Cisco Network Services Manager 5.0 User Guide

Revised: November 13, 2012

Americas Headquarters
Cisco Systems, Inc.
170 West Tasman Drive
San Jose, CA 95134-1706
USA
http://www.cisco.com
Tel: 408 526-4000
800 553-NETS (6387)
Fax: 408 527-0883

Text Part Number: OL-26001-01

Cisco Network Services Manager 5.0 User Guide

CONTENTS

Preface

This document explains how to use Cisco Network Services Manager (Network Services Manager) 5.0 to create, revise, update, and manage physical and virtual networks.

New and Changed Information

The following table describes information that has been added or changed since this guide was first published.

Table 1 New and Changed Information in This Guide

Date Released	Revision	Location
November 13, 2012	Updated the text for the Network Services Manager controller to include the following requirements: • A configured device stack • Functioning connections to the engine and all devices in the stack	• Network Services Manager Controller, page 1-4 • NB API Resources, page 4-1 • Creating Network Containers for Zones, VLANs, and Remote Access, page 4-9
January 9, 2012	Initial release.	—

Audience

This guide is for network operation and management staff who are responsible for creating, updating, and managing tenants, network containers, and other resources by using Network Services Manager technology.

Organization

This guide includes the following sections:

Section	Title	Description
1	Introducing Cisco Network Services Manager	Describes Network Services Manager features at a high level.
2	Understanding Network Services Manager Concepts	Describes and explains important Network Services Manager concepts.
3	Using the Network Services Manager NB API	Describes how to use the Network Services Manager northbound (NB) application programming interface (API).
4	Working with Tenants Using the NB API	Describes how to manage tenants, network containers, and other resources using the northbound API.
5	Using the Network Services Manager Administration UI	Describes how to use the Administration UI.
6	Monitoring Status	Describes techniques you can use in monitoring Network Services Manager.
7	Troubleshooting	Provides hints and suggestions for troubleshooting problems.
A	Northbound API Parameters	Identifies parameters that can be issued via the NB API.
Glossary		Defines terms used with Network Services Manager.

Conventions

This document uses the following conventions:

Convention	Indication
bold font	Commands, keywords, and user-entered text appear in **bold** font.
italic font	Document titles, new or emphasized terms, and arguments for which you supply values are in *italic* font.
[ ]	Elements in square brackets are optional.
{ x | y | z }	Required alternative keywords are grouped in braces and separated by vertical bars.
[x | y | z]	Optional alternative keywords are grouped in brackets and separated by vertical bars.
string	A nonquoted set of characters. Do not use quotation marks around the string or the string will include the quotation marks.
`courier` font	Terminal sessions and information the system displays appear in `courier` font.

< >	Nonprinting characters such as passwords are in angle brackets.
[ ]	Default responses to system prompts are in square brackets.
!, #	An exclamation point (!) or a pound sign (#) at the beginning of a line of code indicates a comment line.

Note Means *reader take note*.

Tip Means *the following information will help you solve a problem*.

Caution Means *reader be careful*. In this situation, you might perform an action that could result in equipment damage or loss of data.

Timesaver Means *the described action saves time*. You can save time by performing the action described in the paragraph.

Related Documentation

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

The following documents are available for Cisco Network Services Manager 5.0:

- *Cisco Network Services Manager 5.0 Quick Start Guide*
- *Cisco Network Services Manager 5.0 Release Notes*
- *Cisco Network Services Manager 5.0 User Guide*
- *Open Source Used in Cisco Network Services Manager 5.0*

Obtaining Documentation and Submitting a Service Request

For information on obtaining documentation, submitting a service request, and gathering additional information, see the monthly *What's New in Cisco Product Documentation*, which also lists all new and revised Cisco technical documentation, at:

http://www.cisco.com/en/US/docs/general/whatsnew/whatsnew.html

Subscribe to the *What's New in Cisco Product Documentation* as an RSS feed and set content to be delivered directly to your desktop using a reader application. The RSS feeds are a free service. Cisco currently supports RSS Version 2.0.

CHAPTER 1

Introducing Cisco Network Services Manager

This chapter provides an overview of Cisco Network Services Manager (Network Services Manager) and includes the following topics:

Network Services Manager Overview

Network Services Manager is network management software that helps build the network services you need to securely create and deploy a cloud computing infrastructure. By using Network Services Manager, you can organize your network resources into a flexible cloud infrastructure that integrates the network with your existing IT tools and processes.

Network Services Manager includes the following features:

Table 1-1 Network Services Manager Features

Feature	Description
Abstraction and virtualization	Network Services Manager virtualizes network services by abstracting a logical representation of the physical network that it manages. The abstraction and virtualization allow you to provision and deploy numerous individual network components quickly, thereby reducing network operations costs and accelerating service delivery.
Policy implementation and adherence	Policies enforce topology models, network features, and network behaviors, thereby ensuring adherence to organizational or architectural requirements, such as security or access to resources.

Table 1-1 Network Services Manager Features (continued)

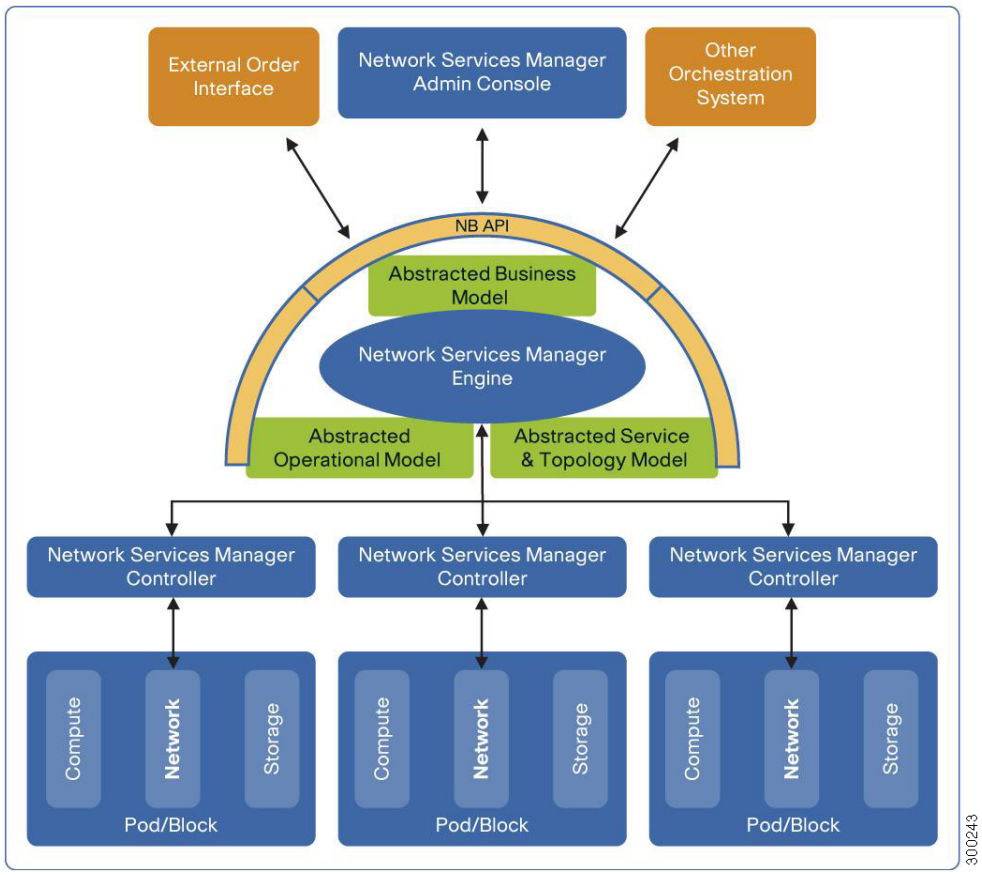
Feature	Description
Dynamic topology support	Network Services Manager retrieves abstract information from a business model, converts that information into device-specific configurations, and implements the resulting configurations on the appropriate devices. If the physical topology changes and the topology model is updated, Network Services Manager evaluates the defined policies against the new topology model to determine the best option for providing the required services while adhering to the rules and policies that you defined. Network Services Manager automatically disseminates the updated device-specific configurations to the affected devices to ensure ongoing adherence to policy.
Use of network containers	A network container is a logical group of virtual network resources that are created and managed as a unit. With network containers, Network Services Manager enables you to quickly and easily configure physical and virtual network infrastructure and network services to interoperate with computing and storage resources.
Administration user interface (UI)	Network Services Manager provides a browser-based Administration UI that enables you to view multiple virtual platforms (domains) and domain-specific policies.
Northbound (NB) Application Programming Interface (API)	The Network Services Manager NB API provides an integration interface for automated network provisioning as part of a larger cloud environment. The NB API enables you to instantiate cloud service models and topologies, create new cloud service offerings, modify existing offerings, and support containers and workloads. Additions, changes, or deletions made using the NB API are reflected in the Administration UI.

Network Services Manager Architecture

Network Services Manager includes the following components, as shown in Figure 1-1:

- Administration UI, page 1-3
- Network Services Manager Engine, page 1-3
- Network Services Manager Controller, page 1-4
- Integration with External Systems, page 1-4

Figure 1-1 Network Services Manager Architecture



Administration UI

The browser-based Administration UI (Admin Console in Figure 1-1) enables you to view network characteristics, services, and behaviors.

Options are available for:

- Viewing the running configuration on a device.
- Viewing metamodels, metaproperties, resources, zones, and sites.
- Viewing system alerts.

For more information, see Using the Network Services Manager Administration UI, page 5-1.

Network Services Manager Engine

The Network Services Manager engine automates the provisioning of end-to-end network services and dynamically generates the configuration instructions that control the devices and services in a multiple-tenant environment. The engine dynamically builds and deploys these configuration instructions by sending them to the Network Services Manager controllers for implementation.

Network Services Manager Controller

The Network Services Manager controller is the agent for the device stack. As the device stack agent, the controller requires:

- A configured device stack
- Functioning connections to the engine and all devices in the stack

When a user creates a container that is subordinate to a tenant (such as a zone, VLAN, or service policy), the engine verifies via the controller that the devices in the stack can support the policies required by the new container.

If the policies can be supported, the controller translates the request for the new container into specific device configurations for each device in the stack, depending on the device role in the topology.

Integration with External Systems

Network Services Manager can be integrated with external systems such as order entry portals or service catalogs, where user requests for network services can be captured and pushed to the Network Services Manager engine for provisioning. Cisco Intelligent Automation for Cloud can also use this interface to move critical information to and from Network Services Manager. This is also true for other ecosystem technologies, such as higher-level orchestration frameworks or specific configuration management systems.

External systems communicate with the Network Services Manager engine through the NB API. The NB API adheres to the Representational State Transfer (REST) conventions and architectural style, which provides a uniform interface for:

- Resource identification
- Resource manipulation through representations
- Self-descriptive messages
- Navigation by hyperlinks

For more information, see Using the Network Services Manager NB API, page 3-1.

CHAPTER 2

Understanding Network Services Manager Concepts

This chapter describes and explains the following Network Services Manager concepts:

Network Services Virtualization

Network Services Manager defines network services virtualization as a model-based definition of:

- A network addressing space.
- The physical and virtual resources in the network addressing space.
- The managed services, capabilities, and relationships between the identified network resources.

Network Services Manager virtualizes network services by abstracting a logical representation of the physical network that it manages. This logical network is driven by policies that control a user's network access to resources. These policies specify high-level resource sharing and can be created by using the NB API.

Metamodels

A metamodel is an abstraction of a model, similar to the way in which a model is an abstraction of one or more concepts (such as terms, items, or attributes). The difference is that metamodels emphasize the properties of the model itself. In Network Services Manager, a metamodel consists of definitions for specific entities and resources that can exist in a network, and the relationships that exist among these entities and resources. Network Services Manager uses metamodels to control network access and specify resource sharing.

Metamodels are defined in XML-based documents that can be imported, exported, or edited using the Network Services Manager NB API. Metamodels use standard variable and parameter substitutions that enable you to specify the attributes required for your environment.

A primary benefit of metamodels is that, when used with the Network Services Manager northbound API, they enable you to quickly provision and deploy numerous individual network components and resources. As a result, they accelerate and simplify cloud deployments, thereby reducing network operations costs and accelerating service delivery.

Metamodels can be set up during the Network Services Manager installation, or written or adapted by Network Services Manager administrators. Metamodels can be tailored to each environment or situation, enabling you to represent your network in the manner that best suits your environment.

Metamodels offer flexibility by allowing you to determine the parameters that can or must be specified when an entity is created via the Network Services Manager northbound API or the Administration UI.

Metamodel definitions can limit the number of resources you can create with a specific metamodel. These metamodel limits ensure that:

- Extraneous network containers are not created. For example, because a tenant only needs one network container for an external network connection (ENC), the limit is one.
- Resources are available to all tenants and cannot be consumed by one tenant alone.

For more information about working with metamodels, see the Using the Network Services Manager NB API, page 3-1.

Policies

Network Services Manager uses policies to manage resources and network access:

- Network access policies define entitlements and access management on a network, and are the primary vehicle for providing VLAN access to the network. Network access can range from a direct connection to a remote location to a private, routed Layer 3 VLAN.
- Business policies give one set of resources access to another set of resources. They are defined in terms of resources, schedule, connection topology, and services. Services can be Network Address Translation (NAT) or firewall services which, in turn, have attributes that define how these resources access each other.

  You can add or remove resources (including local resources, groups, and VLANs) from business policies, controlling which systems or subnets can communicate with each other. You can also activate or deactivate business policies.

Cloud Metamodels and Instances

A cloud metamodel describes a cloud domain, its properties, and any associated resources, policies, and subordinate metamodels that are necessary to express the particular network requirements of that cloud to Network Services Manager.

When you instantiate a cloud from a metamodel in Network Services Manager, the cloud and all related objects described in the metamodel are created in the business model. Taken as a whole, these business model objects define the resources and capabilities of that particular cloud as specified by the metamodel. The cloud can be further defined with subordinate (child) clouds and resource metamodels that are derived from the parent cloud metamodel.

A cloud instance is the entire collection of business model objects needed to define a cloud from the parent cloud through all subordinate clouds, including any contained resources and relevant policies. In essence, the cloud instance contains all the information that Network Services Manager needs to fully provision a cloud at a given site. One site can support multiple cloud instances.

Cloud instances are independent of any site and the particular device configuration that might be found at a site. A cloud instance can be assigned to any site that has the capabilities necessary to implement the cloud (that is, the devices and roles required by the cloud instance).

Resources

A resource is any item that can be associated in a business policy. Examples of resources are groups, subnets, address pools, zones, and VLANs.

Sites

A site corresponds to a single logical location. In many cases, a Network Services Manager site represents a discrete physical location.

Each site is managed by a controller; one or more Network Services Manager-managed devices; and one or more resources, including subnets or VLANs. Address space for a site is managed by specifying subnet definitions or assigning a VLAN to the site.

Physical Topology

Devices managed at a site are typically interconnected in a physical hierarchy. For example, it is common for a device hierarchy to consist of access switches at the bottom, feeding into one or more levels of higher-bandwidth aggregation switches, and terminating with a very-high-bandwidth distribution layer at the top. Additional devices, such as Layer 3 routers or edge firewalls, might reside above the distribution layer. Additional Layer 2 or Layer 3 service-level devices might exist within the switch hierarchy, providing firewall, load balancing, or QoS capabilities. The term *physical topology* refers to the description of all devices managed at a site and the manner in which they are interconnected.

Device Stack (Pod)

Note Network Services Manager uses the word *pod* to refer to the stack of devices that the controller manages. For clarity, this document uses the term *device stack*.

By default, Network Services Manager provides the metamodels and policies that support the device stack shown in Figure 2-1. If your physical topology differs, the metamodels and policies can be modified to support your environment. For more information, contact Cisco Advanced Services or your Cisco representative.

Figure 2-1 Network Services Manager Device Stack

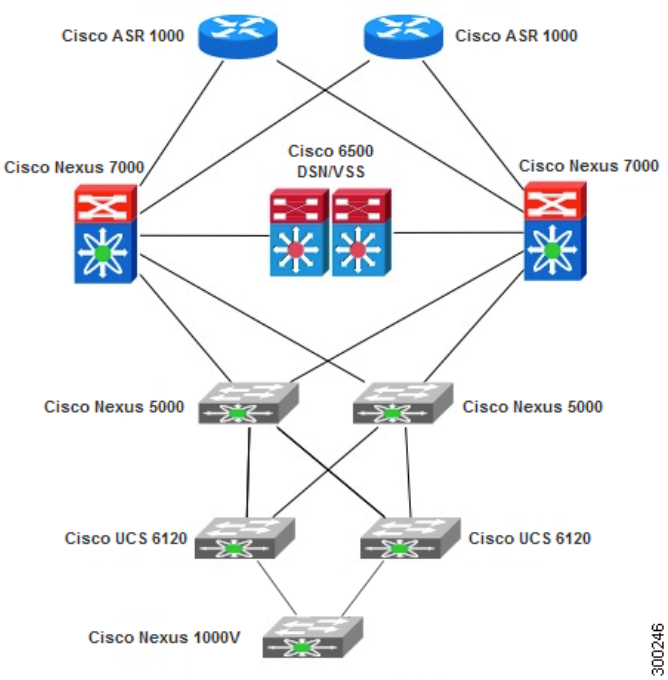


Table 2-1 describes the network elements in the device stack, their roles, and the support that they provide.

Table 2-1 Device Stack Roles and Support

Device	Role	Support
Two Cisco ASR 1000 routers	Edge router	• Virtual Routing and Forwarding (VRF) • VLANs • Dynamic routing via OSPF
Two Cisco 6500 switches in a Virtual Switching System (VSS) configuration acting as a Cisco Data Service Node (DSN)	Layer 3 services	• Enables Network Services Manager to inventory the modules installed on the devices and use those modules to create contexts. • The Cisco DSN/VSS implementation supports VLANs and SVIs. • The Cisco Application Control Engine (ACE) and Firewall Services Modules (FWSMs) in the Cisco 6500 chassis support ACLs. • When in routed mode, the Cisco ACE contexts support static routing.
Two Cisco Nexus 7000 switches	Distribution switch	• VRF • VLANs • SVIs • Dynamic routing via OSPF
Two Cisco Nexus 5000 switches	Aggregation switch	VLANs
Two Cisco UCS 6120 Fabric Interconnects	Aggregation switch	• VLANs • The assignment of VLANs to managed trunk ports and server profile vNICs
One Cisco Nexus 1000V switch	Access switch	VLANs

Dynamic Topology

In Network Services Manager, dynamic topology refers to the process of retrieving abstract information from a business model, converting that abstract information into device-specific configurations, and implementing those configurations on the appropriate devices. The result is a working version of the network based on:

- Knowledge of the physical and logical topologies of the network.
- Knowledge of specific device roles.
- Data paths constructed through the device stack by Network Services Manager that support the required service chains.

When constructing data paths, Network Services Manager evaluates the requested service chain (or policy) against the topology model to determine the best option through the device stack for providing the desired data path. This flexibility in creating data paths removes constraints on the types of service chains allowed, resulting in support for a broader set of service chains.

If required, service chains and network containers can be customized by Cisco Advanced Services to meet specific business needs. For more information, contact Cisco Advanced Services or your Cisco representative.

Domains

A domain is an organizational structure in Network Services Manager that delegates administrative control and provides organizational separation for tenants. A domain can contain all of the business model objects used to manage clouds and enables administrative and management separation for cloud tenants.

Domains contain sites, policies, and so on, providing you with an abstract view of the network in a hierarchy in the Network Services Manager Administration UI. The ROOT domain resides at the top of the hierarchy and contains all other network elements, including tenants and network containers.

Because each object in the system exists within a domain, the entire set of configuration items also has a hierarchical structure.

Tenants and Network Containers

In Network Services Manager, a *tenant* is a domain that contains business model objects that set up and manipulate one or more logical network containers. For example, a tenant might represent a service provider customer who wants to build a virtual cloud and populate it with virtual machines, or an enterprise customer who wants to allocate different services and resources to various departments.

A *tenant network container* identifies the device stack that is deployed for a tenant. It is explicitly created by a tenant using Network Services Manager and is defined by a metamodel. One tenant can have multiple tenant network containers that can be combined as needed to meet the needs of the tenant.

When you create a tenant network container, you associate it with a specific *device stack*, which is a physical stack of devices that is managed by Network Services Manager at a *site*. A site can be a geographic location, a remote unmanaged environment such as the Internet, or an MPLS remote location.

A tenant network container comprises building blocks called *network containers*. Network containers can contain *zones*, VLANs, and ENCs. A zone is a logical collection of network components that share the same security profile. A zone can contain VLANs or ENCs. An ENC allows remote access to a zone via MPLS or a direct connection. A single tenant network container can contain multiple network containers.

Tenant Placement

Network Services Manager uses the concept of blocks in its architecture to enable it to scale the environment as needed. Each block is limited with regard to the number of services (such as VLANs or VRF instances) that can be created within them. However, by locating tenants judiciously, Network Services Manager can optimize the available services.

For example, Network Services Manager can optimize the use of the available services by placing tenants in a way that does not require tenant VLANs to be populated everywhere. If a Cisco UCS device supports 512 VLANs and you populate tenant VLANs on each Cisco UCS device in the stack, you will have fewer than 512 VLANs across the entire stack. However, if you connect the Cisco UCS devices to the stack in a way that constrains a tenant to one block, you can have approximately 512 VLANs per block instead of across the entire stack.

No user interaction is required to take advantage of tenant placement. If no placement information is provided, Network Services Manager uses an algorithm to determine the best arrangement for optimizing the available services.

Redundancy

Network Services Manager automatically configures redundancy for all duplicate devices in the device stack as follows:

- For Layer 3 and above, redundancy is configured by using Hot Standby Redundancy Protocol (HSRP) on the duplicate devices.
- For Layer 2 and below, redundancy is supported by using identical configurations on the duplicate devices.

CHAPTER 3

Using the Network Services Manager NB API

This chapter describes how to use the Network Services Manager Northbound (NB) API. Topics include:

NB API Overview

The primary goal of the Network Services Manager NB API is to facilitate automated network provisioning in cloud environments. To aid in meeting this goal, the NB API follows REST conventions and architectural style.

The primary features of the REST architecture are that it:

- Is stateless—The server does not maintain session states for REST. Instead, the server processes each client request independently and does not rely on a session-specific context. As a result, each request issued to the server must contain all relevant information.
- Uses a client-server architecture—A uniform interface separates clients from servers, thereby freeing clients from data storage concerns and servers from a user interface or user state.
- Can cache responses—Responses are defined as cacheable or noncacheable so that clients do not use old or incorrect data for subsequent requests. By managing caching carefully, you can eliminate unneeded client-server interactions and thereby improve performance and scalability.
- Works with layered systems—Because REST interactions are stateless and can be cached, a client can interact with a server through intermediate layers, such as shared caches, load balancers, or security proxies, which can improve performance, scalability, and security.

A REST-compliant interface adheres to the following standards:

- Resource identification—In REST, a resource is a source of specific information that is identified by using a Uniform Resource Identifier (URI).
- Resource manipulation—To manipulate resources, clients and servers communicate via a standardized interface (such as HTTP) and exchange representations of these resources. A representation is in a particular encoding (such as XML), and contains some or all of the data in the resource. The data in the resource is manipulated through these representations, through the use of HTTP DELETE, PUT, or POST operations.

- Self-descriptive messages—Each message contains enough information to be parsed appropriately by the client or server.
- Hypermedia as the engine—A REST client enters a REST application through a simple fixed URL. All future actions the client takes are discovered within resource representations returned from the server. The media types used for these representations, and the link relations they contain, are standardized. The client transitions through application states by selecting from the links within a representation or by manipulating the representation in other ways afforded by its media type. In this way, the interaction is driven by hypermedia.

NB API Request Methods

The Network Services Manager NB API uses the HTTP request methods described in Table 3-1.

Table 3-1 HTTP Request Methods

HTTP Request Method	Description
GET	Retrieves the specified resource or representation. GET is a read-only operation that does not change the engine state or have any side effects.
POST	Submits data to be processed to the specified resource. The data to be processed is included in the request body. A POST operation can create a new resource, update an existing resource, or both. POST operations are defined as *nonidempotent*; that is, submitting the same POST operation more than once returns a different result for each operation. For example, if you issue a POST operation with the **create** URI to create a new tenant, a tenant is created with its own URI. If you issue the same POST operation again, another tenant is created with a different URI.
PUT	Updates the specified resource with new information. The data that is included in the PUT operation replaces the previous data. PUT operations are defined as *idempotent*; that is, submitting the same PUT operation more than once returns the same results. For example, issuing multiple PUT operations to change a tenant description from New York to Chicago returns the same result each time.
DELETE	Deletes a resource. If you delete a resource that has already been deleted, a 404 Not Found response is returned.

If you are familiar with Create, Read, Update, and Delete (CRUD) operations, the following mapping of CRUD to REST operations might help:

- Create—POST
- Read—GET
- Update—PUT
- Delete—DELETE

These mappings are not precise and can change over time.

NB API Responses

The NB API responds to all requests it receives with either synchronous or asynchronous responses:

- Synchronous responses—Synchronous responses are returned when little processing is required by the server. When using a REST client, synchronous responses are provided in response to GET requests because the response requires only the retrieval of information, such as a list of the network containers.
- Asynchronous responses—POST, PUT, and DELETE requests usually require more processing than a GET request, and the NB API responds by issuing a synchronous response that contains a Task object UID and a Task URL. The NB API continues to work on the initial request in the background until it is complete. The Task URL allows the NB system to determine the status of the original request as it progresses.

  The Task status object in the task response can be used by the NB system to determine the success or failure of updates to the business model. A successful update to the business model triggers the provisioning engine to run, which could also update the service directives for the controller. Provisioning and controller failures are displayed as alerts in the logs and can be viewed in the Administration UI Alert View.

The following topics provide more information about status codes and NB API responses:

- Status Codes and Error Handling, page 3-3
- Response Components, page 3-4

Status Codes and Error Handling

The NB API uses standard HTTP status codes to report the success or failure of the submitted requests:

- HTTP status codes from 200-299 indicate success.
- HTTP status codes 400 and higher indicate failure.

Table 3-2 describes common HTTP status codes and descriptions.

Table 3-2 HTTP Status Codes and Descriptions

Code	Status Reason	Description
200	OK	The request has succeeded.
201	Created	An asynchronous task has completed, and the object has been created.
202	Accepted	An asynchronous task has been accepted, but the processing is not complete.
400	Bad Request	An invalid request has been submitted. Verify that the request uses the correct syntax.
404	Not Found	The specified resource cannot be found.
409	Conflict	An attempt has been made to modify a resource that has already been modified by another request.
500	Internal Server Error	The server encountered an unexpected condition that prevents it from completing the request.

Response Components

The response might or might not contain a response body. If a response body is included:

- A successful response includes a representation of the resource.
- A failure response includes an error object.

Table 3-3 describes the components of a task response in the order in which they are given.

Table 3-3 Components of a Task Response

Information	Description
Task UID	Unique identifier for the task.
Task URL	Link to the task in the form of an HREF.
Task status	Current task status: • Pending—The task has not started processing. • In progress—The system is currently processing the task. • Success—The task completed successfully. • Failure—A problem occurred while the task was being processed.
Time stamps	Task start, end, modification, and expiration times.
Percentage task complete	Percentage of task complete.
Failure reason	Provided if the task fails.

Figure 3-1 shows an example of a task response with a failure status.

Figure 3-1 Example Failure Response


```
- <task>
    <uid>21c94da3f8504d53910169cb4515a2ac</uid>
    <name>Create Tenant Favorite Company</name>
  - <links>
      <link title="Create Tenant Favorite Company" rel="self" href="https://od-testnsve01.cisco.com:8443/NetworkHypervisorAPI/v1/task/21c94da3f8504d53910169cb4515a2ac" />
    </links>
  - <status>
      <taskStatus>failure</taskStatus>
      <responseCode>409</responseCode>
    </status>
    <startTime>2011-11-11T11:43:52.094-05:00</startTime>
    <endTime>2011-11-11T11:43:52.426-05:00</endTime>
    <expiryTime>2011-11-11T12:43:52.094-05:00</expiryTime>
    <percentComplete>100</percentComplete>
  - <fault>
      <faultType>error.domain.exists</faultType>
      <message>Domain named "Favorite Company" already exists.</message>
      <details>Favorite Company</details>
    </fault>
  </task>
```


300282

NB API Client Conventions

Adhere to the following conventions when using the Network Services Manager NB API:

- All service offering requests must include the Network Services Manager namespace information:

```
<serviceOffering xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns="http://www.cisco.com/NetworkServicesManager">
```

- All request headers must include:
  - Accept: application/xml
  - Content-Type: application/xml
- Descriptions must contain no more than 250 characters.
- The NB API user must use HTTPS to invoke the northbound interface.
- All the APIs must be executed by a user with admin access to the system.

CHAPTER 4

Working with Tenants Using the NB API

This chapter describes how to manage tenants and network containers using the Network Services Manager NB API:

NB API Resources

Table 4-1 identifies the REST resources that are available via the NB API, along with a brief description, restrictions, and the supported REST operations.

Table 4-1 Network Services Manager REST Resources

NB API Resource	Description	Restrictions	Supported NB API Operations
ENC	A container that supports remote access via either MPLS or direct connection. Creating a zone network container allows the selection of an ENC.	• Deployed to the device stack of the parent tenant network container. • Name must be unique within the parent container.	• GET • POST • PUT • DELETE

Table 4-1 Network Services Manager REST Resources (continued)

NB API Resource	Description	Restrictions	Supported NB API Operations
Network Container	A container that holds virtualized services and VMs.	• Must reside in a tenant network container or another network container. • Deployed to the device stack of the parent tenant network container. • Name must be unique within the parent container.	• GET • POST • PUT • DELETE • GET the service catalog
Pod	A stack of devices that manages cloud operations.	• Cannot be created by using the NB API. • Device stack must be configured. • Controller must have functioning connections to the engine and all devices in the stack.	—
Service Catalog/ Service Offering	A list of available service offerings.	—	GET
Task	Reports the status of a long-running operation	—	GET
Tenant	A logical entity that can represent an organization or location.	• Name must be unique within an instance of Network Services Manager. • Can be associated with only one domain.	• GET • POST • PUT • DELETE • GET a list of the device stacks for this tenant • GET the service catalog
Tenant Network Container	An entity defined by a metamodel that contains one or more network containers.	• Deployed to a specific device stack. • Name must be unique within the parent container.	• GET • POST • PUT • DELETE • GET the service catalog

Each resource has a name, description, and various configuration settings (parameters). For information on the parameters that can be used when working with metamodels, see Northbound API Parameters, page A-1.

Getting Started with a REST Client

Note The procedures in this guide use Firefox Mozilla with the RESTClient tool.

To get started with a REST client:

Step 1 In your browser, choose **Tools > REST Client**.

Step 2 From the Method drop-down list, choose **GET**.

Step 3 In the location field, enter:

`https://hostname:8443/NetworkServicesManagerAPI/v1/`

where ***`hostname`*** is the name of the Network Services Manager engine.

Step 4 Click **Add Request Header** and add the following headers:

Name	Value
accept	application/xml
content-type	application/xml

Step 5 Click **Send**.

Step 6 When you receive a response, click the **Formatted XML** tab.

The available links for working with the NB API via a REST client are listed, as shown in Figure 4-1.

After you log in and set the request headers, you are ready to continue with the other procedures in this chapter.

XML Example

Figure 4-1 shows an example of top-level links using a REST client.

Figure 4-1 Top-Level Links


```
- <service>
    <name>NetworkServicesManagerAPI</name>
    <description>Cisco Network ServicesManager RESTful Web Service</description>
    <version>1</version>
  - <links>
      <link title="NetworkServicesManagerAPI" rel="self" href= "https://od5-demo01:8443/NetworkServicesManagerAPI/v1" />
      <link rel="tenants" href= "https://od5-demo01:8443/NetworkServicesManagerAPI/v1/top/tenant" />
      <link rel="catalog" href= "https://od5-demo01:8443/NetworkServicesManagerAPI/v1/top/catalog" />
    </links>
  </service>
```


300244

Creating Tenants

As described in Tenants and Network Containers, page 2-6, a tenant is a domain that contains business model objects that set up and manipulate one or more logical network containers.

You can create tenants and network containers using a REST client with a browser and then import these new items into Network Services Manager using the NB API. Any tenants and network containers that you create using the NB API are automatically displayed in the Administration UI.

This procedure requires you to specify one public address pool and one private address pool.

To create a tenant using a REST client:

Step 1 In your browser, choose **Tools > REST Client**.

Step 2 In the REST client:

a. From the Method drop-down list, choose **GET**.

b. Enter the following location:

```
https://hostname:8443/NetworkServicesManagerAPI/v1/top/catalog
```

where ***hostname*** is the name of the Network Services Manager engine.

c. Click **Send**.

Step 3 In the response pane, click **Formatted XML** to view the available tenant service offerings and their UIDs. (See Figure 4-2.)

Step 4 Note the UID of the required tenant service offering (two are highlighted in Figure 4-2). You need this information to create the tenant.

Tip You can copy and paste the UID from the Response Body tab when needed.

Step 5 In the Formatted XML tab, locate the type of tenant you want to create and its **create** hyperlink.

Step 6 Click the **create** hyperlink (see Figure 4-3) to enter it in the REST request field.

Step 7 Make sure that the content-type and application request headers exist, each with the value application/xml.

Step 8 In the Method field, choose **POST**.

Step 9 In the Request Body field, enter the following information:

```
<serviceOffering xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns="http://www.cisco.com/NetworkServicesManager">
     <uid>service-uid</uid>
     <name>service-name</name>
     <description>service-description</description>
   <parameters>
      <parameter>
           <name>public.address.subnet</name>
           <type>subnet</type>
                 <subnet>
                       <ipv4>ipv4-address</ipv4>
                       <mask>pool-mask</mask>
                 </subnet>
       </parameter>
```

```
        <parameter>
            <name>private.address.subnet</name>
            <type>subnet</type>
                  <subnet>
                        <ipv4>ipv4-address</ipv4>
                        <mask>pool-mask</mask>
                  </subnet>
        </parameter>
</parameters>
</serviceOffering>
```

where:

- ***service-uid*** is the UID of the tenant service offering. (See Step 4.)
- ***service-name*** is the name you want to give the new tenant.

  We recommend that you name the tenant something that is meaningful and quickly decipherable, such as Boston or Boston State Street Branch.
- ***service-description*** is a comment about the new tenant. Descriptions are limited to 250 characters.
- ***ipv4-address*** is an IPv4 address in dotted decimal format, such as 192.0.2.1.
- ***pool-mask*** is the mask applied to the IPv4 address to indicate the size of the address pool, such as 24.

Step 10 Click **Send**.

The response pane displays the request with a hyperlink to the create task itself, and the task status: pending, failure, in progress, or success. (See Figure 4-4.)

Step 11 If the request is in pending state, view an updated status by clicking the **self** hyperlink, changing the method to **GET**, and clicking **Send**.

If the creation task is successful, information similar to that in Figure 4-5 is displayed.

If the request reports a failure status, additional information is provided in the response pane so that you can address the failure.

For more information on task status, see Status Codes and Error Handling, page 3-3.

Step 12 After you receive a success status, you can confirm that the tenant is created by either of the following methods:

- In the REST client:

  a. Change the method to **GET**.

  b. Enter the following in the request field:

  ```
  https://hostname:8443/NetworkServicesManagerAPI/v1/top/tenant
  ```

  where ***hostname*** is the name of the Network Services Manager engine.

  c. Click **Send**. The response includes the newly created tenant.
- Log into the Network Services Manager Administration UI and view the tenants. The Domain Navigator includes the newly created tenant.

XML Examples

The following figures show examples of XML responses and selections when using a REST client to create a tenant.

Figure 4-2 Tenant Service Offerings and UIDs


```
- <serviceOfferings>
  - <serviceOffering type="Tenant">
      <uid>9446659dc39634c52bbc62788bedf46bc</uid>
      <name>Create Direct Access Tenant</name>
    - <description>
        A model for a tenant environment with a single protected zone and an unprotected zone. Each tenant has their own public address pool, private address pool and infrastructure pool. These are
        defined on tenant creation. The maximum number of tenants supported by this model is set with the limit attribute
      </description>
      <version>3</version>
    - <links>
        <link title="Create Direct Access Tenant" rel="self" href= "https://od5-demo01:8443/NetworkServicesManagerAPI/v1/serviceoffering/944659dc39634c52bbc62788bedf46bc" />
        <link rel="create" href= "https://od5-demo01:8443/NetworkServicesManagerAPI/v1/top/catalog/create" />
      </links>
    </serviceOffering>
  - <serviceOffering type="Tenant">
      <uid>e0553900eebe480b90d8e9e0b4fc644a</uid>
      <name>Create MPLS Access Tenant</name>
    - <description>
        A model for a tenant environment with a single protected zone and an unprotected zone. Each tenant has their own public address pool, private address pool and infrastructure pool. These are
        defined on tenant creation. The maximum number of tenants supported by this model is set with the limit attribute
      </description>
      <version>1</version>
    - <links>
        <link title="Create MPLS Access Tenant" rel="self" href= "https://od5-demo01:8443/NetworkServicesManagerAPI/v1/serviceoffering/e0553900eebe480b90d8e9e0b4fc644a" />
        <link rel="create" href= "https://od5-demo01:8443/NetworkServicesManagerAPI/v1/top/catalog/create" />
      </links>
    </serviceOffering>
  </serviceOfferings>
```


300241

Figure 4-3 Tenant Create Hyperlink


```
Response Header | Response Body | Formatted XML

- <serviceOfferings>
  - <serviceOffering type="Tenant">
      <uid>3488b483d90046a3a7b5017aefb657cc</uid>
      <name>Create Tenant model</name>
    - <description>
        Each tenant has their own public address pool, private address pool. This model is the most basic stripped down model that allows the tenant to only create L3 routable VLANs.
      </description>
      <version>1</version>
    - <links>
        <link title="Create Tenant model" rel="self" href= "https://od-testnsve01.cisco.com:8443/NetworkHypervisorAPI/v1/serviceoffering/3488b483d90046a3a7b5017aefb657cc" />
        <link rel="create" href= "https://od-testnsve01.cisco.com:8443/NetworkHypervisorAPI/v1/top/catalog/create" />
      </links>
    </serviceOffering>
  </serviceOfferings>
```


300281

Figure 4-4 Example Task Status


```
Response Header | Response Body | Formatted XML

- <task>
    <uid>b4e72d3f0eb942ca92c2e0b072ff566d</uid>
    <name>Create Tenant Company ABC Tenant</name>
  - <links>
      <link title="Create Tenant Company ABC Tenant" rel="self" href= "https://od-testnsve01.cisco.com:8443/NetworkHypervisorAPI/v1/task/b4e72d3f0eb942ca92c2e0b072ff566d" />
    </links>
  - <status>
      <taskStatus>pending</taskStatus>
    </status>
    <startTime>2011-11-08T11:32:53.055-05:00</startTime>
    <expiryTime>2011-11-08T12:32:53.055-05:00</expiryTime>
    <percentComplete>0</percentComplete>
  </task>
```


300242

Figure 4-5 Example Successful Task


```
Response Header | Response Body | Formatted XML
- <task>
    <uid>b4e72d3f0eb942ca92c2e0b072ff566d</uid>
    <name>Create Tenant Company ABC Tenant</name>
  - <links>
      <link title="Create Tenant Company ABC Tenant" rel="self" href= "https://od-testnsve01.cisco.com:8443/NetworkHypervisorAPI/v1/task/b4e72d3f0eb942ca92c2e0b072ff566d" />
    </links>
  - <status>
      <taskStatus>success</taskStatus>
      <responseCode>201</responseCode>
    </status>
    <startTime>2011-11-08T11:32:53.055-05:00</startTime>
    <endTime>2011-11-08T11:32:54.725-05:00</endTime>
    <expiryTime>2011-11-08T12:32:53.055-05:00</expiryTime>
    <percentComplete>100</percentComplete>
  - <result xsi:type="TenantValue">
      <uid>d25198aa06c1473d8bfab12fbac18e67</uid>
      <name>Company ABC Tenant</name>
      <description>Tenant for Company ABC</description>
      <version>2</version>
    - <links>
        <link title="Company ABC Tenant" rel="self" href= "https://od-testnsve01.cisco.com:8443/NetworkHypervisorAPI/v1/tenant/d25198aa06c1473d8bfab12fbac18e67" />
        <link rel="catalog" href= "https://od-testnsve01.cisco.com:8443/NetworkHypervisorAPI/v1/tenant/d25198aa06c1473d8bfab12fbac18e67/catalog" />
        <link rel="pods" href= "https://od-testnsve01.cisco.com:8443/NetworkHypervisorAPI/v1/tenant/d25198aa06c1473d8bfab12fbac18e67/pods" />
      </links>
      <properties/>
    </result>
  </task>
```


300249

Creating Tenant Network Containers

After you create a tenant, you need to create a tenant network container to identify the device stack and IP address pools for the tenant to use.

This procedure requires UIDs for the following (all of which are available by using a REST client as described in this procedure):

- Tenant
- Tenant network container
- Device stack to be used by the tenant network container

To create a tenant network container using a REST client:

Step 1 Obtain the tenant UID:

a. From the Method drop-down list, choose **GET**.

b. Enter the following location in the request field:

`https://hostname:8443/NetworkServicesManagerAPI/v1/top/tenant`

where ***`hostname`*** is the name of the Network Services Manager engine.

c. Click **Send**.

d. In the response, locate the required tenant, and note the UID.

Step 2 Obtain the device stack UID:

a. From the Method drop-down list, choose **GET**.

b. Enter the following location:

```
https://hostname:8443/NetworkServicesManagerAPI/v1/tenant/tenant-uid/pods
```

where:

- ***hostname*** is the name of the Network Services Manager engine.
- ***tenant-uid*** is the tenant UID from Step 1.

c. Click **Send**.

d. In the response, locate the required device stack (pod), and note the UID.

Step 3 Obtain the tenant network container UID:

a. From the Method field, choose **GET**.

b. Enter the following location:

```
https://hostname:8443/NetworkServicesManagerAPI/v1/tenant/tenant-uid/catalog
```

where:

- ***hostname*** is the name of the Network Services Manager engine.
- ***tenant-uid*** is the tenant UID.

c. Click **Send**.

d. In the response, locate the required tenant network container and note the UID.

Step 4 Click the **create** link for the required tenant network container and choose **POST** from the Method drop-down list.

Step 5 In the Request Body field, enter the following information:

```
<serviceOffering xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns="http://www.cisco.com/NetworkServicesManager">
     <uid>tnc-uid</uid>
     <name>tnc-name</name>
     <description>tnc-description</description>
   <parameters>
     <parameter>
           <name>tnc.pod</name>
           <type>uid</type>
           <uid>pod-uid</uid>
       </parameter>
     </parameters>
</serviceOffering>
```

where:

- ***tnc-uid*** is the UID of the tenant network container, available in Step 3.
- ***tnc-name*** is the name of the tenant network container.
- ***tnc-description*** is a description of the tenant network container. Descriptions can contain up to 250 characters.
- ***pod-uid*** is the UID of the device stack, available from Step 2.

Step 6 Click **Send**.

Step 7 If the response displays the status “pending,” click the **self** link, change the method to **GET**, and click **Send**.

A successful response indicates that the tenant network container was created and contains links to the following:

- Newly created tenant network container.
- Service offerings for the tenant network container.
- Parent of the tenant network container. In this example, the parent is the tenant.

XML Example

Figure 4-6 shows an example of a successfully created tenant network container.

Figure 4-6 Successfully Created Tenant Network Container


```
- <task>
    <uid>2a052622baef412e90a7aefa3bd79ff4</uid>
    <name>Create Tenant Network Container REST Tenant TNC</name>
  - <links>
      <link title="Create Tenant Network Container REST Tenant TNC" rel="self" href= "https://od5-demo01:8443/NetworkServicesManagerAPI/v1/task/2a052622baef412e90a7aefa3bd79ff4" />
    </links>
  - <status>
      <taskStatus>success</taskStatus>
      <responseCode>201</responseCode>
    </status>
    <startTime>2011-12-20T17:40:59.087-05:00</startTime>
    <endTime>2011-12-20T17:41:00.831-05:00</endTime>
    <expiryTime>2011-12-20T18:40:59.087-05:00</expiryTime>
    <percentComplete>100</percentComplete>
  - <result xsi:type="TenantNetworkContainerValue">
      <uid>6e65ec7eacaf4c9fa81aff276b2fd32d</uid>
      <name>REST Tenant TNC</name>
      <description>TNC for REST Tenant</description>
      <version>2</version>
    - <links>
        <link title="REST Tenant TNC" rel="self" href= "https://od5-demo01:8443/NetworkServicesManagerAPI/v1/tnc/6e65ec7eacaf4c9fa81aff276b2fd32d" />
        <link rel="catalog" href= "https://od5-demo01:8443/NetworkServicesManagerAPI/v1/tnc/6e65ec7eacaf4c9fa81aff276b2fd32d/catalog" />
        <link title="Favorite Company - Direct Access" rel="parent" href= "https://od5-demo01:8443/NetworkServicesManagerAPI/v1/tenant/900f46e1535d4ef9a12103dde2562455" />
      </links>
    - <properties>
      - <property isReadOnly="false">
          <name>tnc.pod</name>
          <type>uid</type>
          <uid>cee5920ea92042c4a97d0e5e49652cc0</uid>
        </property>
      </properties>
    </result>
  </task>
```


300250

Creating Network Containers for Zones, VLANs, and Remote Access

Network containers are created inside tenant network containers, and can contain zones, VLANs, and ENCs.

Note Before you create a container that is subordinate to a tenant, confirm that the controller has a configured device stack and functioning connections to the engine and all devices in the stack.

The following topics describe how to create zone network containers, and other network containers within them:

- Creating Zone Network Containers, page 4-10
- Creating VLAN Network Containers, page 4-11
- Creating ENC Network Containers, page 4-12

Creating Zone Network Containers

Zone network containers enable you to share or limit access to a tenant and its resources. For example, you might want to create a private and protected zone network container to limit access to sensitive information, and an unprotected zone network container to allow access to the internet. A single tenant network container can contain multiple zone network containers, enabling you to provide the levels of security that you need for your environment.

Zone security options include:

Security Option	Description
Private	The zone can be reached only via a private VPN over MPLS, SSL, or IPsec.
Private and Protected	The zone can be reached only via a private VPN over MPLS, SSL, or IPsec and is behind a firewall.
Private and Unprotected	The zone can be reached only via a private VPN over MPLS, SSL, or IPsec and is not behind a firewall.
Protected	The zone is behind a firewall.
Unprotected	The zone is not behind a firewall.

To create a zone network container in an existing tenant network container:

Step 1 From the Method drop-down list, choose **GET**.

Step 2 Enter the following location:

```
https://hostname:8443/NetworkServicesManagerAPI/v1/tnc/tnc-uid/catalog
```

where:

- ***hostname*** is the name of the Network Services Manager engine.
- ***tnc-uid*** is the UID of the tenant network container.

Step 3 Click **Send**. The response lists the available types of zone network containers.

Step 4 In the response, click the **create** link for the required zone network container.

Step 5 In the request body field, enter the following information:

```
<serviceOffering xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns="http://www.cisco.com/NetworkServicesManager">
     <uid>zone-nc-uid</uid>
     <name>zone-nc-name</name>
     <description>zone-nc-description</description>
   <parameters/>
</serviceOffering>
```

where:

- ***zone-nc-uid*** is the UID of the required network container.
- ***zone-nc-name*** is the name to assign to the network container.
- ***zone-nc-description*** is a description of the network container. Descriptions can contain up to 250 characters.

Step 6 From the Method drop-down list, choose **POST**.

Step 7 Click **Send**.

Step 8 If the response displays a pending status, click the **self** link, change the request method to **GET**, and click **Send**.

A successful response indicates that the tenant network container was created and contains links to the following:

- Newly created network container.
- Service offerings for the network container.
- Parent of the network container. In this case, the parent is the tenant network container.

Creating VLAN Network Containers

After you create a zone network container, you are ready to create one or more VLAN network containers so that users and machines can access resources as specified in the network access policies.

Network Services Manager offers the following types of VLAN network containers:

VLAN Type	Description
Layer 2 unrouted VLAN	Layer 2 unrouted VLAN without an associated IP address. This type of VLAN is typically deployed behind a service VM.
Layer 3 private, routed VLAN	Layer 3 routed VLAN drawn from a pool of private IP addresses.
Layer 3 private, routed VLAN with NAT	Layer 3 routed VLAN drawn from a pool of private IP addresses with NAT applied.
Layer 3 public, routed VLAN	Layer 3 routed VLAN drawn from a pool of public IP addresses.

A zone network container often contains only one VLAN.

To create a VLAN network container:

Step 1 From the Method drop-down list, choose **GET**.

Step 2 Enter the following location:

https://*hostname*.cisco.com:8443/NetworkServicesManagerAPI/v1/nc/*zone-nc-uid*/catalog

where:

- ***hostname*** is the name of the Network Services Manager engine.
- ***zone-nc-uid*** is the UID of the zone network container.

Step 3 Click **Send**.

The response lists the available service offerings for the specified zone network container.

Step 4 In the response, click the **create** link for the required VLAN network container.

Step 5 In the request body field, enter the required information for the VLAN:

- For a Layer 3 public or private routed VLAN, enter the mask.
- For a Layer 2 unrouted VLAN, you do not need to provide any information.

The following example is for a Layer 3 routed VLAN:

```
<serviceOffering xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns="http://www.cisco.com/NetworkServicesManager">
     <uid>vlan-uid</uid>
     <name>vlan-name</name>
     <description>vlan-description</description>
   <parameters>
      <parameter>
           <name>l3.vlan.size</name>
           <type>integer</type>
                 <integer>mask</integer>
       </parameter>
     </parameters>
</serviceOffering>
```

where:

- ***vlan-uid*** is the UID of the type of VLAN that you are creating.
- ***vlan-name*** is the VLAN name.
- ***vlan-description*** is a brief description of the VLAN. Descriptions can contain up to 250 characters.
- ***mask*** is the VLAN size, such as 24.

Step 6 From the Method drop-down list, choose **POST**.

Step 7 Click **Send**.

Step 8 If the response displays a pending status, click the **self** link, change the request method to **GET**, and click **Send**.

A successful response indicates that the VLAN network container was created and contains links to the following:

- Newly created VLAN network container.
- Service offerings for the VLAN network container.
- Parent of the network container. In this case, the parent is the zone network container.

Creating ENC Network Containers

If you need to allow remote access to a zone, you can do so by creating an ENC network container in the zone network container. An ENC network container allows remote access to the zone via either MPLS or a direct connection. ENCs are available for private or public unprotected zones; that is, zones that do not have a firewall.

Because a tenant needs only one ENC network container to allow remote access, you can create only one.

To create an ENC network container:

Step 1 From the Method drop-down list, choose **GET**.

Step 2 Enter the following location:

```
https://hostname.cisco.com:8443/NetworkServicesManagerAPI/v1/nc/zone-nc-uid/catalog
```

where:

- ***hostname*** is the name of the Network Services Manager engine.
- ***zone-nc-uid*** is the UID of the zone network container.

Step 3 Click **Send**.

The response lists the available service offerings for the specified zone network container.

Step 4 In the response, click the **create** link for an ENC network container.

Step 5 In the request body field, enter the following information:

```
<serviceOffering xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns="http://www.cisco.com/NetworkServicesManager">
     <uid>enc-uid</uid>
     <name>enc-name</name>
     <description>enc-description</description>
   <parameters/>
</serviceOffering>
```

where:

- ***enc-uid*** is the UID of the required ENC.
- ***enc-name*** is the name of the ENC.
- ***enc-description*** is a description of the ENC. Descriptions can contain up to 250 characters.

Step 6 From the Method drop-down list, choose **POST**.

Step 7 Click **Send**.

Step 8 If the response displays a pending status, click the **self** link, change the request method to **GET**, and click **Send**.

A successful response indicates that the ENC was created and contains links to the following:

- Newly created ENC.
- Service offerings for the ENC.
- Parent of the network container. In this case, the parent is the zone network container.

Updating Tenants and Network Containers

You can update or modify an existing tenant or network container via the NB API by using the REST **PUT** method. The procedure is the same, regardless of the object you update.

The Request Body content uses the following structure:

```
<service-offering xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns="http://www.cisco.com/NetworkServicesManager">
     <uid>uid</uid>
     <name>name</name>
```

```
    <description>description</description>
    <version>version</version>
  <parameters>
     <parameter>
          <name>parameter-name</name>
          <type>datatype</type>
                <subnet>
                   <ipv4>value</ipv4>
                   <mask>value</mask>
                </subnet>
      </parameter>
      <parameter>
          <name>parameter-name</name>
          <type>datatype</type>
                <subnet>
                   <ipv4>value</ipv4>
                   <mask>value</mask>
                </subnet>
      </parameter>
</parameters>
</service-offering>
```

where:

- *service-offering* is the type of service you are updating. Service names include:
  - tenant
  - tenantNetworkContainer
  - networkContainer
  - externalNetworkConnection
- *uid* is the UID of the object you are updating.
- *name* is the name assigned to this instance.
- *description* is the description for this instance. Descriptions can contain up to 250 characters.
- *version* is the current version of this object. You can obtain the current version number by clicking the **self** link for the object you are updating.
- *parameter-name* is the name of the parameter, such as public.address.subnet.
- *datatype* is the datatype for this parameter, such as subnet or integer.
- *value* is the value assigned to this parameter.

For information on the specific parameters required, see:

- Creating Tenants, page 4-4
- Creating Tenant Network Containers, page 4-7
- Creating Network Containers for Zones, VLANs, and Remote Access, page 4-9

To update an item via the NB API using a REST client:

Step 1 In the REST client, navigate to the required object. For example:

a. Set the request method to **GET**.

b. In the location field, enter the required location, such as:

https://*hostname*:8443/NetworkServicesManagerAPI/v1/top/tenant

where ***hostname*** is the name of the Network Services Manager engine.

c. Click the appropriate **child** link and click **Send**.

d. Continue navigating through the child links until the object you want is listed.

Step 2 Click the **self** link of the required object (see Figure 4-7).

Step 3 In the Request Body field, enter the updated information for the selected object.

Figure 4-8 shows an example for updating a tenant.

Step 4 Change the request method to **PUT**.

Step 5 Click **Send**.

Step 6 If the response displays a pending status, click the **self** link, choose the **GET** method, and click **Send**.

A successful response indicates that the item was updated as requested, and the version number is incremented.

XML Examples

The following figures show examples of updating a tenant using a REST client.

Figure 4-7 Object Self Link


```
- <tenantNetworkContainer>
    <uid>bb9e0c26bfcd46dca866f8ba24344059</uid>
    <name>TNC2</name>
    <description>Watching Loading message</description>
    <version>8</version>
  - <links>
      <link title="TNC2" rel="self" href= "https://od5-demo01:8443/NetworkServicesManagerAPI/v1/tnc/bb9e0c26bfcd46dca866f8ba24344059" />
      <link rel="catalog" href= "https://od5-demo01:8443/NetworkServicesManagerAPI/v1/tnc/bb9e0c26bfcd46dca866f8ba24344059/catalog" />
      <link title="New Tenant to watch" rel="parent" href= "https://od5-demo01:8443/NetworkServicesManagerAPI/v1/tenant/fc71f1bf676849a7abf6b3cb475b5821" />
      <link title="Protected Zone - REST" rel="child" href= "https://od5-demo01:8443/NetworkServicesManagerAPI/v1/nc/bac33f5949404741b3e5b90242fd78f5" />
    </links>
```


300248

Figure 4-8 Request Body Example for Updating a Tenant

```
<tenant xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns="http://www.cisco.com/NetworkServicesManager">
<uid>6a04fbe3e50c46848fc75c8bfc6f98ef</uid>
     <name>My Tenant Name</name>
     <description>My Tenant description</description>
     <version>15</version>
   <parameters>
      <parameter>
           <name>public.address.subnet</name>
           <type>subnet</type>
                 <subnet>
                    <ipv4>1.2.3.0</ipv4>
                    <mask>24</mask>
                 </subnet>
       </parameter>
       <parameter>
           <name>private.address.subnet</name>
           <type>subnet</type>
                 <subnet>
                    <ipv4>2.2.3.0</ipv4>
                    <mask>24</mask>
                 </subnet>
       </parameter>
</parameters>
</tenant>
```

300245

Identifying Created Objects

At times, you might find it useful to identify created objects. For example, you might want to view a list of all created tenants, or view the details of a specific tenant.

To identify created objects using a REST client, choose the **GET** request method, enter the appropriate URL (see Table 4-2), and click **Send**.

Table 4-2 Using the GET Request Method to View Created Objects

Resource	URL
All tenants	https://*hostname*:8443/NetworkServicesManagerAPI/v1/top/tenant
A specific tenant	https://*hostname*:8443/NetworkServicesManagerAPI/v1/tenant/*tenant-uid*
Tenant network container	https://*hostname*:8443/NetworkServicesManagerAPI/v1/tnc/*tnc-uid*
Network container	https://*hostname*:8443/NetworkServicesManagerAPI/v1/nc/*nc-uid*
ENC	https://*hostname*:8443/NetworkServicesManagerAPI/v1/enc/e*nc-uid*

Identifying Service Offerings

When creating new objects, you might find it useful to identify the available service offerings. For example, you might want to view the available types of tenants or network containers.

To view available service offerings using a REST client, choose the **GET** request method and the appropriate URL (see Table 4-3), and click **Send**.

Table 4-3 Using the GET Request Method to View Service Offerings

Resource	URL
Service offering details	https://*hostname*:8443/NetworkServicesManagerAPI/v1/serviceoffering/*offering-uid*[1]
Top level offerings	https://*hostname*:8443/NetworkServicesManagerAPI/v1/
Tenant types	https://*hostname*:8443/NetworkServicesManagerAPI/catalog
Tenants	https://*hostname*:8443/NetworkServicesManagerAPI/v1/tenant/*tenant-uid*/catalog
Tenant network containers	https://*hostname*:8443/NetworkServicesManagerAPI/v1/tnc/*tnc-uid*/catalog
Network containers	https://*hostname*:8443/NetworkServicesManagerAPI/v1/nc/*nc-uid*/catalog
ENCs	https://*hostname*:8443/NetworkServicesManagerAPI/v1/enc/e*nc-uid*/catalog
Pods	https://*hostname*:8443/NetworkServicesManagerAPI/v1/tenant/*tenant-uid*/pods

1. This URL is also the “self” link for the service offering.

Deleting Created Objects

If you no longer need an object, you can delete it. Deleting an object that contains other objects, such as a tenant or tenant network container, also deletes all child objects.

To delete created objects using a REST client, choose the **DELETE** request method, enter the appropriate URL (see Table 4-4), and click **Send**.

Table 4-4 Using the DELETE Request Method

Resource	URL
Tenant	https://*hostname*:8443/NetworkServicesManagerAPI/v1/tenant/*tenant-uid*
Tenant network container	https://*hostname*:8443/NetworkServicesManagerAPI/v1/tnc/*tnc-uid*
Network container	https://*hostname*:8443/NetworkServicesManagerAPI/v1/nc/*nc-uid*
ENC	https://*hostname*:8443/NetworkServicesManagerAPI/v1/enc/e*nc-uid*

CHAPTER 5

Using the Network Services Manager Administration UI

This chapter describes the Network Services Manager Administration UI and includes the following sections:

Getting Started with the Administration UI

To log into the Network Services Manager Administration UI:

Step 1 In your browser, enable popup windows for the Network Services Manager engine. If you do not enable popup windows, you cannot view confirmation dialog boxes or other messages that Network Services Manager displays.

Step 2 Enter the following URL:

`https://hostname:8443/`

where ***`hostname`*** is the name of the Network Services Manager engine.

Step 3 When prompted, accept the security certificate.

Step 4 In the login screen, enter the username and password. The default value is **`admin`** for both the username and password.

Note For security purposes, we recommend that you change the password. To change the login password, see Changing User Passwords, page 5-7.

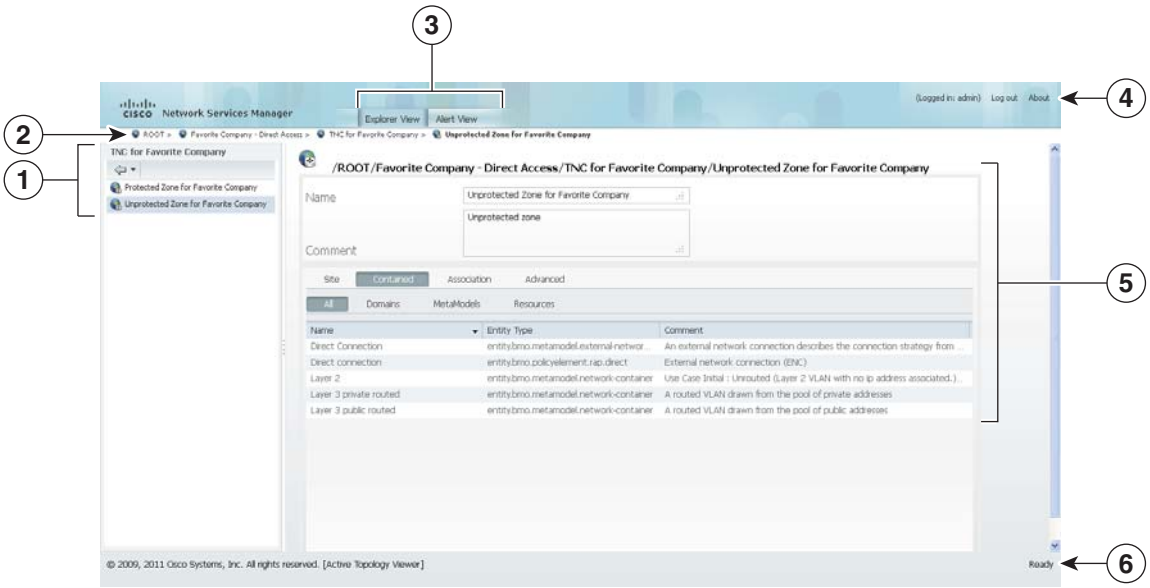
The main Administration screen is displayed with the ROOT domain selected.

Administration UI Features

When you log into Network Services Manager, the Administration UI is displayed as shown in Figure 5-1. The UI includes the following components:

- Explorer and Alert Views, page 5-2
- Domain Navigator, page 5-3
- Breadcrumb Trail, page 5-3
- Content Pane, page 5-3
- Status Bar, page 5-6
- Product Information, page 5-6

Figure 5-1 Administration UI



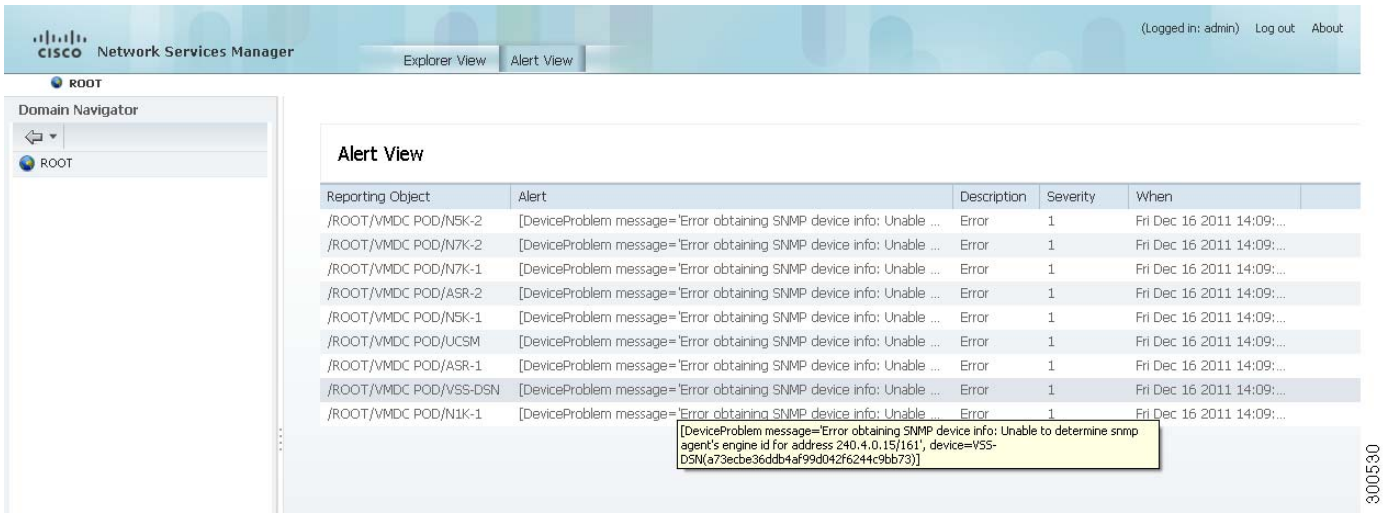
1	Domain Navigator	**4**	Product information
2	Breadcrumb trail	**5**	Content pane
3	Explorer View and Alert View tabs	**6**	Status bar

Explorer and Alert Views

The Administration UI includes the following views, each accessible via a tab at the top of the window:

- Explorer View—The Explorer View enables you to view objects, their properties, and their running configurations.
- Alert View—The Alert View (shown in Figure 5-2) displays a sortable list of system events, specific to the item selected in the Domain Navigator. You can resize the columns or hover your cursor over an entry in the table to see the complete text.

Figure 5-2 Alert View



Domain Navigator

The Domain Navigator displays the domain hierarchy in a dynamic view that allows you to view tenants and the associated objects.

When you log into Network Services Manager, the ROOT domain is selected by default. As you navigate the hierarchy, the breadcrumb trail is updated with the path, and the content pane is updated with the properties of the currently selected item.

The return arrow button at the top of the Domain Navigator pane enables you to return to a higher level in the domain hierarchy.

Tip Use the drop-down list next to the return arrow to select the level you want to return to.

Breadcrumb Trail

The breadcrumb trail identifies your recent navigation path and enables you to easily return to previously viewed screens. It also provides the currently displayed item's path, relative to the ROOT domain. If you select a new item in the Domain Navigator, the breadcrumb trail reflects the new path from ROOT to the selected item.

Content Pane

The content pane contains detailed information about the selected object. The top portion contains:

- The path from ROOT to the selected item.
- Name and Comment fields.

The lower portion of the content pane contains tabs with object-specific properties. The tabs that are displayed depend on the selected object and can contain subordinate tabs. Table 5-1 describes the tabs available for common objects.

Table 5-1 Content Pane Tabs

Tab	Description
ROOT, Tenant, and Tenant Network Container	
Contained	Contains the All, Subdomains, Sites, and Resources tabs. The All tab lists all contained entities, including the entity name, entity type, and a comment. The other tabs list the entities that fall into the identified category.
Metaproperties	Displayed for tenants and tenant network containers. Lists metaproperties associated with the selected object.
Advanced	Contains the Settings, Client Properties, and State tabs. The Advanced tab contains advanced settings for the selected object.
Pod	
Controller	Contains the username and password for the Network Services Manager controller for the selected device stack.
Assignments	Lists the local resources, VLANs, and zones that can be assigned to the device stack and those that are assigned.
Contained	Contains the All, Network Elements, and Interconnects tab. Identifies the network elements and interconnects associated with the selected object, including their entity type and any comments.
Advanced	Contains the Settings, Client Properties, and State tabs. The Advanced tab contains advanced settings for the selected object.
ENC	
General	Displays the ENC name and comment.
Settings	Identifies the start and end dates for the service, and any owned objects.
State Variables	Identifies any parameters defined for the ENC.
Metaproperties	Lists all metaproperties associated with the ENC.
Advanced	Contains the Settings, Client Properties, and State tabs. The Advanced tab contains advanced settings for the selected object.
Zone Network Container	
Site	Identifies the device stack assigned to the site.
Contained	Contains the All, Domains, Metamodels, and Resources tabs. The All tab lists all contained entities, including the entity name, the entity type, and a comment. Entity types can include created VLANs, network containers, and ENCs. The other tabs list the entities that fall into the identified category.

Table 5-1 Content Pane Tabs (continued)

Tab	Description
Association	Contains the Policy Elements and Groups tabs. The Policy Elements tab lists the available policy elements and those that are associated with the selected zone. The Groups tab lists the available groups and those that are associated with the selected zone.
Advanced	Contains the Settings, Client Properties, and State tabs. The Advanced tab contains advanced settings for the selected object.
VLAN	
General	Identifies whether or not an ACL enforcement policy is in effect, whether or not the VLAN is managed, the scope of the VLAN (for example, Layer 2 or Layer 3), and the device stack assigned to the site.
Subnet	Identifies the VLAN subnet and mask, the DHCP IP address if one is configured, the gateway IP address, and the type of IP address pool, such as private or public.
Association	Contains the Policy Elements and Groups tabs. The Policy Elements tab lists the available policy elements and those that are associated with the selected VLAN. The Groups tab lists the available groups and those that are associated with the selected VLAN.
Metaproperties	Lists metaproperties associated with the selected VLAN.
Advanced	Contains the Settings, Client Properties, and State tabs. The Advanced tab contains advanced settings for the selected object.
Group and Local Resource	
General	Displayed for local resources. Identifies the VLAN subnet with the subnet mask and the object that the local resource is associated with.
Members	Displayed for groups. Lists objects that are available to the group and those that participate in the group.
Association	Contains the Policy Elements and Groups tabs. The Policy Elements tab lists the available policy elements and those that are associated with the selected group or local resource. The Groups tab lists the available groups and those that are associated with the selected group or local resource.
Advanced	Contains the Settings, Client Properties, and State tabs. The Advanced tab contains advanced settings for the selected object.

Status Bar

The status bar indicates whether Network Services Manager is loading information or ready for your next selection.

Product Information

Product information is available at the top right of the window and includes:

Field	Description
Username	Account name of the person logged in.
Logout button	Logs you out of Network Services Manager after you confirm the logout request.
About button	Displays Network Services Manager version information.

Verifying Device Configurations

After you add a tenant network container and network containers, you can verify that the configuration on a device matches the information you entered by using the Run Commands feature.

To verify the device configuration in Network Services Manager:

Step 1 Navigate to the required device by choosing **ROOT > Contained > All**.

Step 2 Click **VMDC POD**.

Step 3 Choose **Contained > Network Elements**.

Step 4 Click the required device in the list of network elements.

Step 5 In the properties screen, click **Run Commands**.

Step 6 In the Run Command dialog box, choose the required Show option from the drop-down list and click **Run**.

The configuration is displayed in the dialog box (see Figure 5-3), allowing you to compare the information entered using Network Services Manager with the existing device configuration.

UI Example

Figure 5-3 shows an example device configuration in the Run Commands dialog box.

Figure 5-3 Device Configuration in Run Commands Dialog Box

Changing User Passwords

We recommend that you change user passwords for security purposes.

The following conventions apply when changing user passwords:

- The password must contain at least eight characters.
- The password must contain characters from three of the following groups:
  - Lowercase letters
  - Uppercase letters
  - Numbers
  - Special characters

If your organization requires different password policy settings, review and edit the passwordpolicy.properties file on the engine in the following directory:

/usr/local/overdrive/engine/bin/UtilUpdateUserPassword

To change the password for the Network Services Manager Administration UI or apiclient account:

Step 1 Log into the Network Services Manager engine from the vSphere console window.

Step 2 To enter the root shell, enter **shell**.

Step 3 Navigate to the correct directory by entering:

```
cd /usr/local/overdrive/engine/bin/UtilUpdateUserPassword
```

Step 4 To change a password, do either or both of the following:

- Administration UI password—Enter the following command:

  `java -jar UtilUpdateUserPassword.jar` *`old-password new-password`*

- Apiclient account password—Enter the following command:

  `java -Dusername=apiclient -jar UtilUpdateUserPassword.jar` *`old-password new-password`*

  where:

  - *`old-password`* is the current apiclient account password.
  - *`new-password`* is the new apiclient account password.

Step 5 Leave the root shell by entering **`exit`**.

Step 6 To update affected clients, do one or both of the following:

- Administration UI password—Close any browser windows that are logged into Network Services Manager using the old password, and log in again using the new password.
- Apliclient account password—Update any application using the apiclient account with the new password.

CHAPTER 6

Monitoring Status

This chapter describes how to monitor Network Services Manager status:

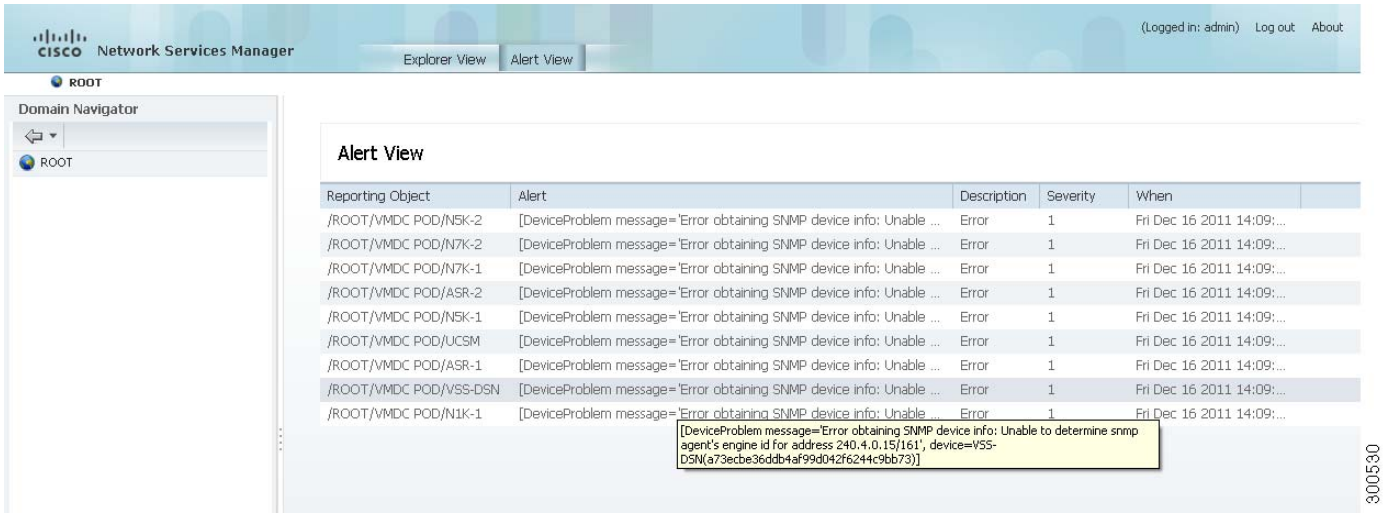
Monitoring with the Alert View

To monitor system alerts issued by Network Services Manager, click the **Alert View tab** at the top of the Administration UI window. The Alert View table displays a sortable list of system events, specific to the item selected in the Domain Navigator, and includes the following:

- Reporting object
- Full text of the alert
- Description of the alert
- Alert severity
- Date and time that the alert occurred

You can resize the table columns or hover your cursor over an entry in the table to see the complete text.

Figure 6-1 Alert View



Causes for alerts include:

- A controller has lost connection to the engine.
- A device has lost connection to the controller.
- A device is not responding to queries from a controller.
- A device fails to complete a controller command, such as adding a VLAN or retrieving a configuration.
- A tenant has lost connectivity.
- Configuration requests exceed metamodel limits.
- Credentials on a device have changed, so that the device is no longer reachable.
- Provisioning errors.

Use the information in the Alert View to identify the source of the alert and the problem.

Monitoring with a Syslog Host

You can configure Network Services Manager to direct all syslog messages to a syslog host for your review and analysis. To configure a syslog host for Network Services Manager, see the instructions in the *Cisco Network Services Manager 5.0 Quick Start Guide*.

CHAPTER 7

Troubleshooting

This chapter provides hints and suggestions for troubleshooting problems you might encounter while working with Network Services Manager:

Where to Look First

To view system events, such as error messages upon object creation or VLAN assignment, click the **Alert View tab**. The alerts displayed in the Alert View table are specific to the item selected in the Domain Navigator. For example, if a tenant is selected in the Domain Navigator, any alerts associated with that tenant are displayed in the Alert View table.

Checking Status

The following questions focus on checking status of devices, controller, communications, and so on.

Q. How can I communicate with the switches if the controller is down?

A. Connect to the controller through its serial line interface:

a. Verify that the controller is powered on.

b. Verify that the controller is plugged into the network.

c. Try to connect to the controller using Secure Shell (SSH).

d. Try to connect to the controller using the serial cable.

Q. I can't find the status change that I am looking for. How can I find it?

A. Look for other changes related to the one you are looking for and see if they took place. Confirm that the controller appliance and processes are both operational.

Troubleshooting Procedures

These questions and answers relate to procedures and how to perform certain actions.

Q. I changed the admin account password recently and do not remember the new password. What do I do?

A. You can restore the username and password to the default values by completing the following steps:

a. Log into the Network Services Manager engine from the vSphere console window for that VM.

b. To enter the root shell, enter **`shell`**.

c. Enter:

```
/usr/local/overdrive/engine/bin/resetPasswordToFactory.sh
```

d. When prompted, indicate whether you need to reset the password for the admin account or the apiclient account.

e. Leave the root shell by entering **`exit`**.

Q. Is there a Go Back button in the UI so I can return to the previous screen?

A. Yes, at the top of the Domain Navigator pane. You can also use the breadcrumb trail, which is displayed above the Domain Navigator. The breadcrumb trail tells you where the currently selected object resides in the domain hierarchy. The domain path at the top of the content pane provides another location indicator.

Troubleshooting Configurations

These questions and answers relate to configuration.

Q. How can I verify that the device is configured correctly after entering a change via the NB API?

A. You can verify that the configuration on a device matches the information that you entered using the NB API in either of the following ways:

– Logging into the device and viewing the running configuration.

– Logging into the Network Services Manager Administration UI and viewing the running configuration:

a. Navigate to the required device by choosing **ROOT > All**.

b. Click **VMDC POD**.

c. Choose **Contained > Network Elements**.

d. Click the required device.

e. In the properties screen, click **Run Commands**.

f. In the Run Command dialog box, choose the required Show option from the drop-down list and click **Run**.

Q. How many controllers can be at one site?

A. Network Services Manager supports one controller per site.

Q. How many devices can be assigned to one controller?

A. You can assign one device stack to a controller. The device stack can contain multiple devices, as described in Device Stack (Pod), page 2-4.

APPENDIX **A**

Northbound API Parameters

Table A-1 identifies parameters that can be issued via the NB API, along with the XML value types. Each value is wrapped in an element that identifies its type, as shown in the table.

Table A-1 NB API Parameters and XML Types

Value Type	XML Example	XML Value Type

Boolean	`<value>`
	`  <name>param1</name>`
	`  <boolean>true</boolean>`
	`</value>`
		boolean

date	`<value>`
	`  <name>param1</name>`
	`  <date>2011-03-04T12:34:45-08:00</date>`
	`</value>`
		date

integer	`<value>`
	`  <name>param1</name>`
	`  <integer>123</integer>`
	`</value>`
		integer

ipv4	`<value>`
	`  <name>param1</name>`
	`  <ipv4>10.1.2.3</ipv4>`
	`</value>`
		IPv4Value

ipv4Addresses	`<value>`
	`  <name>param1</name>`
	`  <ipv4Addresses>`
	`  <ipv4>10.1.2.3</ipv4>`
	`  <ipv4>10.1.2.4</ipv4>`
	`  </ipv4Addresses>`
	`</value>`
		IPv4ListValue

Table A-1 NB API Parameters and XML Types (continued)

Value Type	XML Example	XML Value Type

ipv4Pair	`<value>`
	`  <name>param1</name>`
	`  <ipv4Pair>`
	`  <ipv4Type>`
	`  <name>public</name>`
	`  <ipv4>0.0.0.0</ipv4>`
	`  </ipv4Type>`
	`  <ipv4Type>`
	`  <name>private</name>`
	`  <ipv4>1.1.1.1</ipv4>`
	`  </ipv4Type>`
	`  </ipv4Pair>`
	`</value>`
		IPv4PairValue

ipv4Pairs	`<value>`
	`  <name>param1</name>`
	`  <ipv4Pairs>`
	`  <ipv4Pair>`
	`  <ipv4Type>`
	`  <name>public</name>`
	`  <ipv4>0.0.0.0</ipv4>`
	`  </ipv4Type>`
	`  <ipv4Type>`
	`  <name>private</name>`
	`  <ipv4>1.1.1.1</ipv4>`
	`  </ipv4Type>`
	`  </ipv4Pair>`
	`  <ipv4Pair>`
	`  <ipv4Type>`
	`  <name>three</name>`
	`  <ipv4>10.0.0.3</ipv4>`
	`  </ipv4Type>`
	`  <ipv4Type>`
	`  <name>four</name>`
	`  <ipv4>10.0.0.4</ipv4>`
	`  </ipv4Type>`
	`  </ipv4Pair>`
	`  </ipv4Pairs>`
	`</value>`
		IPv4PairListValue

ipv4Type	`<value>`
	`  <name>param1</name>`
	`  <ipv4Type>`
	`  <name>public</name>`
	`  <ipv4>0.0.0.0</ipv4>`
	`  </ipv4Type>`
	`</value>`
		IPv4TypeValue

macAddress	`<value>`
	`  <name>param1</name>`
	`  <macAddress>01:23:45:67:89:1a</macAddress>`
	`</value>`
		MACAddressValue

Table A-1 NB API Parameters and XML Types (continued)

Value Type	XML Example	XML Value Type

qosPolicies	`<value>`
	`  <name>param1</name>`
	`  <qosPolicies>`
	`  <qosPolicy>`
	`  <inPolicy>InPolicyName</inPolicy>`
	`  <outPolicy>OutPolicyName</outPolicy>`
	`  </qosPolicy>`
	`  </qosPolicies>`
	`</value>`
		QosPolicyListValue

qosPolicy	`<value>`
	`  <name>param1</name>`
	`  <qosPolicy>`
	`  <inPolicy>InPolicyName</inPolicy>`
	`  <outPolicy>OutPolicyName</outPolicy>`
	`  </qosPolicy>`
	`</value>`
		QosPolicyValue

range	`<value>`
	`  <name>param1</name>`
	`  <range>`
	`  <start>123</start>`
	`  <end>345</end>`
	`  </range>`
	`</value>`
		RangeValue

ranges	`<value>`
	`  <name>param1</name>`
	`  <ranges>`
	`  <range>`
	`  <start>123</start>`
	`  <end>135</end>`
	`  </range>`
	`  <range>`
	`  <start>324</start>`
	`  <end>345</end>`
	`  </range>`
	`  </ranges>`
	`</value>`
		RangeListValue

scheduledDate	`<value>`
	`  <name>param1</name>`
	`  <scheduledDate>`
	`  <!-- choices are IMMEDIATE or NEVER -->`
	`  <schedule>NEVER</schedule>`
	`  </scheduledDate>`
	`</value>`
		ScheduledDateValue

string	`<value>`
	`  <name>param1</name>`
	`  <string>abc 123</string>`
	`</value>`
		string

Table A-1 NB API Parameters and XML Types (continued)

Value Type	XML Example	XML Value Type

strings	`<value>`
	`  <name>param1</name>`
	`  <strings>`
	`  <string>abc 123</string>`
	`  <string>def 678</string>`
	`  </strings>`
	`</value>`
		StringListValue

subnet	`<value>`
	`  <name>param1</name>`
	`  <subnet>`
	`  <ipv4>10.0.0.0</ipv4>`
	`  <mask>8</mask>`
	`  </subnet>`
	`</value>`
		SubnetValue

subnets	`<value>`
	`  <name>param1</name>`
	`  <subnets>`
	`  <subnet>`
	`  <ipv4>10.0.0.0</ipv4>`
	`  <mask>8</mask>`
	`  </subnet>`
	`  <subnet>`
	`  <ipv4>192.168.1.0</ipv4>`
	`  <mask>24</mask>`
	`  </subnet>`
	`  </subnets>`
	`</value>`
		SubnetListValue

uid	`<value>`
	`  <name>param1</name>`
	`  <uid>`*tenant-001*`</uid>`
	`</value>`
		uid

vlanAssignment	`<value>`
	`  <name>param1</name>`
	`  <vlanAssignment>`
	`  <subnet>`
	`  <ipv4>192.168.1.0</ipv4>`
	`  <mask>24</mask>`
	`  </subnet>`
	`  <vlan>123</vlan>`
	`  </vlanAssignment>`
	`</value>`
		VlanAssignmentValue

vmAddress	`<value>`
	`  <name>param1</name>`
	`  <vmAddress>`
	`  <MAC>12:34:56:78:90:AB</MAC>`
	`  <ipv4>192.168.0.0</ipv4>`
	`  <fqdn>testhost.cisco.com</fqdn>`
	`  </vmAddress>`
	`</value>`
		VMAddressValue

GLOSSARY

A

access control list (ACL) A list of permissions to objects; for example, read, write, and delete permissions for users or system processes.

access switch Managed switches that work at the desktop layer and that connect workstations and servers to the network. Access switches also provide MAC address filtering, bandwidth sharing, and bandwidth switching (moving data from one network to another). A virtual distributed switch such as a Cisco Nexus 1000 is also an access switch.

aggregation switch A switch that provides aggregate or group networks.

B

business model A representation of all of the resources, network topology, services, and business policies that are managed by Network Services Manager.

The business model includes the topology and can include cloud metamodels.

business policy A mechanism for providing network connectivity in terms of resources, ports, protocols, schedules, and connection topologies.

C

connection topology A network configuration that allows resources or collections of resources to communicate in a specific arrangement: all together, individually to all others (full mesh, bidirectional); server-to-client (spoke-initiated hub and spoke, or peer to peer); hub and spoke but bidirectional, such as for remote desktop help; or a single peer-to-peer pair, whether peer-initiated or bidirectional.

controller The Network Services Manager component that manages the stack of devices associated with a logical site. Multiple controllers can run on a single host, allowing you to manage multiple sites from a single physical server.

D

device Networking hardware such as a switch or router.

device stack A collection of network elements (devices) representing a logical site. A stack of devices configured to manage cloud operations is referred to as a *pod* in Network Services Manager.

distribution switch A device working at the workgroup or distribution layer (including LAN-based routers and Layer 3 switches), that ensures that packets are routed between subnets and VLANs.

domain An organizational structure in Network Services Manager that is used to delegate administrative control and to provide managerial separation for tenants. A domain can contain all of the business model objects that are used to manage clouds.

dynamic topology The process of retrieving abstract information from a business model, converting that abstract information into device-specific configurations, and implementing those configurations on the appropriate devices.

E

engine The Network Services Manager component that analyzes business and network access policies, compares the existing topology to the specified policies, and requests the controllers to reconfigure the devices they control.

L

logical architecture An abstract representation of the network architecture that a tenant can build by using the building blocks provided for cloud operations.

logical topology An abstract method that a controller uses to provision devices by specifying how more primitive provisioning elements are to be combined to implement data paths and policy.

A logical topology specifies the manner and type of configuration that needs to occur at each transition point in the physical topology, and how services should be organized relative to each other and to devices in the physical topology.

M

metamodel The definition of specific entities and resources that can exist in a network, and the relationships that exist among these entities and resources.

N

network access policy The method used to define entitlement and access management on a network.

network container A logical group of virtual network resources that are created and managed as a unit.

P

pod See device stack.

policy A statement of intent for access to resources or the network. Network Services Manager uses policies for comparison when analyzing the network topology and identifying the adjustments that must be made to the topology.

ports and protocols The TCP/UDP port or IP protocol that is permitted in a policy. Network Services Manager has predefined a number of standard ports, such as HTTP, Telnet, and ICMP. The predefined object ANY allows network communications on any port and protocol.

R

resource A host or a network subnet with an IP address and with a subnet mask that is assigned to a single site. Resources can be moved from one site to another, and can include local resources, groups, VLANs, and network identities.

S

site A logical collection of devices. More than one Network Services Manager site can exist at a physical location.

subnet Networked computers and devices with a common IP routing prefix such as 192.168.

T

tenant A domain that contains business model objects that set up and manipulate one or more logical network containers.

tenant network container A metamodel that identifies the device stack deployed for a tenant at a particular site.

U

URI Uniform Resource Identifier. A string of characters used to identify a name or a resource on the Internet. In Network Services Manager, a task resource is assigned a URI as a unique identifier for tracking purposes.

V

virtual machine (VM)

A computer environment such as those provided by VMware that allows one operating system to run on a host operating system as if it were standalone.

VLAN

An abstraction of a switch VLAN that can be managed as a Layer 2 VLAN or a routed VLAN (with an SVI). In Network Services Manager, VLANs are managed as resources from a pool and allocated dynamically for tenant networks. Depending on the need, the VLAN is created on all relevant devices to complete a data path for the tenant in the device stack.

INDEX

A

B

C

N

O

S

T

U

V

Z