
# NAT Provisioning

**This feature is not supported in this release.**

Network Address Translation (NAT) is used to translate IP addresses that are not globally unique into routable IP addresses. NAT provisioning with the Cisco IP Solution Center (ISC) supports network, host, or portion-based static translations, dynamic translations (including NAT and PAT) for both Internet-bound and site-to-site traffic, alternative IP address pools, port-to-port traffic redirect, and overlapping IP address spaces.

Use the ISC API to provision NAT services for the following network configurations:

- Single site—A site with private addresses that require access to the Internet.
- Multiple sites without overlapping IP addresses—Sites with private addresses that need to communicate with other private addresses and, at the same time, require access to the Internet.
- Multiple sites with overlapping IP addresses—Sites with overlapping private addresses that need to communicate with each other using public addresses or non-overlapping private addresses.

This chapter describes NAT service concepts and the steps required to provision NAT services using the ISC API. The provisioning example includes all steps from creating the inventory to auditing the service deployment.

For more information on NAT provisioning using ISC, refer to the *Cisco IP Solution Center Integrated VPN Management Suite Security User Guide, 4.0*.

This chapter contains the following sections:



# NAT Service Requests

Unlike other ISC services, NAT provisioning does not require a service definition (policy). The service request alone creates the NAT configuration and delivers it to target devices.

A NAT service request consists of **SRPeerIPAddressRanges**, which are the peer IP address ranges for all devices in the service request, and **NatLink**, which defines the device-specific parameters. NatLink parameters include:

- CPE device
- Address overlapping and autotranslation
- NAT address pools

- Dynamic and static address translation parameters
- Device peer IP address ranges
- Traffic redirect parameters and the traffic pattern

See the following example:

```
<objectPath xsi:type="ns1:CIMObjectPath">
              <className xsi:type="xsd:string">ServiceRequestDetails</className>
              <properties xsi:type="ns1:CIMPropertyList"
                        soapenc:arrayType="ns1:CIMProperty[]">
                  <properties>
              <objectPath xsi:type="ns1:CIMObjectPath">
                <className xsi:type="xsd:string">SRPeerIPAddressRanges</className>
                <properties xsi:type="ns1:CIMPropertyList"
                   soapenc:arrayType="ns1:CIMProperty[]">
                  <item xsi:type="ns1:CIMProperty">
                    <name xsi:type="xsd:string">IsExclusion</name>
                    <value xsi:type="xsd:string">true</value>
                  </item>
                  <item xsi:type="ns1:CIMProperty">
                    <name xsi:type="xsd:string">IPAndSubnet</name>
                    <value xsi:type="xsd:string">122.133.144.155/24</value>
                  </item>
                </properties>
              </objectPath>
              <objectPath xsi:type="ns1:CIMObjectPath">
                <className xsi:type="xsd:string">SRPeerIPAddressRanges</className>
                <properties xsi:type="ns1:CIMPropertyList"
                   soapenc:arrayType="ns1:CIMProperty[]">
                  <item xsi:type="ns1:CIMProperty">
                    <name xsi:type="xsd:string">IsExclusion</name>
                    <value xsi:type="xsd:string">false</value>
                  </item>
                  <item xsi:type="ns1:CIMProperty">
                    <name xsi:type="xsd:string">IPAndSubnet</name>
                    <value xsi:type="xsd:string">166.177.188.199/32</value>
                  </item>
                </properties>
              </objectPath>
              <objectPath xsi:type="ns1:CIMObjectPath">
                <className xsi:type="xsd:string">NatLink</className>
                <properties xsi:type="ns1:CIMPropertyList"
                     soapenc:arrayType="ns1:CIMProperty[]">
                  <item xsi:type="ns1:CIMProperty">
                    <name xsi:type="xsd:string">Cpe</name>
                    <value xsi:type="xsd:string">ensw2950-1</value>
                  </item>
                  <item xsi:type="ns1:CIMProperty">
                    <name xsi:type="xsd:string">AddressOverlapping</name>
                    <value xsi:type="xsd:string">true</value>
                  </item>
                  <item xsi:type="ns1:CIMProperty">
                    <name xsi:type="xsd:string">AutoTranslation</name>
                    <value xsi:type="xsd:string">false</value>
                  </item>
                </properties>
                <objectPath xsi:type="ns1:CIMObjectPath">
                  <className xsi:type="xsd:string">NatAddrPool</className>
                  <properties xsi:type="ns1:CIMPropertyList"
                      soapenc:arrayType="ns1:CIMProperty[]">
                    <item xsi:type="ns1:CIMProperty">
                      <name xsi:type="xsd:string">PoolType</name>
                      <value xsi:type="xsd:string">ALTERNATE</value>
```

```
    </item>
    <item xsi:type="ns1:CIMProperty">
      <name xsi:type="xsd:string">Overloading</name>
      <value xsi:type="xsd:string">true</value>
    </item>
    <item xsi:type="ns1:CIMProperty">
      <name xsi:type="xsd:string">PoolName</name>
      <value xsi:type="xsd:string">addrPoolNBI-One</value>
    </item>
  </properties>
</objectPath>
<objectPath xsi:type="ns1:CIMObjectPath">
  <className xsi:type="xsd:string">NatAddrPool</className>
  <properties xsi:type="ns1:CIMPropertyList"
      soapenc:arrayType="ns1:CIMProperty[]">
    <item xsi:type="ns1:CIMProperty">
      <name xsi:type="xsd:string">PoolType</name>
      <value xsi:type="xsd:string">PRIMARY</value>
    </item>
    <item xsi:type="ns1:CIMProperty">
      <name xsi:type="xsd:string">Overloading</name>
      <value xsi:type="xsd:string">false</value>
    </item>
    <item xsi:type="ns1:CIMProperty">
      <name xsi:type="xsd:string">PoolName</name>
      <value xsi:type="xsd:string">addrPoolNBI-Two</value>
    </item>
  </properties>
  <objectPath xsi:type="ns1:CIMObjectPath">
    <className xsi:type="xsd:string">DynamicAddrTranslation</className>
    <properties xsi:type="ns1:CIMPropertyList"
                        soapenc:arrayType="ns1:CIMProperty[]">
    <item xsi:type="ns1:CIMProperty">
      <name xsi:type="xsd:string">FromInterface</name>
      <value xsi:type="xsd:string">inside</value>
    </item>
    <item xsi:type="ns1:CIMProperty">
      <name xsi:type="xsd:string">FromPrefix</name>
      <value xsi:type="xsd:string">168.122.34.1/24</value>
    </item>
      <name xsi:type="xsd:string">FromPrefix</name>
      <value xsi:type="xsd:string">192.100.100.1/24</value>
    </item> -->
    <item xsi:type="ns1:CIMProperty">
      <name xsi:type="xsd:string">ToInterface</name>
      <value xsi:type="xsd:string">outside</value>
    </item>
    <item xsi:type="ns1:CIMProperty">
      <name xsi:type="xsd:string">AlternativeTranslation</name>
      <value xsi:type="xsd:string">false</value>
    </item>
    <item xsi:type="ns1:CIMProperty">
      <name xsi:type="xsd:string">OptionalCommand</name>
      <value xsi:type="xsd:string">20</value>
    </item>
  </properties>
  </objectPath>
</objectPath>
<objectPath xsi:type="ns1:CIMObjectPath">
  <className xsi:type="xsd:string">StaticAddrTranslation</className>
  <properties xsi:type="ns1:CIMPropertyList"
                      soapenc:arrayType="ns1:CIMProperty[]">
  <item xsi:type="ns1:CIMProperty">
    <name xsi:type="xsd:string">FromPort</name>
```

```
        <value xsi:type="xsd:string">8088</value>
      </item>
      <item xsi:type="ns1:CIMProperty">
        <name xsi:type="xsd:string">FromIpAddress</name>
        <value xsi:type="xsd:string">122.168.134.56</value>
      </item>
      <item xsi:type="ns1:CIMProperty">
        <name xsi:type="xsd:string">OptionalCommand</name>
        <value xsi:type="xsd:string">9</value>
      </item>
      <item xsi:type="ns1:CIMProperty">
        <name xsi:type="xsd:string">ToPort</name>
        <value xsi:type="xsd:string">8234</value>
      </item>
      <item xsi:type="ns1:CIMProperty">
        <name xsi:type="xsd:string">ToInterface</name>
        <value xsi:type="xsd:string">7</value>
      </item>
      <item xsi:type="ns1:CIMProperty">
        <name xsi:type="xsd:string">AlternativeTranslation</name>
        <value xsi:type="xsd:string">false</value>
      </item>
      <item xsi:type="ns1:CIMProperty">
        <name xsi:type="xsd:string">FromInterface</name>
        <value xsi:type="xsd:string">4</value>
      </item>
      <item xsi:type="ns1:CIMProperty">
        <name xsi:type="xsd:string">InterfaceTranslation</name>
        <value xsi:type="xsd:string">4</value>
      </item>
      <item xsi:type="ns1:CIMProperty">
        <name xsi:type="xsd:string">TransType</name>
        <value xsi:type="xsd:string">PortBased</value>
      </item>
      <item xsi:type="ns1:CIMProperty">
        <name xsi:type="xsd:string">NetMask</name>
        <value xsi:type="xsd:string">255.255.255.252</value>
      </item>
      <item xsi:type="ns1:CIMProperty">
        <name xsi:type="xsd:string">Protocol</name>
        <value xsi:type="xsd:string">UDP</value>
      </item>
      <item xsi:type="ns1:CIMProperty">
        <name xsi:type="xsd:string">ToIpAddress</name>
        <value xsi:type="xsd:string">10.10.10.2</value>
      </item>
    </properties>
  </objectPath>
  <objectPath xsi:type="ns1:CIMObjectPath">
    <className xsi:type="xsd:string">DevicePeerIPAddressRanges</className>
    <properties xsi:type="ns1:CIMPropertyList"
        soapenc:arrayType="ns1:CIMProperty[]">
      <item xsi:type="ns1:CIMProperty">
        <name xsi:type="xsd:string">IsExclusion</name>
        <value xsi:type="xsd:string">true</value>
      </item>
      <item xsi:type="ns1:CIMProperty">
        <name xsi:type="xsd:string">IPAndSubnet</name>
        <value xsi:type="xsd:string">192.168.134.12/24</value>
      </item>
    </properties>
  </objectPath>
  <objectPath xsi:type="ns1:CIMObjectPath">
    <className xsi:type="xsd:string">TrafficRedirect</className>
```

```
        <properties xsi:type="ns1:CIMPropertyList"
             soapenc:arrayType="ns1:CIMProperty[]">
          <item xsi:type="ns1:CIMProperty">
            <name xsi:type="xsd:string">ToInterface</name>
            <value xsi:type="xsd:string">FastEthernet1</value>
          </item>
          <item xsi:type="ns1:CIMProperty">
            <name xsi:type="xsd:string">FromInterface</name>
            <value xsi:type="xsd:string">FastEthernet2</value>
          </item>
        </properties>
        <objectPath xsi:type="ns1:CIMObjectPath">
          <className xsi:type="xsd:string">TrafficPattern</className>
          <properties xsi:type="ns1:CIMPropertyList"
              soapenc:arrayType="ns1:CIMProperty[]">
            <item xsi:type="ns1:CIMProperty">
              <name xsi:type="xsd:string">Action</name>
              <value xsi:type="xsd:string">Permit</value>
            </item>
            <item xsi:type="ns1:CIMProperty">
              <name xsi:type="xsd:string">SourceNetwork</name>
              <value xsi:type="xsd:string">192.168.134.12/24</value>
            </item>
            <item xsi:type="ns1:CIMProperty">
              <name xsi:type="xsd:string">SourceNetwork</name>
              <value xsi:type="xsd:string">102.108.104.102/24</value>
            </item>
            <item xsi:type="ns1:CIMProperty">
              <name xsi:type="xsd:string">DestinationNetwork</name>
              <value xsi:type="xsd:string">192.168.134.14/24</value>
            </item>
          </properties>
        </objectPath>
      </objectPath>
    </objectPath>
  </objectPath>
</action>
```

# Provisioning Example

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

The complete list of XML examples for NAT are located at:
http://www.cisco.com/univercd/cc/td/doc/product/rtrmgmt/isc/4_0/api/apiref/examples/index.htm

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

## Preparing Inventory

**Step 1** Create devices.

Every network element that ISC manages must be defined as a device in the system. An element is any device from which ISC can collect configuration information.

ISC supports these devices for NAT provisioning:

- Cisco IOS devices (**CiscoRouter**)
- PIX security appliances (**PIX**)

***Table 11-1 Create Devices***

| Operation | className | Required Parameters |
|---|---|---|
| createInstance | • CiscoRouter<br>• PIX | One or more of the following:<br>• ManagementIPAddress<br>• HostName<br>• DomainName<br>• Interface |
| | Interface | • Name<br>• IPAddress<br>• InterfaceEncapType (for **CiscoRouter**)<br>• InterfaceType (for **PIX**) |

**XML Examples:**

- CreateCiscoRouter.xml
- CreatePIX.xml

**Step 2** Create customers (organizations).

A customer is a requestor of VPN services. Each customer can contain multiple customer sites. Each site belongs to only one customer and can contain multiple CPEs.

***Table 11-2 Create Organization***

| Operation | className | Required Parameters |
|---|---|---|
| createInstance | Organization | • Name |

**XML Examples:**

- CreateOrganization.xml

**Step 3** Create sites and assign organizations to them.

***Table 11-3 Create Sites***

| Operation | className | Required Parameters |
|---|---|---|
| createInstance | Site | • Name<br>• Organization |

**XML Examples:**

- CreateSite.xml

**Step 4** Create CPE devices and mark the interfaces. For NAT provisioning, mark device interfaces as inside or outside.

- **NATInside**—connects to an internal network or a network where addresses need to be conserved
- **NATOutside**—connects to the Internet or a corporate network segment

*Table 11-4 Create CPE Devices*

| Operation | className | Required Parameters |
|---|---|---|
| createInstance | Cpe | • Site<br>• Device<br>• ManagementType<br>• Interface |
| | Interface | • Name<br>• NATInside [ true \| false ]<br>• NATOutside [ true \| false ]<br>**Note** Only one of these parameters can be set to **true**. Both can be set to **false**. |

**XML Examples:**

- CreateCpe.xml

# Creating a NAT Service Request

NAT service requests do not require a service definition (policy). A NAT service request consists of the service request peer IP address ranges (the range for all devices in the service request), and the device specific parameters, defined in the **NatLink** object.

*Table 11-5 Create a NAT Service Request*

| Operation | className | Required Parameters |
|---|---|---|
| performBatchOperations | | |
| createInstance | ServiceOrder | • ServiceName<br>• Organization<br>• NumberOfRequests<br>• ServiceRequest |
| | ServiceRequest | • RequestName<br>• Type=Nat<br>• ServiceRequestDetails |
| | ServiceRequestDetails | • SRPeerIPAddressRanges (one or more)<br>• NatLink |
| | NatLink | • Cpe<br>• NatAddrPool<br>• DynamicAddrTranslation<br>• StaticAddrTranslation<br>• DevicePeerIPAddressRanges<br>• TrafficRedirect<br>– TrafficPattern<br>• LinkTemplate (optional)<br>**Note** See the "Templates in a Service Request" section on page 4-9. |

**Note** If you have multiple devices in a service request, traffic between the IP address ranges is No-NAT, by default. To change the default for individual devices, add the **IsExclusion=true** parameter to the **DevicePeerIPAddressRanges** class.

**XML Examples:**

- CreateNatServiceOrder.xml
- CreateNatServiceOrderwTemplate.xml

## Auditing Service Requests

A configuration audit occurs automatically each time you deploy a service request. During this configuration audit, ISC 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 in the service request. 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 see the "Tasks" section on page 3-8.