty">
              <name xsi:type="xsd:string">TailTeRouter</name>
              <value xsi:type="xsd:string">IOU-Area0-R3</value>
            </item>
            <item xsi:type="ns1:CIMProperty">
              <name xsi:type="xsd:string">Bw</name>
              <value xsi:type="xsd:string">500</value>
            </item>
            <item xsi:type="ns1:CIMProperty">
              <name xsi:type="xsd:string">TePolicy</name>
              <value xsi:type="xsd:string">ISC-P184-IOU-ABR1:Tunnel1000</value>
            </item>
          </properties>
          <objectPath subAction="createInstance" xsi:type="ns1:CIMObjectPath">
            <className xsi:type="xsd:string">TePathOption</className>
            <properties soapenc:arrayType="ns1:CIMProperty[]"
xsi:type="ns1:CIMPropertyList">
              <item xsi:type="ns1:CIMProperty">
                <name xsi:type="xsd:string">PathOptionNumber</name>
                <value xsi:type="xsd:string">1</value>
              </item>
              <item xsi:type="ns1:CIMProperty">
                <name xsi:type="xsd:string">PathType</name>
```

```
                <value xsi:type="xsd:string">SYSTEM</value>
              </item>
            </properties>
          </objectPath>
        </objectPath>
      </objectPath>
    </ns1:createInstance>
  </soapenv:Body>
</soapenv:Envelope>
```

Create TE Managed Primary Tunnel SR Response

The above create request generates the following XML response, which returns a Computation ID.

Example: DeployManagedTunnelComputationResponse.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" sessiontoken="033A021898F763AF632D4914ECDB926E" />
  </soapenv:Header>
  <soapenv:Body>
    <ns1:createInstanceResponse>
      <returns xsi:type="ns1:CIMPropertyList" soapenc:arrayType="ns1:CIMProperty[]">
        <item xsi:type="ns1:CIMProperty">
          <name xsi:type="xsd:string">LocatorId</name>
          <value xsi:type="xsd:string">1</value>
        </item>
        <item xsi:type="ns1:CIMProperty">
          <name xsi:type="xsd:string">ComputationId</name>
          <value xsi:type="xsd:string">11fd0897b0d</value>
        </item>
      </returns>
    </ns1:createInstanceResponse>
  </soapenv:Body>
</soapenv:Envelope>
```

Request Performance Computation Status

Using the Computation ID returned by the above XML response, the Computation ID can now be plugged into a request that goes to the server to check on the status of the computation.

Example: ViewTePrimaryPlanningRequest.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:enumerateInstances>
```

```
            <objectPath xsi:type="ns1:CIMObjectPath">
                <className xsi:type="xsd:string">TeTunnelSr</className>
                <keyProperties xsi:type="ns1:CIMKeyPropertyList"
soapenc:arrayType="ns1:CIMKeyProperty[]">
                    <item xsi:type="ns1:CIMKeyProperty">
                        <name xsi:type="xsd:string">RequestType</name>
                        <value xsi:type="xsd:string">viewPlanning</value>
                    </item>
                    <item xsi:type="ns1:CIMKeyProperty">
                        <name xsi:type="xsd:string">ComputationId</name>
                        <value xsi:type="xsd:string">11fd0897bod</value>
                    </item>
                </keyProperties>
            </objectPath>
        </ns1:enumerateInstances>
    </soapenv:Body>
</soapenv:Envelope>
```

Receive Computation Response

Following the foregoing status request, the Prime Provisioning server will respond that it is still working on it. After the computation is finished, the server reports back whether it was a success or a failure. In this case we have a success.

Example: ViewPerformComputationResponse.xml

```
<?xml version="1.0"?>
<soapenv:Envelope xmlns:ns0="http://www.cisco.com/cim-cx/2.0" xmlns:ns1="urn:CIM"
xmlns:soapenc="http://schemas.xmlsoap.org/soap/encoding/"
xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <soapenv:Header>
    <ns0:message id="87855" sessiontoken="B7E5F6AFE29AA3B9FAA623CAFA3BE76E"
timestamp="2009-03-11T19:20:22.312Z"/>
  </soapenv:Header>
  <soapenv:Body>
    <ns1:enumerateInstancesResponse>
      <returns soapenc:arrayType="ns1:CIMReturn[]" xsi:type="ns1:CIMReturnList">
        <record>
          <objectPath xsi:type="ns1:CIMObjectPath">
            <className xsi:type="xsd:string">TeTunnelSr</className>
            <properties soapenc:arrayType="ns1:CIMProperty[]"
xsi:type="ns1:CIMPropertyList">
              <item xsi:type="ns1:CIMProperty">
                <name xsi:type="xsd:string">ComputationStatus</name>
                <value xsi:type="xsd:string">SUCCESS-SOLUTION_FOUND</value>
              </item>
            </properties>
            <objectPath xsi:type="ns1:CIMObjectPath">
              <className xsi:type="xsd:string">TeTunnel</className>
              <properties soapenc:arrayType="ns1:CIMProperty[]"
xsi:type="ns1:CIMPropertyList">
                <item xsi:type="ns1:CIMProperty">
                  <name xsi:type="xsd:string">TailTeRouter</name>
                  <value xsi:type="xsd:string">IOU-Area0-R3</value>
                </item>
                <item xsi:type="ns1:CIMProperty">
                  <name xsi:type="xsd:string">OpType</name>
                  <value xsi:type="xsd:string">ADD</value>
                </item>
```

```
<item xsi:type="ns1:CIMProperty">
  <name xsi:type="xsd:string">HeadTeRouterId</name>
  <value xsi:type="xsd:string">5770</value>
</item>
<item xsi:type="ns1:CIMProperty">
  <name xsi:type="xsd:string">ElementStackId</name>
  <value xsi:type="xsd:string">ISC-P1</value>
</item>
<item xsi:type="ns1:CIMProperty">
  <name xsi:type="xsd:string">TailTeRouterId</name>
  <value xsi:type="xsd:string">5774</value>
</item>
<item xsi:type="ns1:CIMProperty">
  <name xsi:type="xsd:string">Conforming</name>
  <value xsi:type="xsd:string">true</value>
</item>
<item xsi:type="ns1:CIMProperty">
  <name xsi:type="xsd:string">AdminStatus</name>
  <value xsi:type="xsd:string"/>
</item>
<item xsi:type="ns1:CIMProperty">
  <name xsi:type="xsd:string">HeadTeRouter</name>
  <value xsi:type="xsd:string">IOU-Area0-R1</value>
</item>
<item xsi:type="ns1:CIMProperty">
  <name xsi:type="xsd:string">LspInUse</name>
  <value xsi:type="xsd:string"/>
</item>
<item xsi:type="ns1:CIMProperty">
  <name xsi:type="xsd:string">TunnelNumber</name>
  <value xsi:type="xsd:string">1000</value>
</item>
<item xsi:type="ns1:CIMProperty">
  <name xsi:type="xsd:string">RouteServerVerified</name>
  <value xsi:type="xsd:string">1</value>
</item>
<item xsi:type="ns1:CIMProperty">
  <name xsi:type="xsd:string">ReRouteEnabled</name>
  <value xsi:type="xsd:string">true</value>
</item>
<item xsi:type="ns1:CIMProperty">
  <name xsi:type="xsd:string">State</name>
  <value xsi:type="xsd:string">REQUESTED</value>
</item>
<item xsi:type="ns1:CIMProperty">
  <name xsi:type="xsd:string">AutoBwEnabled</name>
  <value xsi:type="xsd:string">false</value>
</item>
<item xsi:type="ns1:CIMProperty">
  <name xsi:type="xsd:string">TePolicy</name>
  <value xsi:type="xsd:string">ISC-P184-IOU-ABR1:Tunnel1000</value>
</item>
<item xsi:type="ns1:CIMProperty">
  <name xsi:type="xsd:string">Descriptor</name>
  <value xsi:type="xsd:string">ISC-P1</value>
</item>
<item xsi:type="ns1:CIMProperty">
  <name xsi:type="xsd:string">Bw</name>
  <value xsi:type="xsd:string">500</value>
</item>
<item xsi:type="ns1:CIMProperty">
  <name xsi:type="xsd:string">OperationStatus</name>
  <value xsi:type="xsd:string"/>
</item>
```

```
                  <item xsi:type="ns1:CIMProperty">
                    <name xsi:type="xsd:string">ChangeAchieved</name>
                    <value xsi:type="xsd:string">All</value>
                  </item>
                  <item xsi:type="ns1:CIMProperty">
                    <name xsi:type="xsd:string">ChangeType</name>
                    <value xsi:type="xsd:string">Tunnel Add Change</value>
                  </item>
                  <item xsi:type="ns1:CIMProperty">
                    <name xsi:type="xsd:string">ChangeOrigin</name>
                    <value xsi:type="xsd:string">Compute</value>
                  </item>
                </properties>
                <objectPath xsi:type="ns1:CIMObjectPath">
                  <className xsi:type="xsd:string">TePathOption</className>
                  <properties soapenc:arrayType="ns1:CIMProperty[]"
xsi:type="ns1:CIMPropertyList">
                    <item xsi:type="ns1:CIMProperty">
                      <name xsi:type="xsd:string">LockdownEnabled</name>
                      <value xsi:type="xsd:string">false</value>
                    </item>
                    <item xsi:type="ns1:CIMProperty">
                      <name xsi:type="xsd:string">PathOptionNumber</name>
                      <value xsi:type="xsd:string">1</value>
                    </item>
                    <item xsi:type="ns1:CIMProperty">
                      <name xsi:type="xsd:string">PathType</name>
                      <value xsi:type="xsd:string">SYSTEM</value>
                    </item>
                  </properties>
                </objectPath>
              </objectPath>
            </objectPath>
          </record>
        </returns>
      </ns1:enumerateInstancesResponse>
    </soapenv:Body>
</soapenv:Envelope>
```

Save and Deploy Managed Primary Tunnel SR Request

Based on the successful computation in the previous example, we can now use the Computation ID to save and deploy the service request that represents the solution computed by the Prime Provisioning server. After it has run, it returns the subsequrent deployManagedComputation.xml response.

Example: DeployManagedTunnelComputationRequest.xml

```
<?xml version="1.0"?>
<soapenv:Envelope xmlns:ns0="http://www.cisco.com/cim-cx/2.0" xmlns:ns1="urn:CIM"
xmlns:soapenc="http://schemas.xmlsoap.org/soap/encoding/"
xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <soapenv:Header>
    <ns0:message Wait="false" WaitTimeout="60" id="87855"
sessiontoken="18E1AEAA851F25B08DE5B1CA6C9E59E6" timestamp="2002-12-13T14:55:38.885Z"/>
  </soapenv:Header>
  <soapenv:Body>
    <ns1:performBatchOperation>
      <actions soapenc:arrayType="ns1:CIMAction[]" xsi:type="ns1:CIMActionList">
        <action>
```

```
          <actionName xsi:type="xsd:string">createInstance</actionName>
          <objectPath xsi:type="ns1:CIMObjectPath">
            <className xsi:type="xsd:string">ServiceOrder</className>
            <properties soapenc:arrayType="ns1:CIMProperty[]"
xsi:type="ns1:CIMPropertyList">
              <item xsi:type="ns1:CIMProperty">
                <name xsi:type="xsd:string">ServiceName</name>
                <value xsi:type="xsd:string">SAVE_AND_DEPLOY</value>
              </item>
              <item xsi:type="ns1:CIMProperty">
                <name xsi:type="xsd:string">CarrierId</name>
                <value xsi:type="xsd:string">101</value>
              </item>
              <item xsi:type="ns1:CIMProperty">
                <name xsi:type="xsd:string">DesiredDueDate</name>
                <value xsi:type="xsd:dateTime">2003-12-14T14:55:38.885Z</value>
              </item>
              <item xsi:type="ns1:CIMProperty">
                <name xsi:type="xsd:string">NumberOfRequests</name>
                <value xsi:type="xsd:string">1</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 soapenc:arrayType="ns1:CIMProperty[]"
xsi:type="ns1:CIMPropertyList">
              <item xsi:type="ns1:CIMProperty">
                <name xsi:type="xsd:string">RequestName</name>
                <value xsi:type="xsd:string">SAVE_AND_DEPLOY</value>
              </item>
              <item xsi:type="ns1:CIMProperty">
                <name xsi:type="xsd:string">Type</name>
                <value xsi:type="xsd:string">Task</value>
              </item>
            </properties>
            <objectPath xsi:type="ns1:CIMObjectPath">
              <className xsi:type="xsd:string">ServiceRequestDetails</className>
              <properties soapenc:arrayType="ns1:CIMProperty[]"
xsi:type="ns1:CIMPropertyList">
                <item xsi:type="ns1:CIMProperty">
                  <name xsi:type="xsd:string">SubType</name>
                  <value xsi:type="xsd:string">SAVE_AND_DEPLOY</value>
                </item>
                <item xsi:type="ns1:CIMProperty">
                  <name xsi:type="xsd:string">ComputationId</name>
                  <value xsi:type="xsd:string">11fd0897bod</value>
                </item>
              </properties>
            </objectPath>
          </objectPath>
        </action>
      </actions>
    </ns1:performBatchOperation>
  </soapenv:Body>
</soapenv:Envelope>
```

Save and Deploy Managed Primary Tunnel SR Response

In response to the foregoing deployment request, the following example executes the deployment based on the provided Computation ID.

Example: DeployManagedTunnelComputationResponse.xml

```
<?xml version="1.0"?>
soapenv:Envelope xmlns:ns0="http://www.cisco.com/cim-cx/2.0" xmlns:ns1="urn:CIM"
xmlns:soapenc="http://schemas.xmlsoap.org/soap/encoding/"
xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
 <soapenv:Header>
   <ns0:message id="87855" sessiontoken="B7E5F6AFE29AA3B9FAA623CAFA3BE76E"
timestamp="2009-03-11T19:20:23.487Z" wait="false" waittimeout="60"/>
 </soapenv:Header>
  <soapenv:Body>
    <ns1:performBatchOperationResponse>
     <returns soapenc:arrayType="ns1:CIMActionResponse[]"
xsi:type="ns1:CIMActionResponseList">
       <actionResponse>
         <actionName xsi:type="xsd:string">createInstanceResponse</actionName>
         <objectPath xsi:type="ns1:CIMObjectPath">
           <className xsi:type="xsd:string">ServiceOrder</className>
           <properties soapenc:arrayType="ns1:CIMProperty[]"
xsi:type="ns1:CIMPropertyList">
             <item xsi:type="ns1:CIMProperty">
               <name xsi:type="xsd:string">LocatorId</name>
               <value xsi:type="xsd:string">384</value>
             </item>
             <item xsi:type="ns1:CIMProperty">
               <name xsi:type="xsd:string">ServiceName</name>
               <value xsi:type="xsd:string">SAVE_AND_DEPLOY</value>
             </item>
            </properties>
         </objectPath>
       </actionResponse>
       <actionResponse>
         <actionName xsi:type="xsd:string">createInstanceResponse</actionName>
         <objectPath xsi:type="ns1:CIMObjectPath">
           <className xsi:type="xsd:string">ServiceRequest</className>
           <properties soapenc:arrayType="ns1:CIMProperty[]"
xsi:type="ns1:CIMPropertyList">
             <item xsi:type="ns1:CIMProperty">
               <name xsi:type="xsd:string">RequestName</name>
               <value xsi:type="xsd:string">SAVE_AND_DEPLOY</value>
             </item>
           </properties>
         </objectPath>
       </actionResponse>
     </returns>
   </ns1:performBatchOperationResponse>
 </soapenv:Body>
/soapenv:Envelope>
```

Create and Deploy Unmanaged Primary Tunnel SR

Unmanaged tunnels are provisioned in the same manner as backup tunnels. See the *Cisco Prime Provisioning API 6.8 Programmer Reference* for XML examples.

Example: CreateTeUnmanagedTunnel.xml.

Create and Deploy TE Backup Tunnel SR

This request XML creates and deploys a backup tunnel based on a similar set of XML requests and responses as described above for the managed primary tunnel.

Example: CreateTeBackupTunnel.xml

```
<?xml version="1.0"?>
<soapenv:Envelope xmlns:ns0="http://www.cisco.com/cim-cx/2.0" xmlns:ns1="urn:CIM"
xmlns:soapenc="http://schemas.xmlsoap.org/soap/encoding/"
xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <soapenv:Header>
    <ns0:message id="87855" sessiontoken="p36bttjwy1"
timestamp="2002-12-13T14:55:38.885Z"/>
  </soapenv:Header>
  <soapenv:Body>
    <ns1:createInstance>
      <objectPath subAction="createInstance" xsi:type="ns1:CIMObjectPath">
        <className xsi:type="xsd:string">TeBrTunnelSr</className>
        <properties soapenc:arrayType="ns1:CIMProperty[]" xsi:type="ns1:CIMPropertyList">
          <item xsi:type="ns1:CIMProperty">
            <name xsi:type="xsd:string">RequestType</name>
            <value xsi:type="xsd:string">computeBackup</value>
          </item>
          <item xsi:type="ns1:CIMProperty">
            <name xsi:type="xsd:string">TeProvider</name>
            <value xsi:type="xsd:string">te_provider1</value>
          </item>
        </properties>
        <objectPath subAction="createInstance" xsi:type="ns1:CIMObjectPath">
          <className xsi:type="xsd:string">TeBrTunnel</className>
          <properties soapenc:arrayType="ns1:CIMProperty[]"
xsi:type="ns1:CIMPropertyList">
            <item xsi:type="ns1:CIMProperty">
              <name xsi:type="xsd:string">HeadTeRouter</name>
              <value xsi:type="xsd:string"> IOU-Area0-R3</value>
            </item>
            <item xsi:type="ns1:CIMProperty">
              <name xsi:type="xsd:string">TailTeRouter</name>
              <value xsi:type="xsd:string"> IOU-Area0-R5</value>
            </item>
            <item xsi:type="ns1:CIMProperty">
              <name xsi:type="xsd:string">Bw</name>
              <value xsi:type="xsd:string">500</value>
            </item>
        <item xsi:type="ns1:CIMProperty">
              <name xsi:type="xsd:string">ProtectedInterface</name>
              <value xsi:type="xsd:string"> Ethernet1/0</value>
            </item>
          </properties>
          <objectPath subAction="createInstance" xsi:type="ns1:CIMObjectPath">
            <className xsi:type="xsd:string">TePathOption</className>
            <properties soapenc:arrayType="ns1:CIMProperty[]"
xsi:type="ns1:CIMPropertyList">
              <item xsi:type="ns1:CIMProperty">
                <name xsi:type="xsd:string">PathOptionNumber</name>
                <value xsi:type="xsd:string">1</value>
              </item>
```

```
                <item xsi:type="ns1:CIMProperty">
                  <name xsi:type="xsd:string">PathName</name>
                  <value xsi:type="xsd:string"> IOU-Area0-R3-IOU-Area0-R5-1</value>
                </item>
                <item xsi:type="ns1:CIMProperty">
                  <name xsi:type="xsd:string">PathType</name>
                  <value xsi:type="xsd:string">EXPLICIT</value>
                </item>
              </properties>
            </objectPath>
          </objectPath>
        </objectPath>
      </ns1:createInstance>
    </soapenv:Body>
</soapenv:Envelope>
```

Backup Tunnel SR Response

The following is an example of a response to the backup tunnel service request.

Example: DeployTeBackupTunnel.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" sessiontoken="3A2F67CAFDA12EBEFF773EF55573A17A" />
  </soapenv:Header>
  <soapenv:Body>
   <ns1:createInstanceResponse>
      <returns xsi:type="ns1:CIMPropertyList" soapenc:arrayType="ns1:CIMProperty[]">
        <item xsi:type="ns1:CIMProperty">
          <name xsi:type="xsd:string">LocatorId</name>
          <value xsi:type="xsd:string">19</value>
        </item>
      </returns>
    </ns1:createInstanceResponse>
  </soapenv:Body>
</soapenv:Envelope>
```

Create and Deploy TE Traffic Admission SR

Based on the Locator ID, you can now use a TE Traffic Admission service request to enable services on the created tunnel.

This involves the following steps:

1. Create TE Traffic Admission SR Request.
2. Create TE Traffic Admission SR Response.
3. Save and Deploy TE Traffic Admission SR Request.
4. Save and Deploy TE Traffic Admission SR Response.

Create TE Traffic Admission SR Request

In this example, a TE Traffic Admission service request is created to provision a TE Tunnel.

Example: CreateTeAdmissionSrCBTSRequest.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="343BDF4AEB6A7A4F5D1A24FBF3EC9A50"/>
  </soapenv:Header>
  <soapenv:Body>
  <ns1:createInstance>
    <objectPath xsi:type="ns1:CIMObjectPath">
    <className xsi:type="xsd:string">TeAdmissionSr</className>
    <properties xsi:type="ns1:CIMPropertyList"
                soapenc:arrayType="ns1:CIMProperty[]">
              <item xsi:type="ns1:CIMProperty">
        <name xsi:type="xsd:string">Description</name>
        <value xsi:type="xsd:string">Provider1:nw1-r4-7206-g1:1017</value>
          </item>
          <item xsi:type="ns1:CIMProperty">
        <name xsi:type="xsd:string">TeRouterHostname</name>
        <value xsi:type="xsd:string">nw1-r4-7206-g1</value>
          </item>
          <item xsi:type="ns1:CIMProperty">
        <name xsi:type="xsd:string">TunnelNumber</name>
        <value xsi:type="xsd:string">1017</value>
          </item>
          <item xsi:type="ns1:CIMProperty">
        <name xsi:type="xsd:string">EXPMarking</name>
        <value xsi:type="xsd:string">0 2 4 6</value>
          </item>
          <item xsi:type="ns1:CIMProperty">
        <name xsi:type="xsd:string">TeProvider</name>
        <value xsi:type="xsd:string">Provider1</value>
          </item>
          <item xsi:type="ns1:CIMProperty">
        <name xsi:type="xsd:string">Metric</name>
        <value xsi:type="xsd:string">6</value>
          </item>
          <item xsi:type="ns1:CIMProperty">
        <name xsi:type="xsd:string">MetricType</name>
        <value xsi:type="xsd:string">2</value>
          </item>
          <item xsi:type="ns1:CIMProperty">
        <name xsi:type="xsd:string">AutoRouteAnnounceEnabled</name>
        <value xsi:type="xsd:string">true</value>
          </item>
    </properties>
    </objectPath>
  </ns1:createInstance>
  </soapenv:Body>
</soapenv:Envelope>
```

Create TE Traffic Admission SR Response

If the service request is successful, the response provides a Locator ID needed for the subsequent deployment step.

Example: CreateTeAdmissionSrCBTSResponse.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" sessiontoken="343BDF4AEB6A7A4F5D1A24FBF3EC9A50"
timestamp="2006-10-09T17:46:42.677Z" />
  </soapenv:Header>
  <soapenv:Body>
    <ns1:createInstanceResponse>
      <returns xsi:type="ns1:CIMPropertyList" soapenc:arrayType="ns1:CIMProperty[]">
        <item xsi:type="ns1:CIMProperty">
          <name xsi:type="xsd:string">LocatorId</name>
          <value xsi:type="xsd:string">22</value>
        </item>
      </returns>
    </ns1:createInstanceResponse>
  </soapenv:Body>
</soapenv:Envelope>
```

Save and Deploy TE Traffic Admission SR Request

Run the following XML to save and deploy the TE Traffic Admission service request and have it provisioned on the device.

Example: DeployTeAdmissionSrRequest.xml

```
<?xml version="1.0"?>
<soapenv:Envelope xmlns:ns0="http://www.cisco.com/cim-cx/2.0" xmlns:ns1="urn:CIM"
xmlns:soapenc="http://schemas.xmlsoap.org/soap/encoding/"
xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <soapenv:Header>
     <ns0:message Wait="false" WaitTimeout="60" id="87855" sessiontoken="p36bttjwy1"
timestamp="2002-12-13T14:55:38.885Z"/>
  </soapenv:Header>
  <soapenv:Body>
   <ns1:performBatchOperation>
    <actions soapenc:arrayType="ns1:CIMAction[]" xsi:type="ns1:CIMActionList">
     <action>
      <actionName xsi:type="xsd:string">createInstance</actionName>
      <objectPath xsi:type="ns1:CIMObjectPath">
       <className xsi:type="xsd:string">ServiceOrder</className>
       <properties soapenc:arrayType="ns1:CIMProperty[]" xsi:type="ns1:CIMPropertyList">
        <item xsi:type="ns1:CIMProperty">
         <name xsi:type="xsd:string">ServiceName</name>
         <value xsi:type="xsd:string">TEM-TRADM-CBTS-PROV-001</value>
        </item>
        <item xsi:type="ns1:CIMProperty">
         <name xsi:type="xsd:string">CarrierId</name>
         <value xsi:type="xsd:string">10</value>
```

```
          </item>
          <item xsi:type="ns1:CIMProperty">
           <name xsi:type="xsd:string">DesiredDueDate</name>
           <value xsi:type="xsd:dateTime">2002-12-14T14:55:38.885Z</value>
          </item>
          <item xsi:type="ns1:CIMProperty">
           <name xsi:type="xsd:string">NumberOfRequests</name>
           <value xsi:type="xsd:string">1</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 soapenc:arrayType="ns1:CIMProperty[]" xsi:type="ns1:CIMPropertyList">
           <item xsi:type="ns1:CIMProperty">
             <name xsi:type="xsd:string">RequestName</name>
             <value xsi:type="xsd:string">DEPLOYMENT-TASK</value>
           </item>
           <item xsi:type="ns1:CIMProperty">
             <name xsi:type="xsd:string">Type</name>
             <value xsi:type="xsd:string">Task</value>
           </item>
         </properties>
         <objectPath xsi:type="ns1:CIMObjectPath">
           <className xsi:type="xsd:string">ServiceRequestDetails</className>
           <properties soapenc:arrayType="ns1:CIMProperty[]" xsi:type="ns1:CIMPropertyList">
             <item xsi:type="ns1:CIMProperty">
               <name xsi:type="xsd:string">SubType</name>
               <value xsi:type="xsd:string">DEPLOYMENT</value>
             </item>
             <item xsi:type="ns1:CIMProperty">
               <name xsi:type="xsd:string">LocatorId</name>
               <value xsi:type="xsd:string">22</value>
             </item>
             <item xsi:type="ns1:CIMProperty">
               <name xsi:type="xsd:string">ForceDeploy</name>
               <value xsi:type="xsd:string">false</value>
             </item>
             <item xsi:type="ns1:CIMProperty">
               <name xsi:type="xsd:string">Audit</name>
               <value xsi:type="xsd:string">ForceAudit</value>
             </item>
           </properties>
         </objectPath>
     </objectPath>
      </action>
     </actions>
    </ns1:performBatchOperation>
   </soapenv:Body>
</soapenv:Envelope>
```

Save and Deploy TE Traffic Admission SR Response

After the TE Traffic Admission service request has been processed, a response is returned.

Example: DeployTeAdmissionSrResponse.xml

```
<?xml version="1.0"?>
```

```
<soapenv:Envelope xmlns:ns0="http://www.cisco.com/cim-cx/2.0" xmlns:ns1="urn:CIM"
xmlns:soapenc="http://schemas.xmlsoap.org/soap/encoding/"
xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <soapenv:Header>
    <ns0:message id="87855" sessiontoken="B7E5F6AFE29AA3B9FAA623CAFA3BE76E"
timestamp="2009-03-11T21:34:48.808Z" wait="false" waittimeout="60"/>
  </soapenv:Header>
  <soapenv:Body>
    <ns1:performBatchOperationResponse>
      <returns soapenc:arrayType="ns1:CIMActionResponse[]"
xsi:type="ns1:CIMActionResponseList">
        <actionResponse>
          <actionName xsi:type="xsd:string">createInstanceResponse</actionName>
          <objectPath xsi:type="ns1:CIMObjectPath">
            <className xsi:type="xsd:string">ServiceOrder</className>
            <properties soapenc:arrayType="ns1:CIMProperty[]"
xsi:type="ns1:CIMPropertyList">
              <item xsi:type="ns1:CIMProperty">
                <name xsi:type="xsd:string">LocatorId</name>
                <value xsi:type="xsd:string">22</value>
              </item>
              <item xsi:type="ns1:CIMProperty">
                <name xsi:type="xsd:string">ServiceName</name>
                <value xsi:type="xsd:string">TEM-TRADM-CBTS-PROV-001</value>
              </item>
            </properties>
          </objectPath>
        </actionResponse>
        <actionResponse>
          <actionName xsi:type="xsd:string">createInstanceResponse</actionName>
          <objectPath xsi:type="ns1:CIMObjectPath">
            <className xsi:type="xsd:string">ServiceRequest</className>
            <properties soapenc:arrayType="ns1:CIMProperty[]"
xsi:type="ns1:CIMPropertyList">
              <item xsi:type="ns1:CIMProperty">
                <name xsi:type="xsd:string">RequestName</name>
                <value xsi:type="xsd:string">DEPLOYMENT-TASK</value>
              </item>
            </properties>
          </objectPath>
        </actionResponse>
      </returns>
    </ns1:performBatchOperationResponse>
  </soapenv:Body>
</soapenv:Envelope>
```

After this response has been received, the TE Traffic Admission service request will be in the Requested state in Prime Provisioning. It now needs to be saved and deployed to actually be provisioned on the device.

Note TE Traffic Admission SR's are deleted and unprovisioned from a device by first decommissioning the TE Traffic Admission service request and then saving and deploying the same service request. By saving and deploying the service request after decommissioning it, the TE Traffic Admission configuration is removed from the device. Force Purge will only remove the TE Traffic Admission service request from Prime Provisioning's database; it will not remove the configuration from the device.

Provisioning Example

This section describes the process for using the API to provision TEM, and includes the required operation, object definition (className), and parameter definitions.

This section contains the following:

Process Summary

This TEM provisioning example documents the following process:

1. Create provider.
2. Create regions.
3. Create seed device.
4. Create TE provider.
5. Create and run TE Discovery task.
6. Create service definition (policy).
7. Create explicit path.
8. Create managed primary tunnel.
9. Create service request.

Provisioning Process

To provision TEM using Prime Provisioning APIs, a TEM service definition (policy) and a TEM service request are required.

This section describes the process for provisioning TEM using XML examples.

The complete list of XML examples for TEM is available here:
Cisco Prime Provisioning API 6.8 Programmer Reference

Note For clarity, this provisioning process shows each step as a separate XML request. Many of these steps can be combined using **performBatchOperations**.

Step 1 Create a provider.

The provider is the administrative domain of an ISP, with one 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.

Use the TeAreaIdentifier parameter to define OSPF areas and for discovering multiple OSPF areas.

Table 10-1 Create a Provider

Operation	className	Required Parameters
createInstance	Provider	• Name • AsNumber • TeAreaIdentifier (optional)

XML Examples:

ISCProviderCreateRequest.xml

Step 2 Create regions.

Each provider can contain multiple regions.

Table 10-2 Create a Region

Operation	className	Required Parameters
createInstance	Region	• Name • Provider

XML Examples:

ISCDefaultRegionCreateRequest.xml

Step 3 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.

Table 10-3 Create a Seed Device

Operation	className	Required Parameters
createInstance	CiscoRouter	• HostName • Login • Password • EnablePassword • SnmpRo • SnmpRw

XML Examples:

ISCSeedRouterCreateRequest.xml

Step 4 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 .

Table 10-4 Create a TE Provider

Operation	className	Required Parameters
createInstance	TeProvider	• PrimaryRgTimeout • Provider • DefaultRegion • BackupRgTimeout • MinBwLimit • MaxTunnelCount • FrrProtectionType

XML Examples:

- ISCTEProviderCreateRequest.xml

Step 5 Run a TE Full Discovery task.

Discover the TE network for a particular TE provider to populate the repository with a view to creating primary and backup tunnels.

Table 10-5 Create a TE Full Discovery Task

Operation	className	Required Parameters
createInstance	Discovery	• DesiredDueDate • TeProvider • SeedRouter

XML Examples:

- TEDiscoveryTaskCreateRequest.xml
- incremetal_discovery_device
- incremetal_discovery_link

Step 6 Create service definition (policy).

Service definitions or policies are used to define common tunnel attributes.

Table 10-6 Create a TEM Service Definition (Policy)

Operation	className	Required Parameters
createInstance	ServiceDefinition	• Name • Type • Provider
	ServiceDefinitionDetails	• TeProvider • Name • HoldPriority • FrrProtectionLevel • LinkAffinityMask • SetupPriority • LinkAffinity • BandwidthPoolType

XML Examples:

- TeManagedPolicyCreateRequest.xml
- TeUnmanagedPolicyCreateRequest.xml.

Step 7 Create explicit path.

Paths are defined between source and destination routers, possibly with one or more hops in between. For managed tunnels, the path should not contain any non-TE enabled interfaces.

Table 10-7 Create an Explicit Path

Operation	className	Required Parameters
createInstance	TeExplicitPath	• TeExpPathType • PathName • TeProvider • HeadTeRouter • Provisioning-Pref
	TeLspHop	• TeLspHopType • IpAddress

XML Examples:

- TeExplicitPathExcludeCreateRequest.xml

Step 8 Create a managed primary tunnel.

Once a TE Policy and an explicit path have been set up, a primary tunnel can be created. The process is very similar for managed and unmanaged tunnels.

Table 10-8 Create a Managed Primary Tunnel

Operation	className	Required Parameters
createInstance	TeTunnelSr	• RequestType • TeProvider
	TeTunnel	• HeadTeRouter • TailTeRouter • Bw • TePolicy
	TePathOption	• PathOptionNumber • PathType

XML Examples:

- CreateManagedTunnel.xml

Step 9 Create service request.

A TE service request defines the service definition and assigns interfaces and attributes.

Table 10-9 Create a TE Service Request

Operation	className	Required Parameters
createInstance	ServiceOrder	• ServiceName • DesiredDueDate • NumberOfRequests
	ServiceRequest	• RequestName • Type=Task • ServiceRequestDetails
	ServiceRequestDetails	• SubType • ComputationId

Tip Record the **LocatorId** value from the XML response for the service request. The Locator ID is required for subsequent service request tasks.

XML Example:

- SaveAndDeployPlanningRequest.xml

Using Computation ID and Locator ID

The ComputationID is returned by TEM for certain processes and is as important as the Locator ID.

Locator ID refers to a service request object in Prime Provisioning that you are working on, whereas Computation ID refers to a server computation that must be completed satisfactorily before you can work with the servicer request object (Locator ID) again. The Computation ID is returned by an API request that require server input, for example any managed tunnel operations.

Examples of how the Computation ID and Locator ID are used are found in the section Service Request Examples, page 10-10.

Planning Tools

Planning Tools, also referred to as Placement Tools, are used to perform planning functions on the existing network. They are intended for evaluating planned improvements to a traffic-engineered network based on What-If scenarios.

At the present time, most of these tools do not have API support (except Compute Backup, page 10-29) but can be activated from the GUI. See the *Cisco Prime Provisioning 6.8 User Guide* under Advanced Primary Tunnel Management and Protection Planning.

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.

Compute Backup

Compute Backup is used to let TEM automatically compute the necessary backup tunnels to protect specified network elements.

The Compute Backup examples are found in the *Cisco Prime Provisioning 6.8 User Guide* in the tem\TeProtectedElements folder.

The following example shows how Compute Backup is performed for a TE node.

Example: CreateTeProtectedElementNode.xml

```
<?xml version="1.0"?>
<soapenv:Envelope xmlns:ns0="http://www.cisco.com/cim-cx/2.0" xmlns:ns1="urn:CIM"
xmlns:soapenc="http://schemas.xmlsoap.org/soap/encoding/"
xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <soapenv:Header>
     <ns0:message Wait="false" WaitTimeout="60" id="87855" sessiontoken="p36bttjwy1"
timestamp="2002-12-13T14:55:38.885Z"/>
  </soapenv:Header>
  <soapenv:Body>
   <ns1:performBatchOperation>
    <actions soapenc:arrayType="ns1:CIMAction[]" xsi:type="ns1:CIMActionList">
     <action>
      <actionName xsi:type="xsd:string">createInstance</actionName>
      <objectPath xsi:type="ns1:CIMObjectPath">
       <className xsi:type="xsd:string">ServiceOrder</className>
       <properties soapenc:arrayType="ns1:CIMProperty[]" xsi:type="ns1:CIMPropertyList">
        <item xsi:type="ns1:CIMProperty">
         <name xsi:type="xsd:string">ServiceName</name>
         <value xsi:type="xsd:string">TEM-IOX-TRADM-VX.X-014</value>
        </item>
        <item xsi:type="ns1:CIMProperty">
         <name xsi:type="xsd:string">CarrierId</name>
         <value xsi:type="xsd:string">10</value>
        </item>
        <item xsi:type="ns1:CIMProperty">
         <name xsi:type="xsd:string">DesiredDueDate</name>
         <value xsi:type="xsd:dateTime">2002-12-14T14:55:38.885Z</value>
        </item>
        <item xsi:type="ns1:CIMProperty">
         <name xsi:type="xsd:string">NumberOfRequests</name>
         <value xsi:type="xsd:string">1</value>
        </item>
       </properties>
      </objectPath>
     </action>
     <action>
       <actionName xsi:type="xsd:string">createInstance</actionName>
       <objectPath subAction="modifyInstance" xsi:type="ns1:CIMObjectPath">
<className xsi:type="xsd:string">TeBrTunnelSr</className>
<keyProperties soapenc:arrayType="ns1:CIMProperty[]" xsi:type="ns1:CIMKeyPropertyList">
<item xsi:type="ns1:CIMProperty">
<name xsi:type="xsd:string">TeProvider</name>
<value xsi:type="xsd:string">Provider1</value>
</item>
</keyProperties>
<properties soapenc:arrayType="ns1:CIMProperty[]" xsi:type="ns1:CIMPropertyList">
<item xsi:type="ns1:CIMProperty">
<name xsi:type="xsd:string">RequestType</name>
<value xsi:type="xsd:string">computeBackup</value>
</item>
</properties>
<objectPath subAction="createInstance" xsi:type="ns1:CIMObjectPath">
<className xsi:type="xsd:string">TeProtectedElement</className>
<properties soapenc:arrayType="ns1:CIMProperty[]" xsi:type="ns1:CIMPropertyList">
<item xsi:type="ns1:CIMProperty">
<name xsi:type="xsd:string">Type</name>
<value xsi:type="xsd:string">NODE</value>
</item>
<item xsi:type="ns1:CIMProperty">
<name xsi:type="xsd:string">Name</name>
<value xsi:type="xsd:string">nw1-r4-7206-g1</value>
</item>
```

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

Auditing Service Requests

A configuration audit occurs automatically each time you deploy a service request. During this configuration audit, 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 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 3-11 for more information. To perform a configuration audit as a separate task, see the "Configuration Audit" section on page 3-11.