JUNIPER NETWORKS

Junos® Space

Network Director API

Release
1.5

Published: 2013-10-18

Copyright © 2013, Juniper Networks, Inc.

Juniper Networks, Inc.
1194 North Mathilda Avenue
Sunnyvale, California 94089
USA
408-745-2000
www.juniper.net

Copyright © 2013, Juniper Networks, Inc. All rights reserved.

Juniper Networks, Junos, Steel-Belted Radius, NetScreen, and ScreenOS are registered trademarks of Juniper Networks, Inc. in the United States and other countries. The Juniper Networks Logo, the Junos logo, and JunosE are trademarks of Juniper Networks, Inc. All other trademarks, service marks, registered trademarks, or registered service marks are the property of their respective owners.

Juniper Networks assumes no responsibility for any inaccuracies in this document. Juniper Networks reserves the right to change, modify, transfer, or otherwise revise this publication without notice.

Products made or sold by Juniper Networks or components thereof might be covered by one or more of the following patents that are owned by or licensed to Juniper Networks: U.S. Patent Nos. 5,473,599, 5,905,725, 5,909,440, 6,192,051, 6,333,650, 6,359,479, 6,406,312, 6,429,706, 6,459,579, 6,493,347, 6,538,518, 6,538,899, 6,552,918, 6,567,902, 6,578,186, and 6,590,785.

Junos® Space Network Director API
1.5
Copyright © 2013, Juniper Networks, Inc.
All rights reserved.

The information in this document is current as of the date on the title page.

YEAR 2000 NOTICE

Juniper Networks hardware and software products are Year 2000 compliant. Junos OS has no known time-related limitations through the year 2038. However, the NTP application is known to have some difficulty in the year 2036.

END USER LICENSE AGREEMENT

The Juniper Networks product that is the subject of this technical documentation consists of (or is intended for use with) Juniper Networks software. Use of such software is subject to the terms and conditions of the End User License Agreement ("EULA") posted at http://www.juniper.net/support/eula.html. By downloading, installing or using such software, you agree to the terms and conditions of that EULA.

Table of Contents

Copyright © 2013, Juniper Networks, Inc.

Copyright © 2013, Juniper Networks, Inc.

List of Figures

Copyright © 2013, Juniper Networks, Inc.

Copyright © 2013, Juniper Networks, Inc.

List of Tables

Copyright © 2013, Juniper Networks, Inc.

Copyright © 2013, Juniper Networks, Inc.

About the Documentation

Documentation and Release Notes

To obtain the most current version of all Juniper Networks® technical documentation, see the product documentation page on the Juniper Networks website at http://www.juniper.net/techpubs/.

If the information in the latest release notes differs from the information in the documentation, follow the product Release Notes.

Juniper Networks Books publishes books by Juniper Networks engineers and subject matter experts. These books go beyond the technical documentation to explore the nuances of network architecture, deployment, and administration. The current list can be viewed at http://www.juniper.net/books.

Documentation Conventions

Table 1 on page x defines notice icons used in this guide.

Copyright © 2013, Juniper Networks, Inc.

Table 1: Notice Icons

Icon	Meaning	Description
	Informational note	Indicates important features or instructions.
	Caution	Indicates a situation that might result in loss of data or hardware damage.
	Warning	Alerts you to the risk of personal injury or death.
	Laser warning	Alerts you to the risk of personal injury from a laser.

Table 2 on page x defines the text and syntax conventions used in this guide.

Table 2: Text and Syntax Conventions

Convention	Description	Examples
Bold text like this	Represents text that you type.	To enter configuration mode, type the **configure** command: user@host> **configure**
`Fixed-width text like this`	Represents output that appears on the terminal screen.	`user@host>` **show chassis alarms** `No alarms currently active`
Italic text like this	• Introduces or emphasizes important new terms. • Identifies guide names. • Identifies RFC and Internet draft titles.	• A policy *term* is a named structure that defines match conditions and actions. • *Junos OS CLI User Guide* • RFC 1997, *BGP Communities Attribute*
Italic text like this	Represents variables (options for which you substitute a value) in commands or configuration statements.	Configure the machine's domain name: [edit] root@# **set system domain-name** *domain-name*
Text like this	Represents names of configuration statements, commands, files, and directories; configuration hierarchy levels; or labels on routing platform components.	• To configure a stub area, include the **stub** statement at the **[edit protocols ospf area area-id]** hierarchy level. • The console port is labeled **CONSOLE**.
< > (angle brackets)	Enclose optional keywords or variables.	**stub <default-metric** *metric***>;**

Copyright © 2013, Juniper Networks, Inc.

Table 2: Text and Syntax Conventions *(continued)*

Convention	Description	Examples
| (pipe symbol)	Indicates a choice between the mutually exclusive keywords or variables on either side of the symbol. The set of choices is often enclosed in parentheses for clarity.	**broadcast | multicast** **(*string1* | *string2* | *string3*)**
# (pound sign)	Indicates a comment specified on the same line as the configuration statement to which it applies.	**rsvp { # Required for dynamic MPLS only**
[ ] (square brackets)	Enclose a variable for which you can substitute one or more values.	**community name members [*community-ids*]**

Indention and braces ({ })	Identify a level in the configuration hierarchy.	[edit]
		routing-options {
		  static {
		  route default {
		  nexthop *address*;
		  retain;
		  }
		  }
		}

; (semicolon)	Identifies a leaf statement at a configuration hierarchy level.	
GUI Conventions		
Bold text like this	Represents graphical user interface (GUI) items you click or select.	• In the Logical Interfaces box, select **All Interfaces**. • To cancel the configuration, click **Cancel**.
> (bold right angle bracket)	Separates levels in a hierarchy of menu selections.	In the configuration editor hierarchy, select **Protocols>Ospf**.

Documentation Feedback

We encourage you to provide feedback, comments, and suggestions so that we can improve the documentation. You can send your comments to **techpubs-comments@juniper.net**, or fill out the documentation feedback form at https://www.juniper.net/cgi-bin/docbugreport/ . If you are using e-mail, be sure to include the following information with your comments:

- Document or topic name
- URL or page number
- Software release version (if applicable)

Requesting Technical Support

Technical product support is available through the Juniper Networks Technical Assistance Center (JTAC). If you are a customer with an active J-Care or JNASC support contract,

Copyright © 2013, Juniper Networks, Inc.

or are covered under warranty, and need post-sales technical support, you can access our tools and resources online or open a case with JTAC.

- JTAC policies—For a complete understanding of our JTAC procedures and policies, review the *JTAC User Guide* located at http://www.juniper.net/us/en/local/pdf/resource-guides/7100059-en.pdf.
- Product warranties—For product warranty information, visit http://www.juniper.net/support/warranty/.
- JTAC hours of operation—The JTAC centers have resources available 24 hours a day, 7 days a week, 365 days a year.

Self-Help Online Tools and Resources

For quick and easy problem resolution, Juniper Networks has designed an online self-service portal called the Customer Support Center (CSC) that provides you with the following features:

- Find CSC offerings: http://www.juniper.net/customers/support/
- Search for known bugs: http://www2.juniper.net/kb/
- Find product documentation: http://www.juniper.net/techpubs/
- Find solutions and answer questions using our Knowledge Base: http://kb.juniper.net/
- Download the latest versions of software and review release notes: http://www.juniper.net/customers/csc/software/
- Search technical bulletins for relevant hardware and software notifications: https://www.juniper.net/alerts/
- Join and participate in the Juniper Networks Community Forum: http://www.juniper.net/company/communities/
- Open a case online in the CSC Case Management tool: http://www.juniper.net/cm/

To verify service entitlement by product serial number, use our Serial Number Entitlement (SNE) Tool: https://tools.juniper.net/SerialNumberEntitlementSearch/

Opening a Case with JTAC

You can open a case with JTAC on the Web or by telephone.

- Use the Case Management tool in the CSC at http://www.juniper.net/cm/.
- Call 1-888-314-JTAC (1-888-314-5822 toll-free in the USA, Canada, and Mexico).

For international or direct-dial options in countries without toll-free numbers, see http://www.juniper.net/support/requesting-support.html.

Copyright © 2013, Juniper Networks, Inc.

PART 1

Overview

Copyright © 2013, Juniper Networks, Inc.

Copyright © 2013, Juniper Networks, Inc.

CHAPTER 1

Network Director API Overview

Overview of Network Director API

The Junos Space Network Director API application runs on the Junos Space Network Management Platform, and is exposed by the Network Director orchestration services.

The Network Director API is a set of Representational State Transfer (REST) APIs that enable network management functions, including:

- Provisioning of secure multitenant networks in a shared network infrastructure
- Automation of tenant services in the data center
- Support for Layer 2 and Layer 3 security services
- Provision of a single point of integration with external cloud and data center orchestration tools

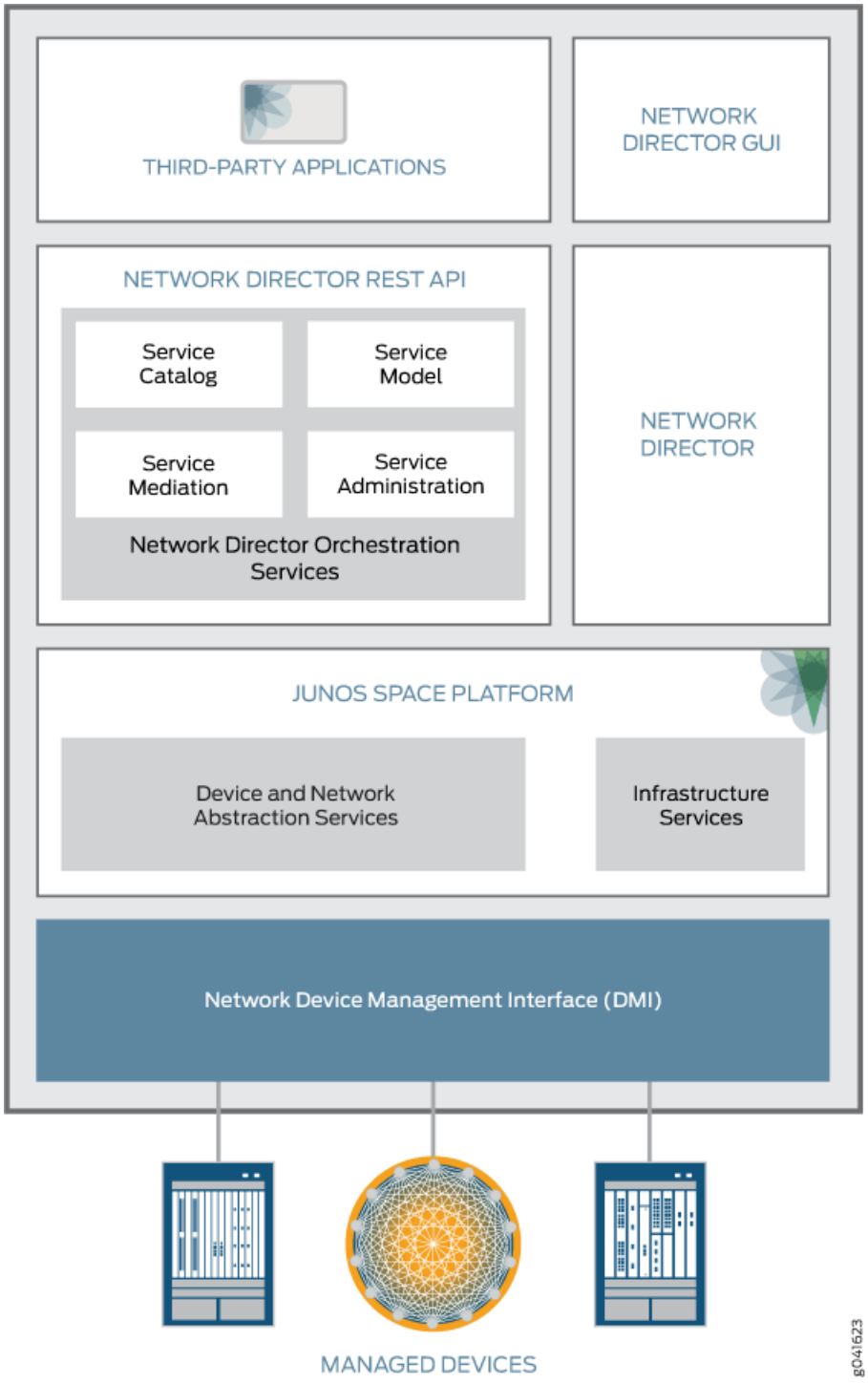
This topic describes:

Network Director API Architecture

Figure 1 on page 4 shows the high-level system architecture of the Network Director API in the Junos Space platform.

Copyright © 2013, Juniper Networks, Inc.

Figure 1: Network Director API Architecture



Supported Devices and Services

Network Director orchestration services and API support the following Juniper Networks devices:

Copyright © 2013, Juniper Networks, Inc.

- EX Series switches
- MX Series routers
- QFX3500 and QFX3600 switches
- QFabric systems
- SRX Series Services Gateways

The type of network services supported depends on the device and the topology of the network.

Table 3 on page 5 describes the supported devices, topologies, and services.

Table 3: Supported Devices, Topologies, and Services

Supported Devices and Topologies	Services	Configuration
• EX Series switches—EX4200 and EX4550 switches • QFabric systems—QFX 3000-G and QFX3000-M QFabric systems • QFX Series—QFX3500 and QFX3600 switches	Layer 2	VLANs on the EX Series switch, QFX Series, or QFabric system
• EX Series switch or QFX Series connected to an MX Series router • QFabric system connected to an MX Series router	• Layer 2 • Layer 2 and Layer 3 • Layer 2 and Layer 3 with Internet access	• VLANs on the EX Series switch, QFX Series, or QFabric system • Layer 3 interfaces on the MX Series router • BGP static configuration on the MX Series router
• EX Series switch or QFX Series connected to an SRX Services Gateway • QFabric system connected to an SRX Services Gateway	• Layer 2 • Layer 2 and Layer 3 • Layer 2 and Layer 3 with firewall	• VLANs on the EX Series switch, QFX Series, or QFabric system • Layer 3 interfaces on the SRX Services Gateway • Security policies on the SRX Services Gateway

Network Director API Requirements

Before you can use the Network Director API, you must first install the following software:

- Junos Space Release 13.1P5 (when available)
- Network Director API Release 1.5 or Network Director Release 1.5

NOTE: **We recommend that you install Network Director API Release 1.5.**

- REST HTTP client, which can be browser-based or script-based
- (Optional) Python 3 or later programming language software for running the sample API scripts included in the Network Director API software

Copyright © 2013, Juniper Networks, Inc.

Network Director Orchestration Services Overview

The Network Director orchestration services provide:

- Support for a variety of predefined physical topology configurations, ranging from a single device to a complete complex data center infrastructure.
- Static configuration management that enables the automatic download of static configurations to managed devices.
- Flexible and extensible network service APIs that enable the definition of endpoints, connectivity between endpoints in Layer 2 and Layer 3, connectivity between Layer 2 and Layer 3, and policy applications, including those for security.

Network Director Orchestration Services Components

Network Director orchestration services are based on the Network as a Service (NaaS) model-based management system. NaaS consists of two logical components: NaaS service and NaaS domain.

- The NaaS service runs in the Junos Space infrastructure and manages the NaaS domain in such a way that the NaaS domain appears to network administrators and tenants as a single device with a set of revenue ports. Revenue ports are data ports on Juniper Networks devices. In a typical data center environment, compute servers are attached to revenue ports through which NaaS service is provided. The Network Director API operations are performed using the NaaS service.
- The NaaS domain represents the network infrastructure that the NaaS service manages. All elements, both physical and logical, in the domain are represented internally as instances of model classes in the Network Director orchestration services repository. The state of the elements in the repository is synchronized with the state in the NaaS domain by way of read-and-write activities such as discovery, configuration, and other events that occur in the management network.

In Network Director orchestration services provide network administrators with administrative and configuration privileges for performing network operations, including:

- Configuring the network infrastructure
- Performing basic create, replace, update, delete (CRUD) operations of tenants
- Activating and deactivating NaaS service

Network Director API Overview

The Network Director orchestration services expose a set of REST APIs that is accessible through standard Web-based HTTP requests. You use the APIs to send requests to NaaS service resources to perform HTTP CRUD operations. The Network Director API is idempotent so that multiple requests to the same NaaS resources do not cause duplicate updates or provisioning errors.

Copyright © 2013, Juniper Networks, Inc.

The NaaS resources are represented in both JavaScript Object Notification (JSON) and Extensible Markup Language (XML) formats, and requests sent to the resource Universal Resource Identifier (URI) can be in either JSON or XML format.

The NaaS resources or services are grouped into classes by the type of services. For example:

- ConnectivityGroups—Provides NaaS service for connectivity groups consisting of endpoints.
- L2ConnectivityServices—Provides Layer 2 connectivity services.
- L3ConnectivityServices—Provides Layer 3 connectivity services.
- SecurityPolicies—Provides NaaS service to enable security policies.

Each class might contain resources. Examples of resource URIs are:

- https://*host-ip*/api/juniper/nd/tenants/*tenant-id*/naas-services/naas-service-id/connectivity-groups/connectivity-group-id/ports
- https://*host-ip*/api/juniper/nd/tenants/*tenant-id*/naas-services/naas-service-id/l2-connectivity-services
- https://*host-ip*/api/juniper/nd/tenants/*tenant-id*/naas-services/naas-service-id/l3-connectivity-services
- https://*host-ip*/api/juniper/nd/tenants/*tenant-id*/naas-services/naas-service-id/l3-connectivity-services/l3-connectivity-services-id/security-policies

Network Director API Setup Overview

After you install the Network Director API software, the only NaaS service that is available for use is the service for importing the physical network topology of the NaaS domain. Before you can use the full set of NaaS service resources with the API, you must initialize NaaS service and import static configurations to your managed devices.

You initialize the NaaS service by importing the physical topology and populating the NaaS service repository with the topology. The NaaS service then performs a discovery of the devices in the NaaS domain. When the device discovery is complete, all devices in the topology are ready for configuration.

After the NaaS service is initialized, you must configure all managed devices. You must push the static configuration files of supported features to the devices. Static configurations are typically stable configurations, but they could be updated in the future if necessary. For example, to enable BGP on a device, you must push the static configuration file for BGP to the device. Pushing static configuration files to your devices enables them to be managed and configured using the Network Director API.

After you have set up the NaaS service for the Network Director API, all NaaS service resources that are supported for your network topology and devices are available for use.

Copyright © 2013, Juniper Networks, Inc.

For more information about the Network Director API setup, see "Setting Up the Network Director API" on page 15.

Common CRUD Operations

Requests are sent to the URI of the resource. For each resource, you can perform one or more of the four common CRUD operations. (A CRUD operation is also called a *resource method*.) Interactions with each resource follow specific formats, depending on the operation.

Table 4 on page 8 describes the common CRUD operations.

Table 4: Common CRUD Operations

Operation	Description	Request Format	Response Format

Create (POST)	Creates an instance of the resource by sending an HTTP POST request to the designated URI for that resource type.	`POST <resourceURI> HTTP/1.1`
		
		`URL = https://<server-ip>/api/juniper/nd/<resource>`
		
		`Content-Type = application/vnd/juniper/nd/<resource>+(json|xml);version=1;charset=UTF-8`
		
		`Accept = application/vnd.juniper.nd.<resource>+(json|xml);version=1`
		
		`<body of request>`
			`HTTP/1.1 200 OK`
			
			`https://<server-ip>/api/juniper/nd/<resource>/<resource-id>`

Read (GET)	Retrieves the representation of a resource by sending an HTTP GET request to the resource URI.	`GET <resourceURI> HTTP/1.1`
		
		`URL = https://<server-ip>/api/juniper/nd/<resource>`
		
		`Content-Type = application/vnd/juniper/nd/<resource>+(json|xml);version=1;charset=UTF-8`
		
		`Accept = application/vnd.juniper.nd.<resource>+(json|xml);version=1`
		
		`<body of request>`
			`HTTP/1.1 200 OK`
			
			`<body of response>`

Copyright © 2013, Juniper Networks, Inc.

Table 4: Common CRUD Operations *(continued)*

Operation	Description	Request Format	Response Format

Update (PUT)	Updates the representation of a resource by sending an HTTP PUT request to the resource URI.
	
	NOTE: If the server detects that the request attempts to update a read-only attribute, the server ignores the update request, but does not generate an error.
		`PUT <resourceURI> HTTP/1.1`
		
		`URL = https://<server-ip>/api/juniper/nd/<resource>`
		
		`Content-Type = application/vnd/juniper/nd/<resource>+(json|xml);version=1;charset=UTF-8`
		
		`Accept = application/vnd.juniper.nd.<resource>+(json|xml);version=1`
		
		`<body of request>`
			`HTTP/1.1 200 OK`

Delete (DELETE)	Deletes a resource by sending an HTTP DELETE request to the resource URI.	`DELETE <resourceURI> HTTP/1.1`
		
		`URL = https://<server-ip>/api/juniper/nd/<resource>`
		
		`Content-Type = application/vnd/juniper/nd/<resource>+(json|xml);version=1;charset=UTF-8`
		
		`Accept = application/vnd.juniper.nd.<resource>+(json|xml);version=1`
		
		`<body of request>`
			`HTTP/1.1 204 OK`
			
			`https://<server-ip>/api/juniper/nd/<resource>/<resource-id>`

API Reference Documentation

For more information about using Network Director APIs, see the API reference documentation (**apiDocumentation.zip**).

For information about accessing the API reference documentation, see "Locating API Reference Information" on page 29.

Related Documentation

- Setting Up the Network Director API on page 15
- Network Director API Setup Sample Files on page 17
- Sample API Scripts Overview on page 11
- Locating API Reference Information on page 29

Copyright © 2013, Juniper Networks, Inc.

Copyright © 2013, Juniper Networks, Inc.

CHAPTER 2

Sample API Scripts

Sample API Scripts Overview

The Network Director API software package includes sample Python API scripts with examples of API requests. To run the sample scripts, you must have the Python 3 programming language software installed on your server.

The Python scripts are available at:

/usr/share/doc/jmp-ndos-1.5/sampleScripts.zip

The scripts are organized in separate directories by the type of NaaS service. The scripts include the promptUser method that provides user prompts for all interactions by displaying available options and applicable input fields.

Table 5 on page 11 lists the available sample API scripts:

Table 5: Sample API Scripts

Script Directory and Filename	Description
CGCRUDScripts/CGCRUD.py	Contains API methods used to perform the create, read, update, and delete (CRUD) HTTP operations on connectivity group objects.
ImportTopologyCRUDScripts/ImportTopology.py	Contains API methods used to read and import the topology file and create the topology in the NaaS service repository.
L2CRUDScripts/L2CRUD.py	Contains API methods used to perform CRUD operations on Layer 2 connectivity NaaS service objects.
L2FacadeScripts/L2Facade.py	Contains API methods used to create the Layer 2 connectivity service using the Facade implementation. This script uses the topology file in the same directory to create the service.
L3CRUDScripts/L3CRUD.py	Contains API methods used to perform CRUD operations on Layer 3 connectivity NaaS service objects.
L3FacadeScripts/L3Facade.py	Contains API methods used to create the Layer 3 connectivity service using the Facade implementation. This script uses the topology file in the same directory to create the service.

Copyright © 2013, Juniper Networks, Inc.

Table 5: Sample API Scripts *(continued)*

Script Directory and Filename	Description
NaasCRUDScripts/NaasCRUD.py	Contains API methods used to perform CRUD operations on all NaaS service objects.
PortCRUDScripts/PortCRUD.py	Contains API methods used to perform CRUD operations on port objects.
SecurityPoliciesCRUDScripts/SecurityPoliciesCRUD.py	Contains API methods used to perform CRUD operations on all security policy objects.
ServiceCatalog/ServiceCatalog.py	Contains the API method used to obtain the service catalog object by the NaaS domain ID.
SPRuleCRUDScripts/SPRuleCRUDScripts.py	Contains API methods used to perform CRUD operations on all security policy rule objects.
TenantCRUDScripts/TenantCRUD.py	Contains API methods used to perform CRUD operations on all tenant objects.

Related Documentation

- Overview of Network Director API on page 3
- Setting Up the Network Director API on page 15

Copyright © 2013, Juniper Networks, Inc.

PART 2

Configuration

Copyright © 2013, Juniper Networks, Inc.

Copyright © 2013, Juniper Networks, Inc.

CHAPTER 3

Network Director API Settings

Setting Up the Network Director API

After you install the Network Director API, invoke the API for importing the physical network topology of the NaaS domain. Before you can use the full set of NaaS service resources with the API, you must initialize the NaaS domain and import static configurations to your managed devices.

After you initialize the NaaS service, you can enable features on the devices in the NaaS domain by importing (pushing) static configurations of each feature to the devices.

Initializing the NaaS Domain

As the network administrator, you initialize the NaaS domain by importing the physical topology. The Network Director API then performs a discovery of the devices in the NaaS domain. When the device discovery is complete, all devices in the topology are ready for configuration.

Before you begin, make sure that:

- The network and devices are running.
- Network Director API 1.5 is installed.

To initialize the Network Director API by importing the NaaS domain topology:

1. Prepare an XML file containing the physical topology of your network. This topology has the same components and devices as the NaaS domain. Include the required information such as system name (for each device), revenue port interface names, and linked interface information.

   For more information, see "Network Director API Setup Sample Files" on page 17.

2. In your REST API client, send an HTTP POST request by using the following input:

Copyright © 2013, Juniper Networks, Inc.

- **Content-Type = application/vnd.juniper.nd.import-topology-xml-request+xml;version=1;charset=UTF-8**
- **URL=https://*host-ip*/api/juniper/nd/import-topology-xml**
- Include the content of your topology file in the POST content.

Importing a topology results in one of three conditions:

- SUCCESS—All the details in topology are valid, and the corresponding components are added and updated successfully in the Network Director API repository. A NaaS domain is created and its ID returned. You may proceed to the next task of pushing configuration files to devices.
- PARTIAL_SUCCESS—The topology file contains errors for one or more systems. Issues may include an invalid revenue port name or number, or an unreachable device. A NaaS domain is created and its ID returned.

  NOTE: **If the return condition is PARTIAL_SUCCESS, you may proceed to the next task of pushing configuration files to devices. However, if the topology contains an error for a device or port, you might not be able to configure that component in the future. We recommend that you correct or remove the error and reimport the topology file.**

- FAILURE—The topology file contains severe errors, resulting in cancellation of the topology import. A NaaS domain is not created. You must correct the errors and reimport the topology file.

Pushing Static Configuration Files to Devices

After the physical topology is imported into the Network Director API repository and the NaaS domain is initialized, all managed devices specified in the topology must be configured. We recommend that you push the static configuration files of supported features to the devices at this time. Static configurations are typically stable configurations, but they could be updated in the future if necessary. For example, to enable BGP on a device, you must push the static configuration file for BGP to the device. Pushing static configuration files to your devices enables them to be managed and configured by using the Network Director API.

To push a static configuration file to your devices:

1. Prepare a static configuration file for each feature, for example, BGP.

   For more information, see "Network Director API Setup Sample Files" on page 17.

2. In your REST API client, send an HTTP POST request by using the following input:

   - **Content-Type= application/vnd.juniper.nd.configuration-push+xml;version=1;charset=UTF-8**
   - **URL=https://*server-ip*/api/juniper/nd/configuration-push/*device host name***
   - Include the content of your static configuration file in the POST content.

Copyright © 2013, Juniper Networks, Inc.

3. Repeat Step 2 for each device in your topology.

Related Documentation

- Overview of Network Director API on page 3
- Network Director API Setup Sample Files on page 17
- Sample API Scripts Overview on page 11

Network Director API Setup Sample Files

After you install the Network Director API software, the only NaaS service that is available for use is the service for importing the physical network topology of the NaaS domain. Before you can use the full set of NaaS service resources with the API, you must initialize the NaaS service and import static configurations to your managed devices.

Before you import the NaaS domain physical topology and static configurations to devices, prepare a topology file and one or more static configuration files.

This topic provides the following sample files:

- Sample Physical Topology File on page 17
- Sample Static BGP Configuration Snippet on page 19

Sample Physical Topology File

The following output shows a sample physical topology file in XML format.

```
<NaasDomains>
   <NaasDomain name="Naas-Domain-Name">
      <topology>
        <systems>
          <system name="ex4-host-name" mgmtIp="192.168.1.0" userId="root" password="SecretPassword">
            <capabilities>
              <capability name="SWITCHING" />
            </capabilities>
            <revenueport_ifds>
              <revenueport_ifd name="ge-0/0/3"/>
              <revenueport_ifd name="ge-0/0/22"/>
              <revenueport_ifd name="ge-0/0/23"/>
              <revenueport_ifd name="ge-0/0/24"/>
            </revenueport_ifds>
          </system>
          <system name="srx-host-name" mgmtIp="10.94.45.185" userId="root" password="password">
              <capabilities>
                <capability name="SECURITY" />
              </capabilities>
              <revenueport_ifds>
                <revenueport_ifd name="ge-0/0/1"/>
                <revenueport_ifd name="ge-0/0/2"/>
                <revenueport_ifd name="ge-1/0/3"/>
                <revenueport_ifd name="ge-0/0/4"/>
                <revenueport_ifd name="ge-0/0/5"/>
                <revenueport_ifd name="ge-0/0/6"/>
              </revenueport_ifds>
          </system>
        </systems>
          <linked_interfaces>
```

Copyright © 2013, Juniper Networks, Inc.

```
            <ifd name="ge-1/0/3">
              <peer name="ge-0/0/23">
                <hostedOn name="ex4-host-name"/>
              </peer>
              <hostedOn name="srx-host-name"/>
            </ifd>
          </linked_interfaces>
      </topology>
   </NaasDomain>
</NaasDomains>
```

Table 6 on page 18 describes the NaaS domain physical topology information that you must provide in the topology file.

Table 6: Physical Topology Input Fields

Field Name	Description
NaasDomain name	Name of the NaaS domain in the topology. Only one NaaS domain is supported.
NaaS Domain Components	
system name	System name of each device managed by the NaaS domain. For each system name, provide the following system information: • Management IP address (mgmtIp) • User ID (userId) of the system administrator • Password (password) for the user ID NOTE: Only one device with switching capability is supported in each NaaS domain.
System Components	
capability name	Capability you enable for each system (managed device). Valid values are: • ROUTING • SECURITY • SWITCHING NOTE: Only one device with switching capability is supported in each NaaS domain.
revenueport_ifd name	Interface name that identifies a revenue port in a system; for example, ge-0/0/3. Revenue ports are attachment points that connect to physical servers.
Linked Interfaces Components (Optional for topologies with a single device)	
ifd name	Name of an interface that is connected to a peer interface. An ifd name must be paired with a peer name.
peer name	Name of the peer interface.

Copyright © 2013, Juniper Networks, Inc.

Table 6: Physical Topology Input Fields *(continued)*

Field Name	Description
hostedOn name	Hostname of an interface. Each linked interface must have a hostname.

Sample Static BGP Configuration Snippet

The following output shows a portion of a sample static BGP configuration file:

```
<configuration>
  <interfaces>
     <interface>
        <name>ge-1/1/0</name>
           <unit>
              <name>0</name>
              <family>
                 <inet>
                    <address>
                       <name>10.20.99.1/30</name>
                    </address>
                 </inet>
              </family>
           </unit>
     </interface>
  </interfaces>
  <routing-options>
     <autonomous-system>
        <as-number>69901</as-number>
     </autonomous-system>
  </routing-options>
  <protocols>
     <bgp>
        <group>
           <name>ext</name>
           <type>external</type>
           <export>adv-direct-to-bgp</export>
           <peer-as>69902</peer-as>
           <neighbor>
              <name>10.20.99.2</name>
           </neighbor>
        </group>
     </bgp>
  </protocols>
  <policy-options>
     <policy-statement>
        <name>adv-direct-to-bgp</name>
        <term>
           <name>A</name>
           <from>
              <protocol>direct</protocol>
              <protocol>static</protocol>
           </from>
           <then>
              <accept/>
           </then>
        </term>
     </policy-statement>
```

Copyright © 2013, Juniper Networks, Inc.

```
  </policy-options>
</configuration>
```

Related Documentation

- Overview of Network Director API on page 3
- Setting Up the Network Director API on page 15
- Sample API Scripts Overview on page 11

Copyright © 2013, Juniper Networks, Inc.

CHAPTER 4

Example of NaaS Service Request

Example: Creating and Activating a NaaS Service Request

This example shows how to create a Network as a Service (NaaS) service request for a tenant in a multitier application network managed by the Juniper Networks Network Director application. A multitier application network contains logical entities or tier groups that are belong in different VLANs. In this example, there are three tier groups, each providing a specific function. The application tier group provides network applications support, the Web tier group provides Internet access, and the client tier group provides client support.

The tier groups are typically connected to different physical compute servers and virtual machines (VMs). In turn, the servers and VMs are connected to devices (such as an MX router) that are managed by the Network Director. The NaaS service is provided to the servers and VMs through the revenue ports on the managed devices.

Requirements

This example uses the following hardware and software components:

- Hardware components:
  - An MX Series router
  - A QFabric system
  - An SRX Series Services Gateway
  - Two compute servers and four VMs
- Software components:
  - Junos Space Release 13.1P5
  - Network Director API Release 1.5

Copyright © 2013, Juniper Networks, Inc.

- REST HTTP client software

Before you begin to create the multitier application NaaS request, be sure that:

- Network Director API Release 1.5 is installed and operating.
- NaaS service is initialized (the physical network topology has been imported into the NaaS service repository and the NaaS domain is created).
- Static configurations are pushed to network devices.

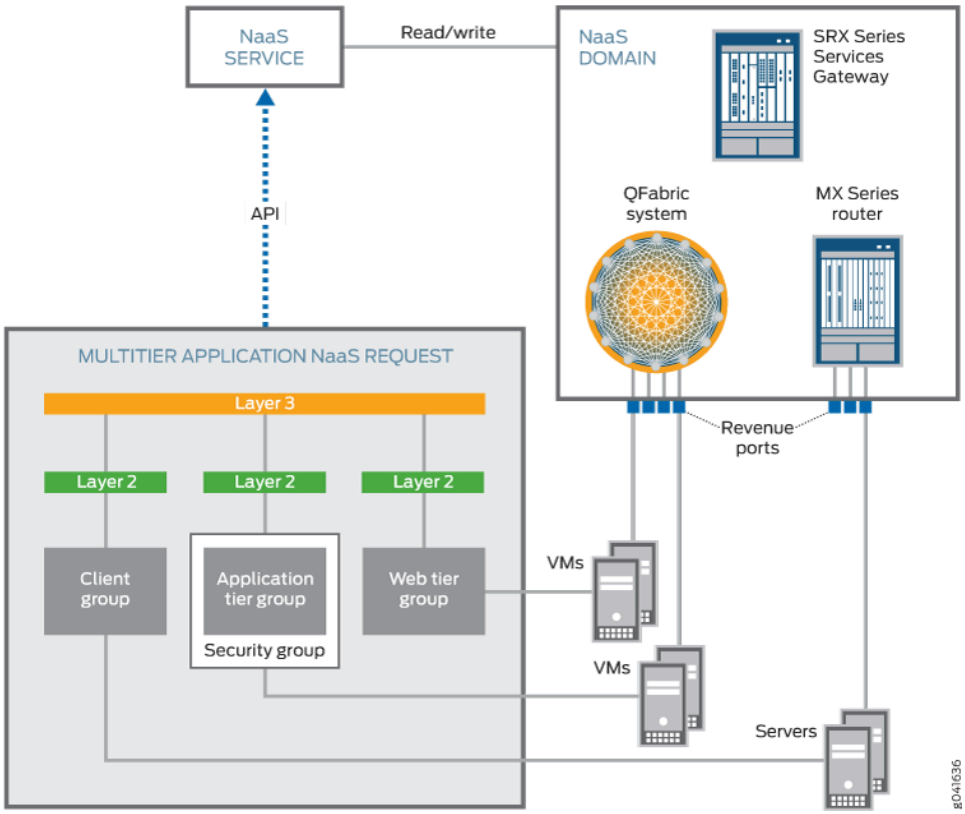
Overview

In this example, the multitier application network contains three tier groups. Each group belongs to a VLAN, and is connected to a physical server or a VM. To configure and manage a multitier application network using the Network Director API, you must send NaaS service requests to create resources for connectivity groups, Layer 2 and Layer 3 connectivity services, security policies, and security policy rules for the tier groups. Once the resources are created, you activate the NaaS services.

Topology

Figure 2 on page 22 shows the topology of a multitier application network.

Figure 2: Multitier Application Network Topology



This example uses the following physical components:

Copyright © 2013, Juniper Networks, Inc.

- A QFabric system
- An MX Series router
- An SRX Series Services Gateway
- Two physical servers
- Four VMs (running on two other physical servers)

This example uses the following logical components:

- Web tier group
- Client tier group
- Application tier group

Creating a Multitier Application NaaS Request

From your REST HTTP client, perform the following tasks.

NOTE: The IP address used in this example is that of the host server on which the Network Director API software is installed.

Creating General Resources

Step-by-Step Procedure

This section describes the steps for creating general resources.

1. Send a POST request to create a tenant resource (if one is not already created).
   - Content type = application/vnd.juniper.nd.tenant+xml;version=1;charset=UTF-8
   - URL = https://172.68.52.10/api/juniper/nd/tenants
   - Request parameters: name="Tenant A"
   - Response parameters: tenantID="tenantA"
2. Send a POST request to create a NaaS service request resource that represents the multitier application network:
   - Content type = application/vnd.juniper.nd.naas-services+xml;version=1;charset=UTF-8
   - URL = https://172.68.52.10/api/juniper/nd/tenants/{tenant-id}/naas-services

Copyright © 2013, Juniper Networks, Inc.

- Request parameters: name="Multitier App Network", naasDomainID="Multitier Domain1"
- Response parameters: naasServiceID="multitierAppNetwork"

Creating the Connectivity Group and Port Resources

Step-by-Step Procedure

This section describes the steps for creating a connectivity group and port resources for each connectivity group.

1. Send a POST request to create a ConnectivityGroup resource that corresponds to the Web tier group:
   - Content type = application/vnd.juniper.nd.connectivity-groups+xml;version=1;charset=UTF-8
   - URL = https://172.68.52.10/api/juniper/nd/tenants/{tenant-id}/naas-services/{naas-service-id}/connectivity-groups
   - Request parameters: group name="Web Tier Group"
   - Response parameters: connectivityGroupID="webTierGroup"
2. Send a POST request to create a port resource for webTierGroup that represents the attachment point for the host VM of the Web tier group:
   - Content type = application/vnd.juniper.nd.ports+xml;version=1;charset=UTF-8
   - URL = https://172.68.52.10/api/juniper/nd/tenants/{tenant-id}/naas-services/{naas-service-id}/connectivity-groups/{connectivity-group-id}/ports
   - Request parameters: port name="xe-0/0/1"
   - Response parameters: portID="webServer1-port"
3. Send a POST request to create a ConnectivityGroup resource for the client group:
   - Content type = application/vnd.juniper.nd.connectivity-groups+xml;version=1;charset=UTF-8
   - URL = https://172.68.52.10/api/juniper/nd/tenants/{tenant-id}/naas-services/{naas-service-id}/connectivity-groups
   - Request parameters: group name="Client Group"
   - Response parameters: connectivityGroupID="clientGroup"
4. Send a POST request to create a port resource for clientGroup that represents the attachment point for the host server of the client group:
   - Content type = application/vnd.juniper.nd.ports+xml;version=1;charset=UTF-8
   - URL = https://172.68.52.10/api/juniper/nd/tenants/{tenant-id}/naas-services/{naas-service-id}/connectivity-groups/{connectivity-group-id}/ports

Copyright © 2013, Juniper Networks, Inc.

- Request parameters: port name="xe-0/0/1"
- Response parameters: portID="Client-port"

5. Send a POST request to create a ConnectivityGroup resource for the application tier group:
   - Content type = application/vnd.juniper.nd.connectivity-groups+xml;version=1;charset=UTF-8
   - URL = https://172.68.52.10/api/juniper/nd/tenants/{tenant-id}/naas-services/{naas-service-id}/connectivity-groups
   - Request parameters: group name="App Tier Group"
   - Response parameters: connectivityGroupID="appTierGroup"
6. Send a POST request to create a port resource for appTierGroup that represents the attachment point for the host VM of the application tier group:
   - Content type = application/vnd.juniper.nd.ports+xml;version=1;charset=UTF-8
   - URL = https://172.68.52.10/api/juniper/nd/tenants/{tenant-id}/naas-services/{naas-service-id}/connectivity-groups/{connectivity-group-id}/ports
   - Request parameters: port name="xe-0/0/1"
   - Response parameters: portID="webServer2-port"

Creating the Layer 2 and Layer 3 Connectivity Services

Step-by-Step Procedure

This section describes the steps for creating Layer 2 and Layer 3 connectivity services for each connectivity group.

1. Send a POST request to create an L2ConnectivityService resource for the Web tier group:
   - Content type = application/vnd.juniper.nd.l2-connectivity-services+xml;version=1;charset=UTF-8
   - URL = https://172.68.52.10/api/juniper/nd/tenants/{tenant-id}/naas-services/{naas-service-id}/l2-connectivity-services
   - Request parameters: name="Web Tier Subnet"", connectivityGroupID="webTierGroup", subnet="192.10.2.1/25"
   - Response parameters: L2ConnectivityServiceID="webTierSubnet"
2. Send a POST request to create an L2ConnectivityService resource for the application tier group:
   - Content type = application/vnd.juniper.nd.l2-connectivity-services+xml;version=1;charset=UTF-8
   - URL = https://172.68.52.10/api/juniper/nd/tenants/{tenant-id}/naas-services/{naas-service-id}/l2-connectivity-services

Copyright © 2013, Juniper Networks, Inc.

- Request parameters: name=”App Tier Subnet””, connectivityGroupID=”appTierGroup”, subnet=”192.10.2.2/25”
- Response parameters: L2ConnectivityServiceID=”appTierSubnet”

3. Send a POST request to create an L2ConnectivityService resource for the client group:
   - Content type = application/vnd.juniper.nd.l2-connectivity-services+xml;version=1;charset=UTF-8
   - URL https://172.68.52.10/api/juniper/nd/tenants/{tenant-id}/naas-services/{naas-service-id}/l2-connectivity-services
   - Request parameters: name=”Client Subnet””, connectivityGroupID=”clientGroup”, subnet=”192.10.2.15/24”
   - Response parameters: L2ConnectivityServiceID=”clientSubnet”
4. Create an L3ConnectivityService resource to provide routing among the Web tier, application tier, and client group VLAN subnets:
   - Content type = application/vnd.juniper.nd.l3-connectivity-services+xml;version=1;charset=UTF-8
   - URL = https://172.68.52.10/api/juniper/nd/tenants/{tenant-id}/naas-services/{naas-service-id}/l3-connectivity-services
   - Request parameters: name=”Multi Tier L3”, underlyings= { ”webTierSubnet”, “appTierSubnet”, “clientSubnet” }
   - Response Parameters: L3ConnectivityServiceID=”multiTierL3”

Creating Security Policy and Security Policy Rule Resources

Step-by-Step Procedure

This section describes the steps for creating security policy and security policy rule resources for defining security rules for traffic sent to appTierGroup.

1. Send a POST request to create a security policy resource:
   - Content type = application/vnd.juniper.nd.security-policies+xml;version=1;charset=UTF-8
   - URL = https://172.68.52.10/api/juniper/nd/tenants/{tenantid}/naas-services/{naas-service-id}/l3-connectivity-services/{l3-connectivity-service-id}/security-policies
   - Request parameters: name=”Http Only”, destinationGroupID= “appTierGroup”, defaultAction=”deny” }
   - Response Parameters: SecurityPolicyID=”httpOnly”
2. Create a security policy rule associated with the *httpOnly* security policy:
   - Content type = application/vnd.juniper.nd.security-policy-rules+xml;version=1;charset=UTF-8
   - POST https://172.68.52.10/api/juniper/nd/tenants/{tenant-id}/naas-services/{naas-service-id}/l3-connectivity-services/{l3-connectivity-service-id}/

Copyright © 2013, Juniper Networks, Inc.

security-policies/{security-policy-id}/security-policy-rules

- Request parameters: name="Allow Http", condition=["tcpPort", "equals", "http"], action="allow"
- Response Parameters: SecurityPolicyRuleID="allowHttp"

Activating the NaaS Services

Step-by-Step Procedure

This section describes the steps for activating NaaS services.

1. Send a PUT request:
   - Content type = application/vnd.juniper.nd.naas-services+xml;version=1;charset=UTF-8
   - PUT https://172.68.52.10/api/juniper/nd/tenants/{tenant-id}/naas-services/{naas-service-id}/
   - Request parameters: naasServiceID="multitierAppNetwork"

Verifying the NaaS Request

Verifying That the NaaS Request Is Activated

Action

To verify that the NaaS services are activated:

1. Send a GET request to see the current state of the NaaS services:
   - Content type = application/vnd.juniper.nd.naas-services+xml;version=1;charset=UTF-8
   - URL = https://172.68.52.10/api/juniper/nd/tenants/{tenant-id}/naas-services/{naas-service-id}/
   - Request parameters: naasServiceID="multitierAppNetwork"
   - Response parameters: current state

   The current state of the NaaS services should be *activated*.

Related Documentation

- Overview of Network Director API on page 3
- Locating API Reference Information on page 29

Copyright © 2013, Juniper Networks, Inc.

Copyright © 2013, Juniper Networks, Inc.

CHAPTER 5

Accessing API Reference Documentation

Locating API Reference Information

Network Director Application Program Interface (NDAPI) is an application that is deployed on Junos Space. NDAPI exposes a set of Representational State Transfer (REST) APIs that allow you to create and manage network connectivity, also known as Network as a Service (NaaS). The *Network Director API Reference* contains details of the methods and collections that comprise these services.

The *Network Director API Reference* is a set of HTML files that is bundled with the API source. It is also available as a separate download from the Applications page of Junos Space: http://www.juniper.net/support/downloads/?p=spacenetdir#sw . The HTML files are packaged as a ZIP file. Unzip the file and open the index file, **application.html** to access the reference information.

Related Documentation

- Overview of Network Director API on page 3
- Setting Up the Network Director API on page 15
- Sample API Scripts Overview on page 11
- Network Director API Setup Sample Files on page 17

Copyright © 2013, Juniper Networks, Inc.

Copyright © 2013, Juniper Networks, Inc.