vCloud API Programming Guide

vCloud Director 1.5

This document supports the version of each product listed and supports all subsequent versions until the document is replaced by a new edition. To check for more recent editions of this document, see http://www.vmware.com/support/pubs.

EN-000579-00

You can find the most up-to-date technical documentation on the VMware Web site at:

http://www.vmware.com/support/

The VMware Web site also provides the latest product updates.

If you have comments about this documentation, submit your feedback to:

docfeedback@vmware.com

VMware, Inc.
3401 Hillview Ave.
Palo Alto, CA 94304
www.vmware.com

Contents

vCloud API Programming Guide

The *vCloud API Programming Guide* provides information about version 1.5 of the vCloud API.

VMware provides many different APIs and SDKs for applications and goals. This guide provides information about the vCloud API for developers who are interested in creating RESTful clients of VMware vCloud Director.

Revision History

The *vCloud API Programming Guide* is revised with each release of the product or when necessary. A revised version can contain minor or major changes.

Table 1. Revision History

Revision Date	Description
01SEP11	API Version 1.5
30AUG10	API Version 1.0
14APR10	API Version 0.9

Intended Audience

This guide is intended for software developers who are building VMware Ready Cloud Services, including interactive clients of VMware vCloud Director. This guide discusses Representational State Transfer (REST) and RESTful programming conventions, the Open Virtualization Format Specification, and VMware Virtual machine technology. You must be familiar with these and other widely deployed technologies such as XML, HTTP, and the Windows or Linux operating system.

Related Publications

The *VMware vCloud Director Administrator's Guide* and *VMware vCloud Director User's Guide* contain detailed information about many of the objects and operations referred to in this guide. Most users of the vCloud API will find the information in those documents valuable when developing client applications. To access the current versions of these and other VMware books, go to http://www.vmware.com/support/pubs.

About the VMware vCloud API 1

The VMware vCloud API provides support for developers who are building interactive clients of VMware vCloud Director using a RESTful application development style.

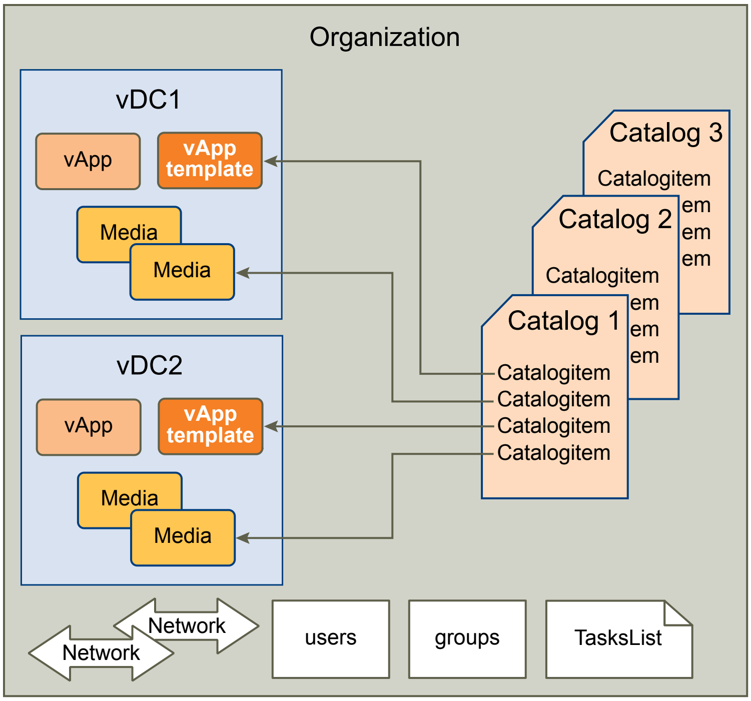
vCloud API clients and vCloud Director servers communicate over HTTP, exchanging representations of vCloud objects. These representations take the form of XML elements. You use HTTP GET requests to retrieve the current representation of an object, HTTP POST and PUT requests to create or modify an object, and HTTP DELETE requests to delete an object.

This chapter includes the following topics:

Object Taxonomy

The vCloud API defines a set of objects common to cloud computing environments. An understanding of these objects, their properties, and their relationships is essential to using the vCloud API.

Figure 1-1. vCloud API Object Taxonomy



vCloud API objects have the following high-level properties:

Organizations

A cloud can contain one or more organizations. Each organization is a unit of administration for a collection of users, groups, and computing resources. Users authenticate at the organization level, supplying credentials established when the user was created or imported.

Users and Groups

An organization can contain an arbitrary number of users and groups. Users can be created by the organization administrator or imported from an LDAP directory service. Groups must be imported from the directory service. Permissions within an organization are controlled through the assignment of rights and roles to users and groups.

Catalogs

Catalogs contain references to virtual systems and media images. A catalog can be shared to make it visible to other members of an organization, and can be published to make it visible to administrators in other organizations. A system administrator specifies which organizations can publish catalogs, and an organization administrator controls access to catalogs by organization members.

Networks

An organization can be provisioned with one or more networks. These organization networks can be configured to provide services such as DHCP, NAT, VPN, and firewalls.

Virtual Datacenters

A virtual datacenter (vDC) is a deployment environment for virtual systems and an allocation mechanism for resources such as networks, storage, CPU, and memory. In a vDC, computing resources are fully virtualized, and can be allocated based on demand, service level requirements, or a combination of the two.

Virtual Systems and Media Images

Virtual systems and media images are stored in a vDC and can be included in a catalog. Media images are stored in their native representation (ISO or floppy). Virtual systems are initially stored as templates, using an open standard format (OVF 1.0). These templates can be retrieved from catalogs and transformed into virtual systems, called vApps, through a process called instantiation, which binds a template's abstract resource requirements to resources available in a vDC. A vApp contains one or more individual virtual machines (`Vm` elements), along with parameters that define operational details:

- How the contained virtual machines are connected to each other and to external networks.
- The order in which individual virtual machines are powered on or off.
- End-user license agreement terms for each virtual machine.
- Deployment lease terms, typically inherited from the containing organization, that constrain the consumption of vDC resources by the vApp.
- Access control information specifying which users and groups can perform operations such as deploy, power on, modify, and suspend on the vApp and the virtual machines that it contains.

Tasks

Asynchronous operations that members of an organization initiate are tracked by task objects, which are kept on the organization's tasks list.

Objects, References, and Representations

The vCloud API represents objects as XML documents in which object properties are encoded as elements and attributes with typed values and an explicit object hierarchy defined by an XML schema.

XML representations of first-class vCloud API objects, such as the objects in Figure 1-1, include these attributes.

id

The object identifier, expressed in URN format. The value of the `id` attribute uniquely identifies the object, persists for the life of the object, and is never reused. The `id` attribute value is intended to provide a context-free identifier that can be used with the vCloud API `entityResolver` (see "Retrieve an Object as an Entity," on page 237), and is also suitable for use by clients that need to access the object using a different API.

type

The object type, specified as a MIME content type.

href

An object reference, expressed in URL format. Because this URL includes the object identifier portion of the `id` attribute value, it uniquely identifies the object, persists for the life of the object, and is never reused. The value of the `href` attribute is a reference to a view of the object, and can be used to access a representation of the object that is valid in a particular context. Although URLs have a well-known syntax and a well-understood interpretation, a client should treat each `href` as an opaque string. The rules that govern how the server constructs `href` strings might change in future releases.

Example: Object id, type, and href Attributes

This XML fragment, extracted from the representation of a vApp, shows its `id`, `type`, and `href` attributes.

```
<VApp
   ...
   id="urn:vcloud:vapp:490af534-1491-452e-8ed6-a5eb54447dac"
   type="application/vnd.vmware.vcloud.vApp+xml"
   href="https://vcloud.example.com/api/vApp/vapp-490af534-1491-452e-8ed6-a5eb54447dac"
   ... >
   ...
</VApp>
```

Links and Link Relations

The vCloud API makes extensive use of `Link` elements to provide references to objects and the actions that they support. These elements are the primary mechanism by which a server tells a client how to access and operate on an object.

The server creates `Link` elements in a response body. They are read-only at the client. If a request body includes a `Link` element, the server ignores it.

Attributes of a Link Element

In the XML representation of a vCloud object, each `Link` element has the following form:

```
<Link rel="relationship"
   type="application/vnd.vmware.vcloud.object_type+xml"
   href="URL"
   name="string"/>
```

Attribute values in a `Link` element supply the following information:

rel

Defines the relationship of the link to the object that contains it. A relationship can be the name of an operation on the object, a reference to a contained or containing object, or a reference to an alternate representation of the object. The relationship value implies the HTTP verb to use when you use the link's `href` value as a request URL.

type

The object type, specified as a MIME content type, of the object that the link references. This attribute is present only for links to objects. It is not present for links to actions.

href

An object reference, expressed in URL format. Because this URL includes the object identifier portion of the `id` attribute value, it uniquely identifies the object, persists for the life of the object, and is never reused. The value of the `href` attribute is a reference to a view of the object, and can be used to access a representation of the object that is valid in a particular context. Although URLs have a well-known syntax and a well-understood interpretation, a client should treat each `href` as an opaque string. The rules that govern how the server constructs `href` strings might change in future releases.

name

The name of the referenced object, taken from the value of that object's `name` attribute. Action links do not include a `name` attribute.

Table 1-1. Link Relationships and HTTP Request Types

rel Attribute Value	Action or Relationship Description	Implied HTTP Verb
add	Add an item to this container.	POST
alternate	References an alternate representation of this object.	GET
catalogItem	References the `CatalogItem` object that refers to this object.	GET
collaboration:abort	Abort this blocking task.	POST
collaboration:fail	Fail this blocking task.	POST
collaboration:resume	Resume this blocking task.	POST
consolidate	Consolidate this virtual machine.	POST
controlAccess	Apply access controls.	POST
copy	Reserved, unimplemented.	N/A
deploy	Deploy this vApp.	POST
disable	Disable this object.	POST
discardState	Discard the suspended state of this virtual machine.	POST
down	References an object contained by this object.	GET
download:alternate	Reserved, unimplemented.	N/A
download:default	References the default location from which this file can be downloaded.	GET
edit	Modify this object.	PUT
enable	Enable this object.	POST
firstPage	Reference to the first page of a paginated response.	GET
installVmwareTools	Install VMware Tools on this virtual machine.	POST
lastPage	Reference to the last page of a paginated response.	GET
media:ejectMedia	Eject virtual media from a virtual device.	POST
media:insertMedia	Insert virtual media into a virtual device.	POST
move	Reserved, unimplemented.	N/A
nextPage	Reference to the next page of a paginated response.	GET
ova	Reserved, unimplemented	N/A
ovf	References the OVF descriptor of this vApp template.	GET
power:powerOff	Power off this vApp or virtual machine.	POST
power:powerOn	Power on this vApp or virtual machine.	POST
power:reboot	Reboot this vApp or virtual machine.	POST
power:reset	Reset this vApp or virtual machine.	POST

Table 1-1. Link Relationships and HTTP Request Types (Continued)

rel Attribute Value	Action or Relationship Description	Implied HTTP Verb
power:shutdown	Shut down this vApp or virtual machine.	POST
power:suspend	Suspend this vApp or virtual machine.	POST
previousPage	Reference to the previous page of a paginated response.	GET
publish	Publish this catalog.	POST
recompose	Recompose this vApp.	POST
reconnect	Reconnect this vCenter Server to this cloud.	POST
register	Register a VCenter Server to this cloud.	POST
reject	Reject this request.	POST
relocate	Relocate this virtual machine.	POST
remove	Remove this object.	DELETE
repair	Repair this ESX/ESXi host.	POST
screen:acquireTicket	Retrieve a screen ticket for this virtual machine.	GET
screen:thumbnail	Retrieve a thumbnail view of the screen of this virtual machine.	GET
task:cancel	Cancel this task.	POST
blockingTask	A list of pending blocking task requests in this cloud.	GET
taskOwner	Reference to the owner of a task	GET
taskParams	Reference to the request parameters of a task	GET
taskRequest	Reference to the request associated with a task	GET
undeploy	Undeploy this vApp.	POST
unlock	Unlock a user account	POST
unregister	Unregister this vCenter Server.	POST
up	References an object that contains this object.	GET
updateProgress	Request an update of this task's progress.	POST
upgrade	Upgrade this ESX/ESXi host.	POST
upload:alternate	Reserved, unimplemented.	N/A
upload:default	References the default location to which this object can be uploaded.	PUT

Client Workflow Overview

vCloud API clients implement a RESTful workflow, making HTTP requests to the server and retrieving the information they need from the server's responses.

About RESTful Workflows

REST, an acronym for Representational State Transfer, describes an architectural style characteristic of programs that rely on the inherent properties of hypermedia to create and modify the state of an object whose serialized representation is accessible at a URL.

If a URL of such an object is known to a client, the client can use an HTTP GET request to retrieve the representation of the object. In the vCloud API, this representation is an XML document. In a RESTful workflow, documents that represent of object state are passed back and forth between a client and a service with the explicit assumption that neither party need know anything about an object other than what is presented in a single request or response. The URLs at which these documents are available often persist beyond the lifetime of the request or response that includes them. The other content of the documents is nominally valid until the expiration date noted in the HTTP `Expires` header.

vCloud REST API Workflows

Application programs written to a REST API use HTTP requests that are often executed by a script or other higher-level language to make remote procedure calls that create, retrieve, update, or delete objects that the API defines. In the vCloud REST API, these objects are defined by a collection of XML schemas. The operations themselves are HTTP requests, and so are generic to all HTTP clients.

To write a RESTful client, you must understand only the HTTP protocol and the semantics of XML, the transfer format that the vCloud API uses. To use the vCloud API effectively in such a client, you need to know only a few things:

- What is the set of objects that the API supports, and what do they represent. For example, what is a vDC and how does it relate to an organization or catalog?
- How does the API represents these objects. For example, what does the XML schema for an `Org` look like? What do the individual elements and attributes represent?
- How does the client refer to an object on which it wants to operate. For example, where are the links to objects in a vDC? How does a client obtain and use them?

You can find this information in the vCloud API XML schemas. The XML elements, attributes, and composition rules defined in these schemas and represent the data structures of objects in the cloud. A client can read an object by making an HTTP GET request to the object's URL. A client can create or modify an object with an HTTP PUT or POST request that includes a new or changed XML body document for the object. A client can usually delete an object with an HTTP DELETE request.

The vCloud API schema reference includes detailed information about the XML representations of all vCloud API objects and examples of HTTP requests that operate on those objects. See "About the Schema Reference Downloadable Archive," on page 19.

RESTful Workflow Patterns

All RESTful workflows follow a common pattern.

1 Make an HTTP request, typically GET, PUT, POST, or DELETE. The target of this request is either a well-known URL such as a the vCloud API `versions` URL, or a URL obtained from the response to a previous request. For example, a GET request to an organization URL returns links to catalog and vDC objects that the organization contains.

2 Examine the response, which always includes an HTTP response code and usually includes a body. In the vCloud API, a response body is an XML representation of an object, including elements and attributes that represent object properties, links that implement operations on the object or provide references to contained or containing objects and, if the object is being created or modified, an embedded task object that tracks the progress of the creation or modification. The response also includes an HTTP response code, which indicates whether the request succeeded or failed, and might be accompanied by a URL that points to a location from which you can retrieve additional information.

These operations can repeat, in this order, for as long as necessary.

vCloud API REST Requests

To retrieve object representations, clients make HTTP requests to object references. The server supplies these references as `href` attribute values in responses to GET requests.

Every cloud has a well-known URL from which an unauthenticated user can retrieve a list of vCloud API versions that the server supports. Each version has its own login URL. A system administrator can use that URL to authenticate to the cloud by logging in to the `System` organization. An authenticated user can discover other vCloud API URLs by making GET requests to URLs retrieved from the login response, and the URLs contained in responses to those requests. See Chapter 3, “Exploring a Cloud,” on page 41.

Requests are typically categorized in terms of the type of requested operation: create, retrieve, update, and delete. This sequence of verbs is often abbreviated with the acronym CRUD.

Table 1-2. CRUD Operations Summary

Operation Type	HTTP Verb	Operation Summary
Create	POST	Creates a new object.
Retrieve	GET	Retrieves the representation of an existing object.
Update	PUT	Modifies an existing object.
Delete	DELETE	Deletes an existing object.

Authentication

HTTP communications between a vCloud API client and server are secured with SSL. The vCloud API also implements Basic HTTP Authentication, as defined by RFC 2617, which enables a client to authenticate individual HTTP requests by including an authentication header in the request. See “Logging In,” on page 24.

Request Headers

The following HTTP headers are typically included in vCloud API requests:

Accept

All requests must include an HTTP `Accept` header that designates the XML namespace in which to process the request. The following header indicates that the request is to be processed in the vCloud API 1.5 namespace:

```
Accept: application/*+xml;version=1.5
```

Accept-Encoding

By default, vCloud Director returns response content as uncompressed XML. Compressing the response can improve performance, especially when the response is large and network bandwidth is a factor. To request a response to be returned as compressed XML, include the following header:

```
Accept-Encoding: gzip
```

The response is encoded using `gzip` encoding as described in RFC 1952, and includes the following header:

```
Content-Encoding: gzip
```

In the default configuration, responses smaller than 64KB are never compressed.

Authorization

All requests from authenticated clients must include an `Authorization` header. See “Logging In,” on page 24 for details about the value of this header.

Content-Type

Requests that include a body must start with the appropriate HTTP `Content-Type` header. Content types for all elements are included in the schema reference. In addition, the `type` attribute of a response body indicates the content type of the document. For example, this response fragment indicates that the content type associated with a `CatalogItem` object is `application/vnd.vmware.vcloud.catalogItem+xml`.

```
<CatalogItem
   type="application/vnd.vmware.vcloud.catalogItem+xml"
   href="https://vcloud.example.com/api/catalogItem/221"
   name="Ubuntu Template with vsftpd"/>
```

A POST or PUT request that supplies a `CatalogItem` in the request body requires the following `Content-Type` header:

```
Content-Type: application/vnd.vmware.vcloud.catalogItem+xml
```

When it appears as the value of a `Content-Type` header or the `type` attribute of an element in the vCloud API, this string is case-insensitive in requests, and can be returned in either mixed case or lowercase characters in responses.

Request Bodies

vCloud Director uses a validating XML parser that requires elements in a request body to agree with the schema in order and number. Request bodies are rejected as invalid unless they meet the following criteria:

- XML namespace attributes must be supplied for all namespaces represented by elements in the request. See “XML Namespace Identifiers,” on page 234.
- If multiple namespaces are represented in the request, XML namespace attributes must include an identifying prefix, and that prefix must be used with all elements from that namespace.
- All required elements must appear in request bodies. All elements that appear in request bodies must appear in the order that the schema establishes, and with content that conforms to the type constraint that the schema specifies.

vCloud API REST Responses

All responses include an HTTP status code and, unless the status code is 204 (No Content), a Content-Type header. Response content depends on the request. Some responses include a document body, some include only a URL, and some are empty.

A vCloud API client can expect a subset of HTTP status codes in a response.

Table 1-3. HTTP Status Codes that the vCloud API Returns

Status Code	Status Description
200 OK	The request is valid and was completed. The response includes a document body.
201 Created	The request is valid. The requested object was created and can be found at the URL specified in the Location header.
202 Accepted	The request is valid and a task was created to handle it. This response is usually accompanied by a `Task` element.
204 No Content	The request is valid and was completed. The response does not include a body.
303 See Other	The response to the request can be found at the URL specified in the `Location` header.
400 Bad Request	The request body is malformed, incomplete, or otherwise invalid.
401 Unauthorized	An authorization header was expected but not found.
403 Forbidden	The requesting user does not have adequate privileges to access one or more objects specified in the request.
404 Not Found	One or more objects specified in the request could not be found in the specified container.
405 Method Not Allowed	The HTTP method specified in the request is not supported for this object.
500 Internal Server Error	The request was received but could not be completed because of an internal error at the server.
501 Not Implemented	The server does not implement the request.
503 Service Unavailable	One or more services needed to complete the request are not available on the server.

Using the vCloud API with vCloud Director

VMware vCloud Director 1.5 supports version 1.5 of the vCloud API. You can use a browser or other HTTP client program to send requests and receive responses.

The vCloud Director REST API Reference documentation includes HTML reference material for all XML elements and complex types defined by the vCloud API. It also includes examples XML representations. See "About the Schema Reference Downloadable Archive," on page 19. For information about HTTP client programs to use with vCloud Director, see "REST Client Programs," on page 19.

Procedure

1 Configure the vCloud Director REST API base URL.

   a Log in to the vCloud Director Web Console as a system administrator.

   b In the vCloud Director Web Console, open **System Settings > Public Addresses**

   c Type the URL in the **VCD public REST API base URL** text box.

2 Download the schema reference.

   The schema reference is an essential supplement to the *vCloud API Programming Guide*.

What to do next

Decide on an HTTP client program to use. See "REST Client Programs," on page 19.

REST Client Programs

You can use the vCloud API with any browser or client application that supports HTTP and SSL.

Any client application that can send HTTP requests over a secure channel by using SSL can be an appropriate tool for developing RESTful applications with the vCloud API. The examples in this information were developed using two open-source programs: cURL (http://curl.haxx.se/) and the RESTclient (http://code.google.com/p/rest-client/).

VMware provides additional SDK products that implement language-specific bindings for the vCloud API, and include their own HTTP client capability. See http://communities.vmware.com/community/developer/forums.

About the Schema Reference Downloadable Archive

XML schema reference documentation in HTML format for the vCloud API is available as a downloadable archive. This archive also includes the schema definition files, and examples XML representations of vCloud API objects.

To use the reference documentation:

1 Download the compressed archive from http://www.vmware.com/support/vcd/doc/rest-api-doc-1.5-html.zip

2 Uncompress the archive into any convenient folder.

3 In the folder, open the file `index.html` in a browser.

How the Schema Reference Documentation is Organized

The schema reference documentation is organized to reflect the division of the vCloud API into user, administrator, and extension categories. Within each category, you can open a list of elements, types that the elements extend, and operations that create, retrieve, update, or delete the objects that the elements represent.

User Operations, Elements, and Types

These operations are performed by all users who have permission to log into an organization. User elements and user types represent the objects that these operations manipulate. See Chapter 3, "Exploring a Cloud," on page 41, Chapter 4, "Provisioning an Organization with vApps, Templates, and Media," on page 53, Chapter 5, "Deploying and Operating vApps," on page 75, and Chapter 2, "Hello vCloud: A Simplified RESTful Workflow," on page 23.

Administrator Operations, Elements, and Types

These operations are performed by organization administrators or system administrators. Administrator elements and types represent the objects that these operations manipulate. See Chapter 6, "Creating, Provisioning, and Managing Organizations," on page 109.

Extension Operations, Elements, and Types

These operations are performed by system administrators who need access to vSphere platform objects from the vCloud API. Extension elements and types represent the objects that these operations manipulate. See Chapter 7, "Using vCloud API Extensions to Provision and Manage a Cloud," on page 161.

Searching In a Category

You can enter a search string in the Quick Index text box to search the lists of operations, elements, and types in any category.

- In an **Operations** list, you can search for the following items:
  - All or part of the name of the object on which you want to operate. The search returns a list of all of the operations that are possible on that object. For example, selecting User Operations and typing **`vApp`** in the Quick Index text box returns a list of all of the requests that operate on a vApp object.
  - The name of an action to perform. For example, selecting User Operations and typing **`power`** in the Quick Index text box returns a list of all the requests that change the power state of a vApp.
  - An HTTP verb (GET, PUT, POST, DELETE) to view a list of all the requests that use that verb. For example, selecting User Operations and typing **`PUT`** in the Quick Index text box returns a list of all of the requests that update an object.
- In an **Elements** or **Types** list, type all or part of the element or type name.

Search terms are not case-sensitive.

Operation Summary Syntax

Operations consist of an HTTP verb and a request URL. The reference documentation represents the verb and the URL using the following syntax:

HTTP_VERB /object_type/{id}[/action/*action_name*]

In this syntax, the initial `/` character is assumed to follow a site-specific API URL, such as `https://vcloud.example.com/api`. The following strings represent variables in the remainder of the URL:

HTTP_VERB The HTTP verb used to request the operation.

object_type An abbreviation of the MIME type of the object referenced by the operation. This abbreviation is constructed from the final component of the object's media type, between the `.` and the `+xml` designation. For example, for an object whose media type is `application/vnd.vmware.vcloud.catalogItem+xml`, the *`object_type`* is shown as `catalogItem`.

{id} The unique identifier of the object of the operation.

action_name The name of an action. Required only when the operation request URL includes the string `/action/`.

Element and Type Reference Pages

For each element or complex type, the reference documentation provides a page that lists the following items:

Element The name of the element.

Type The name of the type that the element extends.

Namespace The XML namespace in which this element or type name is defined.

Description A description of the purpose and contents of the element or type.

Since The vCloud API version in which this element or type first appeared.

Schema The name of the XML schema definition file in which this element or type is defined. Click to open the file in your browser, or right-click to download it.

Media Type The MIME type associated with this element or type.

Extends

The base type from which this element is derived.

XML Representation

The XML representation of the element or type. Names of contained elements are links to the reference pages for those elements.

Attributes

A table listing the following properties of each attribute of the element or type:

- **Attribute**

  The name of the attribute.

- **Type**

  The primitive XML type of the attribute.

- **Required**

  `Yes` for attributes that are required. `No` for attributes that are optional.

- **Modifiable**

  A value of `always` means that a client request can modify the value of this attribute. A value of `create` means that this attribute can be set or modified only as part of object creation. A value of `none` means that this attribute is read-only.

- **Since**

  The vCloud API version in which this attribute first appeared.

- **Description**

  A description of the purpose and contents of the attribute.

Elements

A table listing the following properties of each element defined in the type:

- **Element**

  The name of the element.

- **Type**

  A link to the definition of the complex type that the element is based on.

- **Occurrence**

  The occurrence constraint for the element. The constraint can be one of the following expressions:

  - **0..***

    Optional. Can occur zero or more times.

  - **0..1**

    Optional. Can occur at most once.

  - **1**

    Required. Must occur exactly once.

- **Modifiable**

  A value of `always` means that a client request can modify the contents of this element. A value of `create` means that element contents can be set or modified only as part of object creation. A value of `none` means that this element is read-only.

- **Since**

  The vCloud API version in which this element first appeared.

- **Description**

  A description of the purpose and contents of the element.

Operations

A summary of the operations permitted on the element. Operations are categorized by request type; one of create, retrieve, update, and delete. This sequence of verbs is often abbreviated with the acronym CRUD.

Schema Definition Files

XML schema definition files (`*.xsd`) are included in the `etc` folder of schema reference downloadable archive. This folder contains several subfolders:

1.0	Schema definition files for vCloud API version 1.0.
1.5	Schema definition files for vCloud API version 1.5.
schemas	Additional schema definition files that are version-independent or from external sources such as DMTF.

About the vCloud API Examples

The *vCloud API Programming Guide* includes many examples of HTTP requests and responses. These examples show the workflow and content associated with operations such as browsing, provisioning, and managing your cloud and its contents, and operating virtual systems.

Example requests generally conform to the rules listed in "Request Bodies," on page 17. Most example responses show only those elements and attributes that are relevant to the operation being discussed. Ellipses (…) indicate omitted content within response bodies. Several additional conventions apply.

- The following HTTP header, which is required in all requests that access version 1.5 of the vCloud API, is omitted from most examples.

  ```
  Accept: application/*+xml;version=1.5
  ```

- All other request headers required by the vCloud API are included in example requests that are not fragments of some larger example. Although the examples show these strings using the character case in which the implementation defines them, header names and values are case-insensitive, and can be submitted or returned in any character case. Other HTTP headers, such as `Date`, `Content-Length`, and `Server`, are omitted because they are not relevant to the specifics of any example.

- The XML version and encoding header

  ```
  <?xml version="1.0" encoding="UTF-8"?>
  ```

  is included in example requests but omitted from example responses.

- Object IDs shown in `href` attribute values appear as small integers, for example `vapp-7` or `org/3`. In the vCloud API that vCloud Director supports, object IDs are universal unique identifiers (UUIDs) as defined by RFC 4122, for example `vapp-f5e185a4-7c00-41f1-8b91-0e552d538101` or `org/89a1a8f9-c518-4f53-960c-950db9e3a1fd`.

Hello vCloud: A Simplified RESTful Workflow 2

vCloud API clients and vCloud Director servers communicate over HTTP, exchanging XML representations of vCloud API objects.

This simplified example of a RESTful workflow includes requests that discover and deploy a particular vApp, in this case, an FTP server with a connection to the public Internet.

These examples assume that you have access to a catalog that includes a vApp template with certain characteristics and an organization network that supports connections to the public Internet. The workflow and examples are flexible, and can accommodate various vApp templates and cloud capabilities.

1. Logging In on page 24

   vCloud Director requires API requests to be authenticated. The first step in any RESTful workflow is to obtain an authentication token.

2. Find a Catalog and a vDC on page 26

   Before you can deploy a vApp, you must find a vApp template in one of your organization's catalogs and a vDC in your organization to use for the deployment.

3. Retrieve the Contents of a Catalog on page 27

   You can make a GET request to a catalog URL to retrieve a list of vApp templates and media images available from the catalog.

4. Retrieve a Catalog Item on page 28

   You can examine the list of items in a catalog to find items of interest based on the values of their `name` and `type` attributes. You must retrieve a catalog item to get a `Description` and a usable reference to the underlying object.

5. Retrieve Deployment Information From the vDC on page 30

   To deploy your template as a vApp, you must specify a vDC to deploy it in and a network to connect it to.

6. Deploy the vApp on page 31

   To deploy the vApp, you construct an `InstantiateVAppTemplateParams` element that specifies a vApp template to use and a network to connect to, then POST the element to the `action/instantiateVAppTemplate` URL of the vDC.

7. Get Information About a vApp on page 34

   When you instantiate a vApp template, the server returns the URL of the resulting vApp. You can use this URL with a GET request to retrieve information that you can use to connect to the vApp, modify its configuration, and operate it.

8 Displaying the Virtual Machine Console on page 37

After a vApp is powered on, you can retrieve a screen ticket from one of its virtual machines. You use that ticket with the VMRC browser plug-in to gain access to the console of the virtual machine.

9 Delete the vApp on page 38

To delete the vApp, power it off, undeploy it, and then use an HTTP DELETE request to delete the vApp object.

10 Log Out on page 39

To log out and terminate a vCloud API session, delete the `Session` you created when you logged in.

Logging In

vCloud Director requires API requests to be authenticated. The first step in any RESTful workflow is to obtain an authentication token.

Every cloud has a login URL that a client can obtain by making an unauthenticated GET request to the vCloud Director `api/versions` URL. See "Retrieve the Login URL and List of Supported API Versions," on page 42. Because all other vCloud API requests must be authenticated, any vCloud API workflow must begin with a login request that supplies user credentials in the form that Basic HTTP authentication requires.

For information about how to create a login request and view the response, see "Example: Login Request and Response," on page 25.

Prerequisites

Verify that the following conditions are met:

- You know the username and password of the system administrator or a member of one of the organizations in the cloud. The Hello vCloud workflow requires you to log in as a user who has permission to create and operate vApps.
- You have access to an organization in which at least one vDC was created and provisioned with a network. For more information about setting up an organization to support the Hello vCloud workflow, see Chapter 6, "Creating, Provisioning, and Managing Organizations," on page 109.
- The organization contains a catalog in which at least one vApp template is available. For more information about adding a vApp template to a catalog, see Chapter 4, "Provisioning an Organization with vApps, Templates, and Media," on page 53.

Procedure

1 Make an API `versions` request to vCloud Director to obtain the login URL for the REST API.

2 Use the login URL to create a login session.

POST a request to this URL that includes your username, password, and organization name in a MIME Base64 encoding. See "Example: Login Request and Response," on page 25.

3 Examine the response.

The response code indicates whether the request succeeded, or how it failed.

A successful login request returns an authentication token that you can use in subsequent requests. It also returns a `Session` element, which contains one or more `Link` elements, each of which provides a URL that you can use to explore a subset of objects in the cloud. If you log in as a system administrator or organization administrator, this list includes multiple links. See "Example: Create a Login Session," on page 44. Otherwise, the list typically includes a link of type `application/vnd.vmware.vcloud.orgList+xml`, as shown in the response portion of "Example: Login Request and Response," on page 25. You can use this link to find out more about your organization and the objects it contains.

For more information about the other links in the `Session` element, see "Create a Login Session," on page 44.

Example: Login Request and Response

A request to create a login session must supply the user's credentials in the following form:

user@organization:*password*

- *user* is the user's login name.
- *organization* is the name of an organization of which the user is a member.
- *password* is the user's password.

These credentials must be supplied in a MIME Base64 encoding, as specified in RFC 1421.

Note System administrators must log in to the `System` organization. See "Administrator Credentials and Privileges," on page 111.

This example shows a login request and response for a user named HelloUser logging into an organization named ExampleOrg in a cloud whose login URL is https://vcloud.example.com/api/sessions.

Request:

```
POST https://vcloud.example.com/api/sessions
Authorization: Basic encoded-credentials
Accept: application/*+xml;version=1.5
```

Response:

```
200 OK
x-vcloud-authorization: cn9uYmdugN8E2j96+5Lqrc3YBvFsEgDHXzyfJrJ/6bM=
Content-Type: application/vnd.vmware.vcloud.session+xml;version=1.5
...
<Session
   xmlns="http://www.vmware.com/vcloud/v1.5"
   user="HelloUser"
   org="ExampleOrg"
   ... >
   <Link
      rel="down"
      type="application/vnd.vmware.vcloud.orgList+xml"
      href="https://vcloud.example.com/api/org"/>
  <Link
      rel="down"
      type="application/vnd.vmware.vcloud.query.queryList+xml"
      href="https://vcloud.example.com/api/query" />
   <Link
      rel="entityResolver"
      type="application/vnd.vmware.vcloud.entity+xml"
      href="https://vcloud.example.com/api/entity/" />
</Session>
```

The response code indicates whether the request succeeded, or how it failed.

- If the request is successful, the server returns HTTP response code 200 (OK) and headers that include an authorization header of the following form:

  ```
  x-vcloud-authorization: token
  ```

  This header must be included in each subsequent vCloud API request.
- If the authentication header is missing, the server returns HTTP response code 403.

- If the credentials supplied in the authentication header are invalid, or if the token has expired, the server returns HTTP response code 401. The token expires after a configurable interval of client inactivity. The default is 30 minutes after the token is created. After the token expires, you must log in again to obtain a new token.

Find a Catalog and a vDC

Before you can deploy a vApp, you must find a vApp template in one of your organization's catalogs and a vDC in your organization to use for the deployment.

After you log in, you can make a GET request to your organization's URL to retrieve the XML representation of the organization. This representation shows the organization's attributes and contents, including links to its catalogs, vDCs, and networks.

Prerequisites

Verify that you are logged in as a system administrator or member of an organization in the cloud.

Procedure

1 Examine the list of organizations to which you have access.

Make a GET request to the URL in the `href` value of the `orgList` link, which is present in the response to all login requests.

```
GET https://vcloud.example.com/api/org/
```

Unless you are a system administrator, the response to this request is an `OrgList` element containing a single `Org` element, which represents your organization.

```
<OrgList
   xmlns="http://www.vmware.com/vcloud/v1.5"
   type="application/vnd.vmware.vcloud.orgList+xml"
   href="https://vcloud.example.com/api/org">
   <Org
      type="application/vnd.vmware.vcloud.org+xml"
      name="ExampleOrg"
      href="https://vcloud.example.com/api/org/5" />
</OrgList>
```

2 Retrieve the representation of your organization.

See the request portion of "Example: Retrieve the Contents of an Organization," on page 26.

3 Examine the response to find the links to the organization's catalogs, networks, and vDCs.

See the response portion of "Example: Retrieve the Contents of an Organization," on page 26.

Example: Retrieve the Contents of an Organization

This example retrieves the `ExampleOrg` organization listed in the `OrgList` element shown in Step 1.

Request:

```
GET https://vcloud.example.com/api/org/5
```

Response:

```
200 OK
Content-Type: application/vnd.vmware.vcloud.org+xml
...
<Org
   name="ExampleOrg"
   type="application/vnd.vmware.vcloud.org+xml"
   href="https://vcloud.example.com/api/org/5">
   <Link
      rel="down"
      type="application/vnd.vmware.vcloud.catalog+xml"
      href="https://vcloud.example.com/api/catalog/32"
      name="ExampleCatalog" />
   <Link
      rel="down"
      type="application/vnd.vmware.vcloud.vdc+xml"
      href="https://vcloud.example.com/api/vdc/5"
      name="ExampleVdc01" />
   <Link
      rel="down"
      type="application/vnd.vmware.vcloud.network+xml"
      href="https://vcloud.example.com/api/network/14"
      name="IsolatedOrgNet" />
   <Link
      rel="down"
      type="application/vnd.vmware.vcloud.network+xml"
      href="https://vcloud.example.com/api/network/54"
      name="Internet" />
   <Link ... />
   <Link ... />
   <Description>Example Corp’s Primary Organization</Description>
</Org>
```

Links in the response whose `rel` attribute has a value of `down` are references to objects that the organization contains. This example shows the subset of those items that we reference in the Hello vCloud example:

- A catalog named `ExampleCatalog`, at URL https://vcloud.example.com/api/catalog/32, where you can look for vApp templates.
- A vDC named `ExampleVdc01`, at URL https://vcloud.example.com/api/vdc/5, where you can deploy the vApp.
- Two networks: one named `Internet`, at URL https://vcloud.example.com/api/network/54, and one named `IsolatedOrgNet`, at URL https://vcloud.example.com/api/network/14. You can connect the vApp to either of these networks.

Retrieve the Contents of a Catalog

You can make a GET request to a catalog URL to retrieve a list of vApp templates and media images available from the catalog.

To use a vApp template or media image listed in a catalog, retrieve the catalog to discover the set of `CatalogItem` elements it contains, then make an additional request to retrieve the `CatalogItem` of interest.

Prerequisites

Verify that you are logged in as a system administrator or member of an organization in the cloud.

Procedure

1 Retrieve the XML representation of your organization.

  Use a request like this one:

  ```
  GET https://vcloud.example.com/api/org/5
  ```

2 Examine the response to find the links to the organization's catalogs.

These links have the following form:

```
<Link
   rel="down"
   type="application/vnd.vmware.vcloud.catalog+xml"
   href="https://vcloud.example.com/api/catalog/id"
   name="catalog_name" />
```

3 Retrieve the contents of the catalog.

Use a GET request of the form shown in the request portion of "Example: Retrieving the Contents of a Catalog," on page 28.

Example: Retrieving the Contents of a Catalog

This example retrieves the catalog shown in the response portion of "Example: Retrieve the Contents of an Organization," on page 26.

Request:

```
GET https://vcloud.example.com/api/catalog/32
```

Response:

```
200 OK
Content-Type: application/vnd.vmware.vcloud.catalog+xml
...
<Catalog
   xmlns="http://www.vmware.com/vcloud/v1.5"
   name="ExampleCatalog"
   type="application/vnd.vmware.vcloud.catalog+xml"
   href="https://vcloud.example.com/api/catalog/32">
   <Description>Main Org Catalog</Description>
   <CatalogItems>
      <CatalogItem
         type="application/vnd.vmware.vcloud.catalogItem+xml"
         name="Ubuntu Template with vsftpd"
         href="https://vcloud.example.com/api/catalogItem/221"/>
       <CatalogItem ... />
       <CatalogItem ... />
   </CatalogItems>
</Catalog>
```

Retrieve a Catalog Item

You can examine the list of items in a catalog to find items of interest based on the values of their `name` and `type` attributes. You must retrieve a catalog item to get a `Description` and a usable reference to the underlying object.

Every vApp template or media image that is added to the catalog is represented as a `CatalogItem` element. When a client browses a catalog, it can read only the `name`, `type`, and `href` of each `CatalogItem`. To retrieve an item from the catalog, the client requires more information. In "Example: Retrieve a Catalog Item," on page 29, the client makes a GET request to the URL in the value of the `href` attribute of a `CatalogItem`. The response provides more information, including a description of the referenced object and another URL that the client can use to retrieve a representation of the object.

Prerequisites

Verify that you are logged in as a system administrator or member of an organization in the cloud.

Procedure

1. Retrieve the representation of a catalog in your organization.

   Use a request like this one:

   ```
   GET https://vcloud.example.com/api/catalog/32
   ```

2. Examine the response to find the `CatalogItem` elements that the catalog contains.

   The value of the `name` attribute of a `CatalogItem` element is taken from the `name` attribute of the referenced object. You can use it as a preliminary indicator of what the item represents.

3. Retrieve a `CatalogItem`.

   Use a GET request of the form shown in the request portion of "Example: Retrieve a Catalog Item," on page 29.

Example: Retrieve a Catalog Item

This example retrieves the `CatalogItem` shown in the response portion of "Example: Retrieving the Contents of a Catalog," on page 28.

Request:

```
GET https://vcloud.example.com/api/catalogItem/221
```

In addition to the `name` attribute and `Description` element, the `CatalogItem` contains a `rel="up"` link to the catalog that contains it, and other links that you can use to manage the `CatalogItem`.

Response:

```
200 OK
Content-Type: application/vnd.vmware.vcloud.catalogItem+xml
...
<CatalogItem
   xmlns="http://www.vmware.com/vcloud/v1.5"
   name="Ubuntu Template with vsftpd"
   id="urn:vcloud:catalogitem:221"
   href="https://vcloud.example.com/api/catalogItem/221" ... >
  <Link
      rel="up"
      type="application/vnd.vmware.vcloud.catalog+xml"
      href="https://vcloud.example.com/api/catalog/32" />
   <Link
      rel="down"
      type="application/vnd.vmware.vcloud.metadata+xml"
      href="https://vcloud.example.com/api/catalogItem/221/metadata" />
   <Link
      rel="edit"
      type="application/vnd.vmware.vcloud.catalogItem+xml"
      href="https://vcloud.example.com/api/catalogItem/221" />
   <Link
      rel="remove"
      href="https://vcloud.example.com/api/catalogItem/221" />
   <Description>Approved template for public FTP sites</Description>
   <Entity
```

```
      href="https://vcloud.example.com/api/vAppTemplate/vappTemplate-111"
      type="application/vnd.vmware.vcloud.vAppTemplate+xml"
      name="Ubuntu Template with vsftpd"/>
</CatalogItem>
```

Retrieve Deployment Information From the vDC

To deploy your template as a vApp, you must specify a vDC to deploy it in and a network to connect it to.

Instantiation, deployment, and operation of a vApp all take place in the context of a vDC. The XML representation of a vDC object defines that context in detail. For this exercise, you need several pieces of information from the vDC:

- The URL that a client can use to request an `instantiateVAppTemplate` operation in the vDC.
- A list of networks to which the vApp can connect.

"Example: Deployment Information in a vDC," on page 30 shows this subset of vDC contents.

Prerequisites

Verify that the following conditions are met:

- Verify that you are logged in as a system administrator or member of an organization in the cloud.
- Retrieve the representation of your organization. See the request portion of "Example: Retrieve the Contents of an Organization," on page 26. The response portion contains links to the organization's vDCs.

Procedure

1 Examine the `Org` response to find the links to the organization's vDCs.

  Links to vDCs have the form:

```
<Link
   rel="down"
   type="application/vnd.vmware.vcloud.vdc+xml"
   href="https://vcloud.example.com/api/vdc/id"
   name="vDC_name" />
```

2 Retrieve the contents of the vDC.

  Use a GET request of the form shown in the request portion of "Example: Deployment Information in a vDC," on page 30.

Example: Deployment Information in a vDC

This example shows a request to retrieve the XML representation of a vDC. It shows only the subset of the response that contains deployment information.

Request:

```
GET https://vcloud.example.com/api/vdc/5
```

Response:

```
200 OK
Content-Type: application/vnd.vmware.vcloud.vdc+xml
...
<Vdc
   xmlns="http://www.vmware.com/vcloud/v1.5"
   name="ExampleVdc01"
   type="application/vnd.vmware.vcloud.vdc+xml"
   href="https://vcloud.example.com/api/vdc/5">
   ...
   <Link
      rel="add"
      type="application/vnd.vmware.vcloud.instantiateVAppTemplateParams+xml"
      href="https://vcloud.example.com/api/vdc/5/action/instantiateVAppTemplate" />
   ...
   <AvailableNetworks>
      <Network
         href="https://vcloud.example.com/api/network/14"
         type="application/vnd.vmware.vcloud.network+xml"
         name="IsolatedOrgNet" />
      <Network
         href="https://vcloud.example.com/api/network/54"
         type="application/vnd.vmware.vcloud.network+xml"
         name="Internet" />
   </AvailableNetworks>
   ...
</Vdc>
```

The information that you need is available in the following elements of the response:

- A `Link` element that contains an the `action` URL for `instantiateVAppTemplate`. The `rel` attribute of this link has a value of `add`. It implements an action that adds an object to the vDC.
- A list of `AvailableNetworks` that includes all the networks in the organization that contains this vDC.

Deploy the vApp

To deploy the vApp, you construct an `InstantiateVAppTemplateParams` element that specifies a vApp template to use and a network to connect to, then POST the element to the `action/instantiateVAppTemplate` URL of the vDC.

To create a vApp from a vApp template, you must bind the template's abstract resource requirements, such as network connections, storage resources, memory, and CPU capacity, to appropriate resources in the target vDC. This binding operation is called instantiation.

Prerequisites

Verify that you are logged in as a system administrator or member of an organization in the cloud.

Procedure

1. Retrieve the XML representation of the vApp template.

   Make a GET request to the URL provided in the `href` attribute of the `Entity` contained by the `CatalogItem` that references the template.

2. Examine the template to find the `Vm` elements of the virtual machines that it contains.

   Look for a `NetworkConnection` element in the `Vm`. You need some of the information in that element to create a vApp network that the virtual machine can connect to.

3. Create an `InstantiateVAppTemplateParams` element.

   See "Example: Deploying a vApp," on page 32 for guidelines.

4. Make a POST request to the `action/instantiateVAppTemplate` URL of the vDC.

   Supply the `InstantiateVAppTemplateParams` element as the request body.

The server takes the requested action and returns a VApp element. The element has a status attribute value of 0, meaning it is unresolved because it is still being constructed. It also contains a Task element that tracks the progress of the request.

See the response portion of "Example: Deploying a vApp," on page 32.

Example: Deploying a vApp

This simple instantiateVAppTemplate request assumes that the vApp template includes one Vm and has no special requirements other than connecting that Vm to a network. For a look at a more complex instantiation request, see "Example: Instantiate a vApp Template," on page 79. The InstantiateVAppTemplateParams includes the following information:

- A name for the vApp, supplied in the name attribute of the InstantiateVAppTemplateParams element. This request also provides a description, which is optional but a good practice.
- A reference to the template to deploy. This reference is obtained from the href attribute of the Entity contained by the CatalogItem that you retrieved in "Retrieve a Catalog Item," on page 28 and suppled in the Source element of the InstantiateVAppTemplateParams.
- Configuration parameters for a vApp network, supplied in the NetworkConfigSection element. This specification includes the following parameters:
    - A name for the network, supplied in the name attribute of the NetworkConfigSection element. The name you specify for the vApp network must match the value of the network attribute of the NetworkConnection of the Vm. This example assumes that this NetworkConnection element includes the following values, which specify that the Vm connects to a network named VappNetwork:

      ```
      <NetworkConnectionSection
         ...
          <NetworkConnection
            network="VappNetwork">
            ...
         </NetworkConnection>
      </NetworkConnectionSection>
      ```

    - A reference to the organization network to which the vApp network connects, specified in the ParentNetwork element. The URL used in this reference is one shown in the AvailableNetworks element in "Example: Deployment Information in a vDC," on page 30.
    - A fence mode, specified in the FenceMode element. A value of bridged indicates that the vApp network is connected directly to the organization network.

  For more information about creating networks with the vCloud API, see "About vCloud Director Networks," on page 118.

The target of the request is the instantiateVAppTemplate URL of this vDC. See "Example: Deployment Information in a vDC," on page 30. Because the operation creates a new vApp object, the HTTP request type is POST.

Request:

```
POST https://vcloud.example.com/api/vdc/5/action/instantiateVAppTemplate
Content-Type: application/vnd.vmware.vcloud.instantiateVAppTemplateParams+xml
...
<?xml version="1.0" encoding="UTF-8"?>
<InstantiateVAppTemplateParams
   xmlns="http://www.vmware.com/vcloud/v1.5"
   name="Linux FTP server"
   deploy="true"
   powerOn="true"
```

```
   xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
   xmlns:ovf="http://schemas.dmtf.org/ovf/envelope/1">
   <Description>Example FTP Server</Description>
   <InstantiationParams>
      <NetworkConfigSection>
         <ovf:Info>Configuration parameters for logical networks
         </ovf:Info>
         <NetworkConfig
            networkName="vAppNetwork">
            <Configuration>
               <ParentNetwork
                  href="https://vcloud.example.com/api/network/54" />
               <FenceMode>bridged</FenceMode>
            </Configuration>
         </NetworkConfig>
      </NetworkConfigSection>
   </InstantiationParams>
   <Source
      href="https://vcloud.example.com/api/vAppTemplate/vappTemplate-111" />
</InstantiateVAppTemplateParams>
```

The response to the instantiation request is a sparsely populated vApp element that includes the following information:

- The status of the vApp. The status value 0 indicates that the vApp is unresolved, because instantiation is not complete.
- The name of the vApp, as supplied in the request.
- The vApp URL, shown in the href attribute of the VApp element. You can use this reference to retrieve information about the vApp.
- A task created to track the instantiation. The Task element has an operation attribute that describes what is happening, and contains an Owner element that is a reference the vApp being created. The vApp is the owner of the task.

Response:

```
201 Created
Content-Type: application/vnd.vmware.vcloud.vApp+xml
...
<VApp
   xmlns="http://www.vmware.com/vcloud/v1.5"
   xmlns:ovf="http://schemas.dmtf.org/ovf/envelope/1"
   deployed="false"
   status="0"
   name="Linux FTP server"
   type="application/vnd.vmware.vcloud.vApp+xml"
   href="https://vcloud.example.com/api/vApp/vapp-7">
   <Link
      rel="up"
      type="application/vnd.vmware.vcloud.vdc+xml"
      href="https://vcloud.example.com/api/vdc/5"/>
   <Description>Example FTP Server vApp</Description>
   <Tasks>
      <Task
         status="running"
         operation="Creating Virtual Application Linux FTP server(7)"
         ... >
         <Owner
            type="application/vnd.vmware.vcloud.vApp+xml"
            name="LinuxFtpServer"
            href="https://vcloud.example.com/vApp/vapp-7" />
      </Task>
   </Tasks>
</VApp>
```

Get Information About a vApp

When you instantiate a vApp template, the server returns the URL of the resulting vApp. You can use this URL with a GET request to retrieve information that you can use to connect to the vApp, modify its configuration, and operate it.

As other examples have shown, a client can always use an HTTP GET request to the URL in the object's href attribute to discover the current state of any vCloud API object, including a vApp.

Prerequisites

Verify that you are logged in as a system administrator or member of an organization in the cloud.

Procedure

1 Retrieve the XML representation of the vApp.

Make a GET request to the URL in the href attribute of the VApp element that is returned when you create the vApp from the template.

2 Examine the response.

See "Example: Getting Information About the vApp," on page 34.

Example: Getting Information About the vApp

This response reveals several things about the vApp:

- The vApp is deployed (its deployed attribute is set to true) and powered on (status="4"). See "Object Creation Status," on page 236.
- The Vm in its Children collection is also powered on and deployed. The Vm is connected to the vApp network created during instantiation. See "Example: Deploying a vApp," on page 32. Properties of this network are included in the NetworkConfigSection of the vApp, although most are not shown here. Properties of the virtual machine's connection to the network, including its IP address, are shown in the NetworkConnection of the Vm.
- Action links for all operations except powerOn are present in the VApp element and the Vm element that it contains. Because the vApp is already powered on, that operation is invalid for the vApp in its current state, so the link is not part of the response. The link for deploy is always present, even in a deployed vApp, because the deploy action is always valid. The Vm element also includes several links for actions that are not applicable to a vApp, such as acquiring a screen ticket or thumbnail, and inserting or removing media, are specific to a virtual machine. Other actions, like shutdown and reboot, can be applied to either object. See Chapter 5, "Deploying and Operating vApps," on page 75.

Request:

```
GET https://vcloud.example.com/api/vApp/vapp-7
```

Response:

```
200 OK
Content-Type: application/vnd.vmware.vcloud.vApp+xml
...
<VApp
   status="4"
   name="Linux FTP server"
   type="application/vnd.vmware.vcloud.vApp+xml"
   href="https://vcloud.example.com/api/vApp/vapp-7" ... >
   ...
   <Link
      rel="power:reboot"
      href="https://vcloud.example.com/api/vApp/vapp-7/power/action/reboot" />
   <Link
      rel="power:powerOff"
      href="https://vcloud.example.com/api/vApp/vapp-7/power/action/powerOff" />
   <Link
      rel="undeploy"
      href="https://vcloud.example.com/api/vApp/vapp-7/action/undeploy" />
   <Link
      rel="deploy"
      href="https://vcloud.example.com/api/vApp/vapp-7/action/deploy" />
   <Link
      rel="power:shutdown"
      href="https://vcloud.example.com/api/vApp/vapp-7/power/action/shutdown" />
   <Link
      rel="power:reset"
      href="https://vcloud.example.com/api/vApp/vapp-7/power/action/reset" />
   <Link
      rel="power:suspend"
      href="https://vcloud.example.com/api/vApp/vapp-7/power/action/suspend" />
   <Link
      rel="down"
      type="application/vnd.vmware.vcloud.controlAccess+xml"
      href="https://vcloud.example.com/api/vApp/vapp-7/controlAccess/" />
   <Link
      rel="controlAccess"
      type="application/vnd.vmware.vcloud.controlAccess+xml"
      href="https://vcloud.example.com/api/vApp/vapp-7/action/controlAccess/" />
   ...
   <Description>Example FTP Server vApp</Description>
   <LeaseSettingsSection ... >
      ...
   </LeaseSettingsSection>
   <ovf:StartupSection ... >
      ...
   </ovf:StartupSection>
   <ovf:NetworkSection ... >
      <ovf:Info />
      <ovf:Network
         ovf:name="vAppNetwork">
         <ovf:Description />
      </ovf:Network>
   </ovf:NetworkSection>
   <NetworkConfigSection
      href="https://vcloud.example.com/api/vApp/vapp-7/networkConfigSection/"
      ovf:required="false">
      <Link
         rel="edit"
         type="application/vnd.vmware.vcloud.networkConfigSection+xml"
         href="https://vcloud.example.com/api/vApp/vapp-7/networkConfigSection/" />
      <ovf:Info>Configuration parameters for vAppNetwork</ovf:Info>
      <NetworkConfig
         networkName="vAppNetwork">
         <Configuration>
            <IpScope>
               ...
            </IpScope>
            <ParentNetwork
               type="application/vnd.vmware.vcloud.network+xml"
               name="Internet"
               href="https://vcloud.example.com/api/network/54" />
            <FenceMode>bridged</FenceMode>
         </Configuration>
         <IsDeployed>true</IsDeployed>
      </NetworkConfig>
   </NetworkConfigSection>
   <Children>
      <Vm
         deployed="true"
         status="4"
         name="ubuntu10-x86"
         type="application/vnd.vmware.vcloud.vm+xml"
         href="https://vcloud.example.com/api/vApp/vm-4">
         ...
         <Link
            rel="power:reboot"
            href="https://vcloud.example.com/api/vApp/vm-4/power/action/reboot" />
         <Link
            rel="power:powerOff"
            href="https://vcloud.example.com/api/vApp/vm-4/power/action/powerOff" />
         <Link
            rel="undeploy"
            href="https://vcloud.example.com/api/vApp/vm-4/action/undeploy" />
         <Link
            rel="deploy"
            href="https://vcloud.example.com/api/vApp/vm-4/action/deploy" />
         <Link
            rel="power:shutdown"
            href="https://vcloud.example.com/api/vApp/vm-4/power/action/shutdown" />
         <Link
            rel="power:reset"
            href="https://vcloud.example.com/api/vApp/vm-4/power/action/reset" />
         <Link
            rel="power:suspend"
            href="https://vcloud.example.com/api/vApp/vm-4/power/action/suspend" />
         <Link
            rel="up"
            type="application/vnd.vmware.vcloud.vApp+xml"
            href="https://vcloud.example.com/api/vApp/vapp-7" />
         <Link
            rel="screen:thumbnail"
            href="https://vcloud.example.com/api/vApp/vm-4/screen" />
         <Link
            rel="screen:acquireTicket"
            href="https://vcloud.example.com/api/vApp/vm-4/screen/action/acquireTicket" />
         <Link
            rel="media:insertMedia"
            type="application/vnd.vmware.vcloud.mediaInsertOrEjectParams+xml"
            href="https://vcloud.example.com/api/vApp/vm-4/media/action/insertMedia" />
         <Link
            rel="media:ejectMedia"
            type="application/vnd.vmware.vcloud.mediaInsertOrEjectParams+xml"
            href="https://vcloud.example.com/api/vApp/vm-4/media/action/ejectMedia" />
         ...
         <Description />
         <ovf:VirtualHardwareSection>
            ...
         </ovf:VirtualHardwareSection>
         ...
         <NetworkConnectionSection>
            ...
            <NetworkConnection>
               network="vAppNetwork">
               <NetworkConnectionIndex>0</NetworkConnectionIndex>
               <IpAddress>10.147.201.10</IpAddress>
               <IsConnected>true</IsConnected>
               <MACAddress>00:50:56:01:01:49</MACAddress>
               <IpAddressAllocationMode>DHCP</IpAddressAllocationMode>
            </NetworkConnection>
         </NetworkConnectionSection>
         <GuestCustomizationSection>
            ...
         </GuestCustomizationSection>
         ...
      </Vm>
   </Children>
</VApp>
```

Displaying the Virtual Machine Console

After a vApp is powered on, you can retrieve a screen ticket from one of its virtual machines. You use that ticket with the VMRC browser plug-in to gain access to the console of the virtual machine.

A screen ticket is a string that includes the virtual machine's IP address, its managed object reference, and a residual that is encoded as described in RFC 2396. Each `Vm` element in a vApp includes a link where `rel="screen:acquireTicket"` if the virtual machine it represents is powered on. You can use that link to retrieve a screen ticket that you can use with the VMRC API to open a VMware Remote Console for the virtual machine.

NOTE You cannot access virtual machines in vCenter with the version of `vmware-vmrc` that vCloud Director installs.

Prerequisites

- Verify that you are logged in as a system administrator or member of an organization in the cloud.
- Verify that the virtual machine whose console you want to display is powered on.

- Verify that your browser has an installed copy of the `vmware-vmrc` plug-in. This plug-in is installed by your browser whenever you use the vCloud Director Web Console to access the console of a running virtual machine. After this plug-in is installed, you can find it in the folder where your browser stores plug-ins.

Procedure

1 Retrieve the screen ticket.

POST a request to the `acquireTicket` link of the `Vm`.

Request:

```
POST https://vcloud.example.com/api/vApp/vm-4/screen/action/acquireTicket
```

Response:

```
200 OK
Content-Type: application/vnd.vmware.vcloud.screenTicket+xml
...
<ScreenTicket xmlns="http://www.vmware.com/vcloud/v1.5"
   xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
   xsi:schemaLocation="http://www.vmware.com/vcloud/v1 ...>ticket-string
 </ScreenTicket>
```

The ticket string itself has the following form:

```
mks://ip-address/VM-MoRef/ticket=encoded-ticket
```

- *ip-address* is the IP address of the virtual machine.
- *VM-MoRef* is the managed object reference of the virtual machine.
- *encoded-ticket* is the encoded screen ticket. You must decode this ticket using a function such as the Java `URLDecoder` or PERL `url_escape` before you can use it.

2 Use the ticket with the VMRC API.

The ticket is valid for 30 seconds. To use it, you must initialize the VMRC browser plug-in and use the VMRC API, as described in the VMware Technical Note *Using the VMRC API*.

Delete the vApp

To delete the vApp, power it off, undeploy it, and then use an HTTP DELETE request to delete the vApp object.

A powered-on vApp has a link that you can use with a POST request to power it off. A powered-off vApp has a link that you can use with a DELETE request to remove it.

Prerequisites

Verify that you are logged in as a system administrator or member of an organization in the cloud.

Procedure

1 Retrieve the XML representation of the vApp.

Make a GET request to the URL provided in the `href` attribute of the `VApp` element returned when you created the vApp from the template. See "Get Information About a vApp," on page 34.

2 Power off the vApp.

Make a POST request to the vApp's `power/action/powerOff` link, which has the following form:

```
<Link
      rel="power:powerOff"
      href="https://vcloud.example.com/api/vApp/vapp-7/power/action/powerOff"/>
```

3 Retrieve the XML representation of the vApp again.

   a Verify that the value of its status attribute is 8, which indicates that it is powered off.

   b Verify that it includes a rel="remove" link of the following form:

```
<Link
      rel="remove"
      href="https://vcloud.example.com/api/vApp/vapp-7"/>
```

4 Remove the vApp.

   Make a DELETE request to the vApp's rel="remove" link, as shown in the request portion of "Example: Delete a vApp," on page 39.

The server starts a task to manage the events that lead up to the removal of the vApp, and returns a Task element that you can use to track the progress of the task.

Example: Delete a vApp

Request:

```
DELETE https://vcloud.example.com/api/vApp/vapp-7
```

Response:

```
202 Accepted
...
<Task
   xmlns="http://www.vmware.com/vcloud/v1.5"
   status="running"
   startTime="2010-06-25T08:10:23.650-07:00"
   operation="Deleting Virtual Application Linux FTP server (7)"
   expiryTime="2010-09-23T08:00:55.402-07:00"
   type="application/vnd.vmware.vcloud.task+xml"
   href="https://vcloud.example.com/api/task/3478" ... >
</Task>
```

Log Out

To log out and terminate a vCloud API session, delete the Session you created when you logged in.

The logout request, like all other authenticated requests, must include the authorization header, as shown in "Example: Logging Out," on page 39.

Prerequisites

Verify that you are logged in.

Procedure

- Make a DELETE request specifying the href of the current Session object.

Example: Logging Out

This example deletes the current user's Session, which logs the user out.

Request:

```
DELETE https://vcloud.example.com/api/session
x-vcloud-authorization: cn9uYmdugN8E2j96+5Lqrc3YBvFsEgDHXzyfJrJ/6bM=
```

Response:

```
200 OK
```

Exploring a Cloud 3

You can use HTTP GET requests in a process of serial discovery to browse containers such as organizations, catalogs, and vDCs in a cloud.

Responses to these requests include metadata about the container itself and references to the objects it contains. These references are provided in `Link` elements, which have `href` attributes whose values the client can use in requests to get more information about the objects themselves. This process is sometimes called serial discovery, where the contents of one response provides links to locations where you can look for more information. The hierarchical structure of vCloud API container objects lends itself to graphical representation as a folder hierarchy or tree view of vCloud API objects, and enables clients to use the same set of objects and operations to implement a breadth-first or depth-first approach to browsing.

The list of entry points from which you can begin browsing is contained in the `Session` element that is returned in response to a successful login. This list is based on your role and privileges.

This chapter includes the following topics:

Summary of vCloud API Browsing Requests

Browsing requests provide read-only access to a cloud and the objects it contains.

- *API-URL* is a URL of the form `https://vcloud.example.com/api`.
- *id* is a unique identifier in the form of a UUID, as defined by RFC 4122.

Request URLs are always available when you GET the representation of the object on which they operate. URL forms are for reference purposes only. For more information about the requests and responses for each operation, see "About the Schema Reference Downloadable Archive," on page 19.

Table 3-1. Summary of vCloud API Browsing Requests

Operation	Request	Request Body	Response
Show login URL and list of supported API versions	GET *API-URL*/versions	None	`SupportedVersions`
Log in [NEW]	POST *API-URL*/sessions	None	`Session`

Table 3-1. Summary of vCloud API Browsing Requests (Continued)

Operation	Request	Request Body	Response
Log out [NEW]	DELETE *API-URL*/session	None	200 OK
Log in [DEPRECATED]	POST *API-URL*/login	None	`OrgLIst`
Log out [DEPRECATED]	POST *API-URL*/login	None	200 OK
Retrieve a list of entry points for browsing operations [NEW]	GET *API-URL*/session	None	200 OK
Retrieve a list of organizations to which you have access	GET *API-URL*/org/	None	`OrgList`
Retrieve the contents of an organization	GET *API-URL*/org/*id*	None	`Org`
Retrieve properties of a network	GET *API-URL*/network/*id*	None	`OrgNetwork`
Retrieve the contents of a catalog	GET *API-URL*/catalog/*id*	None	`Catalog`
Retrieve properties of a catalog item	GET *API-URL*/catalogItem/*id*	None	`CatalogItem`
Retrieve the contents of a vDC	GET *API-URL*/vdc/*id*	None	`Vdc`
Retrieve properties of a media image	GET *API-URL*/media/*id*	None	`Media`
Retrieve a vApp template	GET *API-URL*/vAppTemplate/vappTemplate-*id*	None	`VAppTemplate`
Retrieve properties of a vApp	GET *API-URL*/vApp/vapp-*id*	None	`VApp`
Retrieve properties of a virtual machine	GET *API-URL*/vApp/vm-*id*	None	`Vm`

Retrieve the Login URL and List of Supported API Versions

Every cloud has a login URL that a client can obtain by making an unauthenticated GET request to the vCloud Director `api/versions` URL. The response to this request also lists vCloud API versions that the server supports.

Procedure

1 Make an API version request to vCloud Director to obtain the login URL for the REST API.

See the request portion of "Example: Versions Request and Response," on page 43.

2 Examine the response to find the login URLs.

Each version of the vCloud API that the server supports has its own login URL. Look for a line of the following form:

```
<LoginUrl>http://vcloud.example.com/api/...</LoginUrl>
```

3 Use the login URL to log in to the cloud.

Example: Versions Request and Response

The `api/versions` request does not need to be authenticated. The response, a small subset of which is shown here, includes a `VersionInfo` element for each API version that the server supports. Each `VersionInfo` element contains:

- A `LoginUrl` element that contains the URL to which a client can make a login request to access that version of the vCloud API. See "Logging In," on page 24.
- `MediaTypeMapping` elements for each complex type supported by that version of the vCloud API.

Request:

```
GET http://vcloud.example.com/api/versions
```

Response:

```
200 OK
Content-Type: text/xml
...
<SupportedVersions
   xmlns="http://www.vmware.com/vcloud/versions"
   xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
   xsi:schemaLocation="http://www.vmware.com/vcloud/versions
   http://vcloud.example.com/api/versions/schema/versions.xsd">
   <VersionInfo>
      <Version>1.0</Version>
      <LoginUrl>https://vcloud.example.com/api/login</LoginUrl>
      <MediaTypeMapping>
         <MediaType>application/vnd.vmware.vcloud.catalog+xml</MediaType>
         <ComplexTypeName>CatalogType</ComplexTypeName>
         <SchemaLocation>http://vcloud.example.com/api/v1.0/schema/master.xsd</SchemaLocation>
      </MediaTypeMapping>
      <MediaTypeMapping>
         ...
      </MediaTypeMapping>
  </VersionInfo>
  <VersionInfo>
      <Version>1.5</Version>
      <LoginUrl>https://vcloud.example.com/api/sessions</LoginUrl>
     <MediaTypeMapping>
         <MediaType>application/vnd.vmware.vcloud.catalog+xml</MediaType>
         <ComplexTypeName>CatalogType</ComplexTypeName>
         <SchemaLocation>http://vcloud.example.com/api/v1.5/schema/master.xsd</SchemaLocation>
      </MediaTypeMapping>
      <MediaTypeMapping>
         ...
      </MediaTypeMapping>
           ...
  </VersionInfo>
</SupportedVersions>
```

Note You can use the URL in the `SchemaLocation` element with a GET request to retreive the file in which that complex type is defined. For example:

```
GET http://vcloud.example.com/api/v1.5/schema/master.xsd
```

Create a Login Session

The vCloud API login mechanism authenticates a user and creates a `Session` object that contains the URLs from which that user can begin browsing.

Prerequisites

- Verify that you know the login URL. See "Retrieve the Login URL and List of Supported API Versions," on page 42
- Verify that you know the username and password of the system administrator or a member of one of the organizations in the cloud.

System administrators must log in to the `System` organization. See "Administrator Credentials and Privileges," on page 111.

Procedure

1 Use the login URL to authenticate to the cloud.

POST a request to this URL. The request must include your username, organization name, and password in a MIME Base64 encoding. See "Example: Create a Login Session," on page 44.

2 Examine the response.

The response code indicates whether the request succeeded, or how it failed.

- If the authentication header is missing, the server returns HTTP response code 403.
- If the credentials supplied in the authentication header are invalid, the server returns HTTP response code 401.
- If the request is successful, the server returns HTTP response code 200 (OK) and headers that include an authorization header of the form:

```
x-vcloud-authorization: token
```

This header must be included in each subsequent vCloud API request.

The `Session` element returned from a successful login contains one or more URLs from which you can begin browsing.

The list of URLs in the `Session` object is based on the role and privileges of the authenticated user. A `Session` object expires after a configurable interval of client inactivity. To change the length of this client inactivity timeout, a system administrator can change the value of `SessionTimeoutMinutes` in the system's `GeneralSettings`. See "Retrieve or Update System Settings," on page 165.

A `Session` object can be deleted by its owner or an administrator. After your `Session` expires or is deleted, you are not authenticated.

Example: Create a Login Session

A request to create a login session must supply the user's credentials in the following form:

`user@organization``:`*`password`*

- *user* is the user's login name.
- *organization* is the name of an organization of which the user is a member.
- *password* is the user's password.

These credentials must be supplied in a MIME Base64 encoding, as specified in RFC 1421.

This example shows a login request and response for a system administrator logging in to a cloud whose `sessions` URL is `https://vcloud.example.com/api/sessions`.

Request:

```
POST https://vcloud.example.com/api/sessions
Authorization: Basic encoded-credentials
Accept: application/*+xml;version=1.5
```

Response:

```
200 OK
x-vcloud-authorization: cn9uYmdugN8E2j96+5Lqrc3YBvFsEgDHXzyfJrJ/6bM=
Content-Type: application/vnd.vmware.vcloud.session+xml
...
<Session
   xmlns="http://www.vmware.com/vcloud/v1.5"
   user="sysadmin"
   org="System"
   ... >
   <Link
      rel="down"
      type="application/vnd.vmware.vcloud.orgList+xml"
      href="https://vcloud.example.com/api/org"/>
   <Link
      rel="down"
      type="application/vnd.vmware.admin.vcloud+xml"
      href="https://vcloud.example.com/api/admin"/>
   <Link
      rel="down"
      type="application/vnd.vmware.admin.vmwExtension+xml"
      href="https://vcloud.example.com/api/admin/extension"/>
  <Link
      rel="down"
      type="application/vnd.vmware.vcloud.query.queryList+xml"
      href="https://vcloud.example.com/api/query" />
   <Link
      rel="entityResolver"
      type="application/vnd.vmware.vcloud.entity+xml"
      href="https://vcloud.example.com/api/entity/" />
</Session>
```

This response includes the following link types:

orgList
A link to the list of organizations that you can access. See "Retrieve a List of Organizations Accessible to You," on page 46.

vcloud
A link to administrative objects and operations. See Chapter 6, "Creating, Provisioning, and Managing Organizations," on page 109

vmwExtension
A link to the vCloud API extensions, accessible to a system administrator. See Chapter 7, "Using vCloud API Extensions to Provision and Manage a Cloud," on page 161.

queryList
A link to the set of typed queries you can run. See Chapter 9, "Using the Query Service," on page 201.

entity
A link to the entity resolver. See "Retrieve an Object as an Entity," on page 237.

Retrieve a List of Organizations Accessible to You

A successful login request returns a Session element, which contains a link to a list of all organizations that you are permitted to access.

Every authenticated user has an associated Session object that contains one or more Link elements. The set of Link elements in your Session is based on your role and privileges. Each of these elements includes a URL that you can use with a GET request to explore a subset of objects in the cloud.

All Session elements include a link that you can use to retrieve an OrgList element. For an ordinary user, this list includes just the organization to which the user logged in. For an organization administrator or system administrator, the list includes all organizations in the cloud.

Prerequisites

Create a login session. See "Create a Login Session," on page 44.

Procedure

1 Retrieve the XML representation of your Session object.

Use a request like this one:

```
GET https://vcloud.example.com/api/session
```

2 Examine the contents of the Session element to locate the link to the organization list.

This link has the following form:

```
<Link
   rel="down"
   type="application/vnd.vmware.vcloud.orgList+xml"
   href="https://vcloud.example.com/api/org"/>
```

3 Retrieve the list of organizations by making a GET request to the href value of the Link.

See "Example: Retrieve an Organization List," on page 46.

Example: Retrieve an Organization List

Request:

```
GET https://vcloud.example.com/api/org
```

The request returns an OrgList element similar to the one shown here. Additional Org elements are returned only when a system administrator makes the request.

Response:

```
200 OK
Content-Type: application/vnd.vmware.vcloud.orgList+xml
...
<OrgList
   xmlns="http://www.vmware.com/vcloud/v1.5"
   type="application/vnd.vmware.vcloud.orgList+xml"
   href="https://vcloud.example.com/api/org">
   <Org
      type="application/vnd.vmware.vcloud.org+xml"
      name="ExampleOrg"
      href="https://vcloud.example.com/api/org/5" />
   <Org ... />
   <Org ... />
</OrgList>
```

Retrieve an Administrative View of a Cloud

A successful login by an organization or system administrator returns a Session element that contains a link that you can use to retrieve a VCloud element. The VCloud element provides access to a cloud-wide namespace of objects that an organization administrator can view and, in most cases, modify.

The primary administrative objects in a cloud include organizations, provider vDCs, rights, roles, and external networks. Each object type is represented in a VCloud element by zero or more references. The vCloud API defines several objects that are used only in administrative operations. Some, like User, Group, and Role, are unique to administrative operations. Others extend common vCloud API objects to add elements and attributes that only administrators can view or modify. An AdminOrg, for example, provides an administrative view of an Org, and an AdminVdc does the same thing for a Vdc.

A system administrator can obtain more information about any of these objects by making a GET request to its URL, which is the value of its href attribute.

The vCloud element includes links that enable a system administrator to add organizations and roles. Subordinate objects such as users, catalogs, and vDCs are contained by individual organizations and not listed at this level.

Prerequisites

Use the credentials of an organization administrator or system administrator to create a login session. See "Create a Login Session," on page 44.

Procedure

1 Retrieve the XML representation of your Session object.

Use a request like this one:

```
GET https://vcloud.example.com/api/session
```

2 Examine the contents of the Session element to locate the link to the VCloud object.

This link has the following form:

```
<Link
   rel="down"
   type="application/vnd.vmware.admin.vcloud+xml"
   href="https://vcloud.example.com/api/admin"/>
```

3 Retrieve the VCloud element by making a GET request to the href value of the Link described in Step 2.

See "Example: Retrieve an Administrative View of a Cloud," on page 47

Example: Retrieve an Administrative View of a Cloud

Request:

```
GET https://vcloud.example.com/api/admin
```

Response:

```
200 OK
Content-Type: application/vnd.vmware.admin.vcloud+xml
...
<VCloud
```

```
xmlns="http://www.vmware.com/vcloud/v1.5"
name="vCloud"
href="https://vcloud.example.com/api/admin">
<Link
   rel="add"
   type="application/vnd.vmware.admin.role+xml"
   href="https://vcloud.example.com/api/admin/roles" />
<Link
   rel="add"
   type="application/vnd.vmware.admin.organization+xml"
   href="https://vcloud.example.com/api/admin/orgs" />
<Link
   rel="down"
   type="application/vnd.vmware.admin.systemOrganization+xml"
   name="System"
   href="https://vcloud.example.com/api/admin/org/99" />
<Description>Example Corporation’s vCloud</Description>
<OrganizationReferences>
  <OrganizationReference
      type="application/vnd.vmware.admin.organization+xml"
      name="Engineering"
      href="https://vcloud.example.com/api/admin/org/1"/>
   <OrganizationReference
      type="application/vnd.vmware.admin.organization+xml"
      name="Engineering"
      href="https://vcloud.example.com/api/admin/org/44"/>
   <OrganizationReference ... />
   ...
</OrganizationReferences>
<ProviderVdcReferences>
   <ProviderVdcReference
      type="application/vnd.vmware.admin.providervdc+xml"
      name="Main Provider"
      href="https://vcloud.example.com/api/admin/providervdc/2" />
   <ProviderVdcReference ... />
   ...
</ProviderVdcReferences>
<RightReferences>
   <RightReference
      type="application/vnd.vmware.admin.right+xml
      name="vApp_Deploy"
      href="https://vcloud.example.com/api/admin/right/3" />
   <RightReference
      type="application/vnd.vmware.admin.right+xml
      name="Catalog:Sharing"
      href="https://vcloud.example.com/api/admin/right/7" />
   <RightReference ... />
   ...
</RightReferences>
<RoleReferences>
   <RoleReference
      type="application/vnd.vmware.admin.role+xml"
      name="Organization Administrator"
      href="https://vcloud.example.com/api/admin/role/102" />
   <RoleReference
```

```
         type="application/vnd.vmware.admin.role+xml"
         name="Catalog Creator"
         href="https://vcloud.example.com/api/admin/role/103" />
      <RoleReference />
   </RoleReferences>
   <Networks>
      <Network
         type="application/vnd.vmware.admin.network+xml"
         name="ExternalNetwork-VC1"
         href="https://vcloud.example.com/api/admin/network/7" />
      <Network
         type="application/vnd.vmware.admin.network+xml"
         name="ExternalNetwork-VC2"
         href="https://vcloud.example.com/api/admin/network/33" />
    </Networks>
</VCloud>
```

Retrieve a List of vSphere Platform Operations and Objects for a Cloud

A successful login by a system administrator returns a `Session` element that contains a link that you can use to retrieve a `VMWExtension` element.

Every vCloud Director installation depends on vSphere platform resources such as vCenter, ESX/ESXi, and vShield Manager. The `VMWExtension` element provides access to a cloud-wide namespace of vSphere platform objects that are registered for use by the cloud, and links that allow you to add vSphere servers and related resources such as networks and resource pools to your cloud. The administrative extensions to the vCloud API provide a system administrator with programmatic access to these resources.

Prerequisites

Use the credentials of a system administrator to create a login session. See "Create a Login Session," on page 44.

Procedure

1. Retrieve the XML representation of your `Session` object.

   Use a request like this one:

   ```
   GET https://vcloud.example.com/api/session
   ```

2. Examine the contents of the `Session` element to locate the link to the `VMWExtension` object.

   This link has the following form:

   ```
   <Link
      rel="down"
      type="application/vnd.vmware.admin.vmwExtension+xml"
      href="https://vcloud.example.com/api/admin/extension"/>
   ```

3. Retrieve the list of organizations by making a GET request to the `href` value of the `Link` described in Step 2.

   The request returns a `VMWExtension` element, as shown in "Example: Retrieve a List of vSphere Platform Operations and Objects for a Cloud," on page 49.

Example: Retrieve a List of vSphere Platform Operations and Objects for a Cloud

Request:

```
GET https://vcloud.example.com/api/admin/extension
```

Response:

```
200 OK
Content-Type: application/vnd.vmware.admin.vmwextension+xml
...
<vmext:VMWExtension
   xmlns:vmext="http://www.vmware.com/vcloud/extension/v1.5"
   xmlns:vcloud="http://www.vmware.com/vcloud/v1.5"
   type="application/vnd.vmware.admin.vmwExtension+xml">
   <vcloud:Link
      rel="down"
      type="application/vnd.vmware.admin.vmwProviderVdcReferences+xml"
      href="https://vcloud.example.com/api/admin/extension/providerVdcReferences" />
   <vcloud:Link
      rel="down"
      type="application/vnd.vmware.admin.vmwExternalNetworkReferences+xml"
      href="https://vcloud.example.com/api/admin/extension/externalNetworkReferences" />
   <vcloud:Link
      rel="down"
      type="application/vnd.vmware.admin.vmwNetworkPoolReferences+xml"
      href="https://vcloud.example.com/api/admin/extension/networkPoolReferences" />
   <vcloud:Link
      rel="down"
      type="application/vnd.vmware.admin.vmwVimServerReferences+xml"
      href="https://vcloud.example.com/api/admin/extension/vimServerReferences" />
   <vcloud:Link
      rel="down"
      type="application/vnd.vmware.admin.vmwHostReferences+xml"
      href="https://vcloud.example.com/api/admin/extension/hostReferences" />
   <vcloud:Link
      rel="down"
      type="application/vnd.vmware.admin.systemSettings+xml"
      href="https://vcloud.example.com/api/admin/extension/settings" />
   <vcloud:Link
      rel="add"
      type="application/vnd.vmware.admin.vmwprovidervdc+xml"
      href="https://vcloud.example.com/api/admin/extension/providervdcs" />
   <vcloud:Link
      rel="add"
      type="application/vnd.vmware.admin.vmwexternalnet+xml"
      href="https://vcloud.example.com/api/admin/extension/externalnets" />
   <vcloud:Link
      rel="add"
      type="application/vnd.vmware.admin.networkPool+xml"
      href="https://vcloud.example.com/api/admin/extension/networkPools" />
   <vcloud:Link
      rel="add"
      type="application/vnd.vmware.admin.registerVimServerParams+xml"
      href="https://vcloud.example.com/api/admin/extension/action/registervimserver" />
   <vcloud:Link
      rel="down"
      type="application/vnd.vmware.admin.licensingReportList+xml"
      href="https://vcloud.example.com/api/admin/extension/licensing/reports" />
   <vcloud:Link
      rel="down"
      type="application/vnd.vmware.admin.datastoreList+xml"
      href="https://vcloud.example.com/api/admin/extension/datastores" />
   <vcloud:Link
      rel="blockingTask"
      type="application/vnd.vmware.admin.blockingTaskList+xml"
      href="https://vcloud.example.com/api/admin/extension/blockingTasks/" />
</vmext:VMWExtension>
```

Provisioning an Organization with vApps, Templates, and Media 4

The vCloud API provides several ways for you to make vApp templates, vApps, and media images available to users in an organization.

With the vCloud API, you can upload and download OVF packages, and upload media images. Transfer operations are characterized as uploads when the operation transfers content from the local host to a remote host, and as downloads when the local host requests the transfer of content from a remote host. A POST request initiates uploads, and a GET request initiates downloads. The vCloud Director transfer service facilitates uploads and downloads and provides temporary storage for files. After they are uploaded, you can add templates and media images to catalogs as needed.

In addition to uploading, you can use the following operations to provision an organization with vApp templates, vApps, and media images:

Cloning The vCloud API clone operation copies a vApp, vApp template, or media image. You can specify whether to delete the source object after the operation completes. Deleting the source object after cloning it moves or renames it. After they are cloned, you can add templates and media images to catalogs as needed.

Capturing The vCloud API capture operation captures a vApp to create a vApp template. You can add the captured template to a catalog, download it, or both.

Importing A system administrator can import a virtual machine from a vCenter server that is registered to the cloud. You can import the virtual machine as a vApp or as a vApp template. You can add an imported template to a catalog, download it, or both.

This chapter includes the following topics:

Summary of vCloud API Provisioning Requests

Provisioning requests add vApp templates and media to a vDC and a catalog. You can also use provisioning requests to copy, move, rename, and delete these objects.

- *API-URL* is a URL of the form `https://vcloud.example.com/api`.
- *id* is a unique identifier in the form of a UUID, as defined by RFC 4122.

Request URLs are always available when you GET the representation of the object on which they operate. URL forms are for reference purposes only. For more information about the requests and responses for each operation, see "About the Schema Reference Downloadable Archive," on page 19.

Table 4-1. Summary of Provisioning Requests

Operation	Request	Request Body	Response
Upload OVF to create a vApp template.	POST *API-URL*/vdc/*id*/action/uploadVAppTemplate	`UploadVAppTemplateParams`	`VAppTemplate`
Download a vApp template as OVF.	GET *download-URL*	None	Depends on file type
Enable a vApp template for download.	POST *API-URL*/vAppTemplate/vAppTemplate-*id*/action/enableDownload	None	`Task`
Disable a vApp template for download.	POST *API-URL*/vAppTemplate/vAppTemplate-*id*/action/disableDownload	None	204 No Content
Upload a media image.	POST *API-URL*/vdc/*id*/media	`Media`	`Media`
Copy or move a media image.	POST *API-URL*/vdc/*id*/action/cloneMedia	`CloneMediaParams`	`Media`
Copy or move a vApp template.	POST *API-URL*/vdc/*id*/action/cloneVAppTemplate	`CloneVAppTemplateParams`	`VAppTemplate`
Copy or move a vApp.	POST *API-URL*/vdc/*id*/action/cloneVApp	`CloneVAppParams`	`VApp`
Change the name or description of a vApp template.	PUT *API-URL*/vAppTemplate/vappTemplate-*id*	`VAppTemplate`	`Task`
Change the name or description of a vApp.	PUT *API-URL*/vApp/vapp-*id*	`VApp`	`Task`
Change the name or description of a media image.	PUT *API-URL*/media/*id*	`Media`	`Task`
Delete a vApp template, vApp, or media image.	DELETE *object-URL*	None	`Task`
Add an item to a catalog.	POST *API-URL*/catalog/*id*/catalogItems	`CatalogItem`	`CatalogItem`
Remove an item from a catalog.	DELETE *API-URL*/catalog/*id*/catalogItem/id	None	204 No content

Table 4-1. Summary of Provisioning Requests (Continued)

Operation	Request	Request Body	Response
Control access to catalogs.	POST *API-URL*/catalog/*id*/action/controlAccess	`ControlAccessParams`	`ControlAccessParams`
Retrieve the owner of a media object. [NEW]	GET *API-URL*/media/*id*/owner	None	`Owner`
Retrieve the owner of a vApp template [NEW]	GET *API-URL*/vAppTemplate/vappTemplate-*id*/owner	None	`Owner`
Retrieve the owner of a vApp [NEW]	GET *API-URL*/vApp/*id*/owner	None	`Owner`
Update the owner of a vApp [NEW]	PUT *API-URL*/vApp/*id*/owner	`Owner`	204 No Content

Upload an OVF Package to Create a vApp Template

A vCloud API client that has access to an OVF package can use a standard workflow to upload the package and create a vApp template.

The initial configuration of a vApp is established in the OVF package on which its source template is based. In the vCloud API, vApp templates are based OVF 1.0, an open standard format. For more information about OVF and how the vCloud API uses it, see "About OVF," on page 75.

An OVF package includes several kinds of files.

An OVF descriptor
An XML file that contains metadata that describe a virtual machine or collection of related virtual machines and the deployment environment they require. By convention, this file has the suffix `.ovf`.

Virtual disk files
The descriptor lists these files and includes information about their format.

An optional certificate
You can use this file to certify the authenticity of the package.

An optional manifest
Contains a SHA-1 digest of each of the files in the package.

Upload Workflow

The upload workflow for OVF packages uses a combination of vCloud API requests and standard HTTP file transfer requests.

1 The client uses a POST request that specifies a name and description for the template, and a transfer format for the data.

2 The server returns an unresolved `VAppTemplate` element with (`status="0"`) that includes an upload URL for the OVF descriptor.

3 The client uses an HTTP PUT request to upload the descriptor to the upload URL.

4 The server reads the descriptor and modifies the `vAppTemplate` to include an upload URL for each file listed in the `References` section of the descriptor. While the server is modifying the `vAppTemplate`, the client makes periodic requests for it and examines the response for additional upload URLs. When the response contains additional upload URLs that were not present in the initial response, template construction is complete.

5 The client uses HTTP PUT requests to upload each of the files.

6 If the OVF package includes a manifest file, the entire upload is validated against the contents of the manifest file.

Both monolithic and ranged, or chunked, PUT requests are supported. After starting an upload, a client can make periodic requests to assess its progress. After all of the files are uploaded, and validated if a manifest is present, the server processes them and updates the vApp template. When processing is complete, the server sets the value of the template's `status` attribute to `8`, indicating that it is ready for use. This status value indicates that all of the virtual machines in the template are powered off. For more information, including a complete list of possible status values and their meanings, see "Object Creation Status," on page 236.

Restrictions on Uploaded Content

The vCloud Director transfer service imposes the following restrictions on uploaded OVF content:

- You can upload either OVF 1.0 or OVF 1.1 content. OVF 1.1 packages are converted to OVF 1.0 for download, and any OVF 1.1 content is lost.
- You cannot upload a compressed OVF package.
- If you upload an OVF package in which any `VirtualSystem` element has an `ovf:id` attribute value that is longer than 13 characters, the name of the `Vm` that represents that `VirtualSystem` in the `vAppTemplate` that the upload creates is rewritten as the first 13 characters of the `ovf:id` attribute followed by three digits. For example, `NewVirtualMachine1` and `NewVirtualMachine2` become `NewVirtualMa001` and `NewVirtualMa002`.

1. Initiating the OVF Upload on page 56

   To initiate the OVF upload, a client makes a POST request to the `uploadVAppTemplate` URL of the target vDC. The request body is an `UploadVAppTemplateParams` element.

2. Uploading the OVF Descriptor on page 59

   You upload the OVF descriptor by making a PUT request to an upload URL and supplying the descriptor's contents as an `Envelope` element in the request body. If the request is valid, the server responds with a `200 OK` status.

3. Retrieving the Upload URLs on page 59

   After an OVF descriptor is uploaded, the server validates it and, if it is valid, updates the corresponding template with upload URLs for each of the files referenced in the descriptor. You must retrieve the template to see these URLs.

4. Uploading Referenced Files on page 61

   You can use a PUT request to upload each file that the vApp template references.

Initiating the OVF Upload

To initiate the OVF upload, a client makes a POST request to the `uploadVAppTemplate` URL of the target vDC. The request body is an `UploadVAppTemplateParams` element.

The first step in uploading an OVF package is to request vCloud Director to create a `vAppTemplate` object to represent the template that will be constructed from the upload. This request returns a response that includes a URL to which you can upload the package's descriptor file.

Prerequisites

Verify that you have the following information:

- An OVF package to upload.
- You are logged in as a user who has permission to upload OVF packages and create vApp templates.
- You know the URL of the target vDC that will receive the upload. Retrieve the XML representation of your organization to see a list of the vDCs that it contains.

Procedure

1 Find the uploadVappTemplate link in the target vDC.

Retrieve the XML representation of the vDC using a request like the one shown in the request portion of "Example: Deployment Information in a vDC," on page 30. The response contains an uploadVappTemplate link, which has the following form:

```
<Link
      rel="add"
      type="application/vnd.vmware.vcloud.uploadVAppTemplateParams+xml"
      href="https://vcloud.example.com/api/vdc/5/action/uploadVAppTemplate" />
```

2 Create an UploadVAppTemplateParams element that specifies a name for the template.

The Description element is optional. Using it to provide a concise description of this object is a best practice. See the request portion of "Example: Initiating the Upload," on page 57.

3 (Optional) If the OVF package includes a manifest, include a manifestRequired="true" attribute in the UploadVAppTemplateParams element.

Some OVF packages include a manifest document, which provides a checksum for each file in the package. When the UploadVAppTemplateParams element includes a manifestRequired="true" attribute, the set of File elements returned after you upload the OVF descriptor includes one for the manifest itself.

4 Make an HTTP POST request to the uploadVAppTemplate link that you retrieved in Step 1, supplying the UploadVAppTemplateParams element in the request body.

See the request portion of "Example: Initiating the Upload," on page 57.

5 Examine the response.

The response, a VAppTemplate element, contains a File element that specifies an upload URL for the package's OVF descriptor. See the response portion of "Example: Initiating the Upload," on page 57.

The server creates a new VAppTemplate object in the target vDC and returns the object's XML representation in the response. See the response portion of "Example: Initiating the Upload," on page 57.

Example: Initiating the Upload

This example assumes an OVF package that has no manifest.

Request:

```
POST https://vcloud.example.com/api/vdc/5/action/uploadVAppTemplate
Content-Type: application/vnd.vmware.vcloud.uploadVAppTemplateParams+xml
...
<?xml version="1.0" encoding="UTF-8"?>
<UploadVAppTemplateParams
   name="Ubuntu Template"
   xmlns="http://www.vmware.com/vcloud/v1.5"
   xmlns:ovf="http://schemas.dmtf.org/ovf/envelope/1">
   <Description>Ubuntu vApp Template</Description>
</UploadVAppTemplateParams>
```

Response:

```
201 Created
Content-Type: application/vnd.vmware.vcloud.vAppTemplate+xml
...
<VAppTemplate
   xmlns="http://www.vmware.com/vcloud/v1.5"
   xmlns:ovf="http://schemas.dmtf.org/ovf/envelope/1"
   ovfDescriptorUploaded="true"
   goldMaster="false"
   status="0"
   name="Ubuntu Template"
   id="urn:vcloud:vapptemplate:268"
   href="https://vcloud.example.com/api/vAppTemplate/vappTemplate-268"
   type="application/vnd.vmware.vcloud.vAppTemplate+xml">
   <Link
      rel="up"
      type="application/vnd.vmware.vcloud.vdc+xml"
      href="https://vcloud.example.com/api/vdc/5"/>
   <Link
      rel="remove"
      href="https://vcloud.example.com/api/vAppTemplate/vappTemplate-268" />
   <Description>Ubuntu vApp Template</Description>
   <Files>
      <File
         name="descriptor.ovf"
         bytesTransferred="0">
         <Link
            rel="upload:default"
            href="https://vcloud.example.com/transfer/.../descriptor.ovf" />
      </File>
   </Files>
   <Owner>
      ...
   </Owner>
   <Children />
   <LeaseSettingsSection>
      ...
   </LeaseSettingsSection>
   <CustomizationSection>
      ...
   </CustomizationSection>
</VAppTemplate>
```

The response body includes the following attributes:

- An href attribute whose value is a link to the new VAppTemplate object.
- An ovfDescriptorUploaded attribute with a value of false, indicating that the OVF descriptor file is not uploaded
- A status attribute with a value of 0, indicating that the file references in the descriptor are not uploaded. (A VAppTemplate with a status of 0 is said to be unresolved.)
- A goldMaster attribute, initially set to false
- An id attribute. See “Objects, References, and Representations,” on page 11.

The response body also includes a File element with an upload URL (rel="upload:default") for the OVF descriptor. The server creates the name attribute of this File element, which specifies a container that the server creates to receive the contents of the descriptor. The name attribute has no relation to the file name of the descriptor in the client’s file system.

In addition to the File element, the response includes Owner, Children, LeaseSettingsSection, and CustomizationSection elements that the server creates and sets to their default contents. For more information about these elements, see the schema reference.

Uploading the OVF Descriptor

You upload the OVF descriptor by making a PUT request to an upload URL and supplying the descriptor's contents as an `Envelope` element in the request body. If the request is valid, the server responds with a `200 OK` status.

Prerequisites

Verify that you initiated the upload with an `uploadVAppTemplate` request and received an upload URL for the descriptor. See "Initiating the OVF Upload," on page 56.

Procedure

1 Upload the OVF descriptor.

Make a PUT request to the upload URL returned in the response to the `uploadVAppTemplate` request, and supply the OVF descriptor as the request body. The OVF descriptor contains a single `Envelope` element.

2 Verify that the request succeeded.

A response of the following form indicates that the request was valid and is being processed:

```
200 OK
```

Example: Uploading the OVF Descriptor

Request:

```
PUT https://vcloud.example.com/transfer/.../descriptor.ovf
Content-Type text/xml
...
<?xml version="1.0" encoding="UTF-8"?>
<Envelope
   xmlns="http://schemas.dmtf.org/ovf/envelope/1"
   ... >
   ...
</Envelope>
```

Response:

```
200 OK
```

Retrieving the Upload URLs

After an OVF descriptor is uploaded, the server validates it and, if it is valid, updates the corresponding template with upload URLs for each of the files referenced in the descriptor. You must retrieve the template to see these URLs.

Procedure

1 Retrieve the `VAppTemplate` to verify that the OVF descriptor is uploaded.

Use the URL returned in the response to your `uploadVAppTemplate` request. See the request portion of "Example: Upload URLs in a vAppTemplate," on page 60.

2 Verify that the value of the template's `ovfDescriptorUploaded` attribute is `true`.

3 Examine the template to find the upload URLs for the files referenced in the OVF descriptor.

These URLs are contained in `Link` elements where `rel="upload:default"`.

Example: Upload URLs in a vAppTemplate

This request uses the vApp template URL returned in "Example: Initiating the Upload," on page 57.

Request:

```
GET https://vcloud.example.com/api/vAppTemplate/vappTemplate-268
```

Response:

```
200 OK
Content-Type: application/vnd.vmware.vcloud.vAppTemplate+xml
...
<VAppTemplate
   xmlns="http://www.vmware.com/vcloud/v1.5"
   xmlns:ovf="http://schemas.dmtf.org/ovf/envelope/1"
   ovfDescriptorUploaded="true"
   goldMaster="false"
   status="0"
   name="Ubuntu Template"
   id="urn:vcloud:vapptemplate:268"
   href="https://vcloud.example.com/api/vAppTemplate/vappTemplate-268"
   type="application/vnd.vmware.vcloud.vAppTemplate+xml">
   ...
   <Description>Ubuntu vApp Template</Description>
   <Files>
      <File
         size="3940"
         bytesTransferred="3940"
         name="descriptor.ovf">
         <Link
            rel="upload:default"
            href="https://vcloud.example.com/transfer/.../descriptor.ovf"/>
      </File>
      <File
         size="1024"
         bytesTransferred="0"
         name="manifest.mf">
         <Link
            rel="upload:default"
            href="https://vcloud.example.com/transfer/.../manifest.mf"/>
      </File>
      <File
         size="1950489088"
         bytesTransferred="0"
         name="disk0.vmdk">
         <Link
            rel="upload:default"
            href="https://vcloud.example.com/transfer/.../disk0.vmdk"/>
      </File>
   </Files>
   ...
</VAppTemplate>
```

In this example, which omits most of the additional elements shown in "Example: Initiating the Upload," on page 57, the `ovfDescriptorUploaded` attribute has a value of `true` and the `status` attribute has a value of `0`. If the descriptor fails validation, `status` is set to `-1`, and the template contains a `Task` element whose `Error` element indicates the reason for the failure.

Each of the `File` elements includes an upload link where `rel="upload:default"` and several attributes.

size

The file size, taken from the `size` attribute of the `File` element in the OVF descriptor.

bytesTransferred

For all file references other than the descriptor, this attribute is initially set to a value of `0`, indicating that the upload has not begun. In the `File` element that references the OVF descriptor, the value of the `bytesTransferred` attribute is equal to the value of the `size` attribute, indicating that all the bytes in the descriptor were transferred.

name

The file name, taken from the `href` attribute of the `File` element in the OVF descriptor.

NOTE Upload URLs remain valid while a transfer session is in progress, and for a maximum of 60 minutes of transfer session idle time. A system administrator can change this default value. See "Retrieve or Update System Settings," on page 165.

Uploading Referenced Files

You can use a PUT request to upload each file that the vApp template references.

Prerequisites

- Verify that you uploaded the OVF descriptor. See "Uploading the OVF Descriptor," on page 59.
- Retrieve the upload URLs for all files in the package. See "Retrieving the Upload URLs," on page 59.

Procedure

1 Find the `upload:default` URL for the file you want to upload.

2 Use the `upload:default` URL to construct a PUT request for the file.

The request specifies an upload URL and a content length in bytes. See "Example: Uploading File Data," on page 61.

After all the files are uploaded, the vApp template is complete, and has a `status` attribute value of `8`. If the upload included a manifest file, the server checks each file in the upload to verify that its checksum matches the one stated in the manifest. If a checksum does not match, the template's status attribute is set to `-1` and the template contains a `Task` element whose `Error` element indicates the reason for the failure.

Example: Uploading File Data

This example shows an upload request for one of the files that an OVF package requires. The upload request is a `Content-Length` header followed by the serialized file content.

Request:

```
PUT https://vcloud.example.com/transfer/.../disk0.vmdk
Content-length: 1950489088
  ...serialized contents of file disk0.vmdk...

EOF
```

Response:

```
200 OK
```

Monitoring the Progress of an Upload

After you initiate the upload of a file referenced by a vApp template, you can monitor the progress of the upload by periodically retrieving the vApp template and tracking the value of the file's `bytesTransferred` attribute.

To monitor the progress of an upload, you can watch the `bytesTransferred` attribute of the file. Each `File` element in the template includes a `bytesTransferred` attribute whose value indicates the number of bytes that the server received.

Prerequisites

Verify that you initiated the upload of a file referenced by the vApp template.

Procedure

1 Make a GET request specifying the URL of the vApp template.

  See the request portion of "Example: Monitoring the Progress of an Upload," on page 62.

2 Compare the values of the `size` and the `bytesTransferred` attributes of each `File` element.

  When these two values are equal, the file transfer is complete.

After all the files are uploaded, the response includes final values for the `bytesTransferred` attribute of each `File`, and a `Task` that tracks the events leading up to resolution of the template with the uploaded files, as shown in "Example: Monitoring the Progress of an Upload," on page 62.

Example: Monitoring the Progress of an Upload

Request:

```
GET https://vcloud.example.com/api/vAppTemplate/vappTemplate-268
```

The complete `VAppTemplate` body is returned. This example omits most of it for clarity.

Response:

```
200 OK
Content-Type: application/vnd.vmware.vcloud.vAppTemplate+xml
...
<VAppTemplate
   ...
   name="Ubuntu Template"
   id="urn:vcloud:vapptemplate:268"
   href="https://vcloud.example.com/api/vAppTemplate/vappTemplate-268"
   type="application/vnd.vmware.vcloud.vAppTemplate+xml" ... >
   ...
   <Files>
      ...
      <File
         size="1950489088"
         bytesTransferred="500000000"
         name="disk0.vmdk">
         <Link
            rel="upload:default"
            href="https://vcloud.example.com/transfer/.../disk0.vmdk"/>
      </File>
   </Files>
   ...
</VAppTemplate>
```

Using Ranged PUT requests to Complete a Partial Upload

You typically need ranged PUT requests for very large uploads, especially when network bandwidth or latency might cause the operation to time out.

If the response to an upload progress request indicates that the upload terminated before it was complete, you can use the size and bytesTransferred values from the response to construct a ranged PUT request of the remaining contents, as shown in "Example: Ranged PUT Request to Complete a Partial Upload," on page 63.

Procedure

1 Retrieve the VAppTemplate and find the File element that references the partially uploaded file.

2 Make a PUT request that specifies a Content-Range and Content-Length and includes the serialized contents of the range.

   For Content-Range, specify the value of the File element's bytesTransferred attribute for the low end of the range and the value of its size attribute for the high end of the range. For Content-Length, subtract the value of the File element's bytesTransferred attribute from the value of its size attribute.

Example: Ranged PUT Request to Complete a Partial Upload

The following request completes the upload of the file disk0.vmdk shown in this fragment of a VAppTemplate.

```
<VAppTemplate
   ... >
   ...
   <Files>
      ...
      <File
         size="1950489088"
         bytesTransferred="500000000"
         name="disk0.vmdk">
         ...
      </File>
   </Files>
   ...
</VAppTemplate>
```

Request:

```
PUT https://vcloud.example.com/transfer/.../disk0.vmdk
Content-Range: bytes 500000000-1950489087/1950489088
Content-Length: 1450489088
   ...serialized contents of specified range...
EOF
```

Response:

```
200 OK
```

Download a vApp Template as OVF

You can download a vApp template as an OVF package. After you find the template by browsing a catalog, you can request that the template be enabled for download.

When you enable a vApp template for download, the server performs several operations to create an OVF package and make it available to the transfer service.

1. The server reconstructs the OVF descriptor using information in the vApp template. The server excludes any deployment-specific information that the template contains, and populates the descriptor's `References` element with links to files, such as `.vmdk` files, that are part of the package.
2. The server copies the reconstructed OVF descriptor to transfer service storage, along with all files that the OVF descriptor references.
3. The server updates the `vAppTemplate` with a link that contains a URL from which you can retrieve the OVF descriptor.

After the template is enabled for download, you can retrieve it to see the `download:default` URL for the descriptor, then parse the descriptor to find the URLs from which you can retrieve the referenced files.

1. Enable the vApp Template for Download on page 64

   Before you can download a vApp template, an administrator or privileged user must explicitly enable it for download.

2. Download the OVF Descriptor on page 65

   To download the OVF descriptor, you make a GET request to the `download:default` URL in the download-enabled `VappTemplate`.

3. Download a Referenced File on page 66

   After you download the OVF descriptor of a vApp template, you can examine the contents of the descriptor to discover download URLs for files that the template references.

Enable the vApp Template for Download

Before you can download a vApp template, an administrator or privileged user must explicitly enable it for download.

Prerequisites

Verify that you are logged in as an administrator or other user who has privileges to enable vApp template for download.

Procedure

1. Retrieve the template to find its `action/enableDownload` link.

   Every vApp template includes a link of the following form, where *id* is the `id` of the template:

   ```
   <Link
      rel="enable"
      href="https://vcloud.example.com/api/vAppTemplate/vappTemplate-id/action/enableDownload"/>
   ```

2. Enable the template for download.

   Make a POST request to the template's `action/enableDownload` URL.

   Request:

   ```
   POST https://vcloud.example.com/api/vAppTemplate/vappTemplate-268/action/enableDownload
   ```

Response:

```
202 Accepted
Content-Type: application/vnd.vmware.vcloud.task+xml
...
<Task
   ...
   operation="Enabling download of Virtual Application Template Ubuntu Template (268)" ... >
   ...
</Task>
```

The `Task` in the response tracks the creation of the download package.

3 When the task completes, retrieve the template, which now contains a download URL for the OVF descriptor.

  This URL remains valid while a transfer session is in progress, and for a maximum of 60 minutes of transfer session idle time. A system administrator can change this default value. See "Retrieve or Update System Settings," on page 165.

Example: vApp Template with Download URL for OVF Descriptor

Request:

```
GET https://vcloud.example.com/api/vAppTemplate/vappTemplate-268
```

Response:

```
200 OK
Content-Type: application/vnd.vmware.vcloud.vAppTemplate+xml
...
<VAppTemplate
   ovfDescriptorUploaded="true"
   status="8"
   name="Ubuntu Template"
   ... >
   ...
   <Link type="text/xml"
      rel="download:default"
      href="https://vcloud.example.com/transfer/.../descriptor.ovf"/>
   ...
</VAppTemplate>
```

Download the OVF Descriptor

To download the OVF descriptor, you make a GET request to the `download:default` URL in the download-enabled `VappTemplate`.

Prerequisites

- Verify that you are logged in as a system administrator or member of an organization in the cloud.
- Verify that you have a vApp template that is enabled for download. See "Enable the vApp Template for Download," on page 64.

Procedure

1 Examine the `VAppTemplate` to find the the `download:default` URL for the descriptor.

2 Make a GET request to the `download:default` URL to retrieve the descriptor

  See "Example: Downloading the OVF Descriptor," on page 66.

Example: Downloading the OVF Descriptor

This example downloads the OVF descriptor from the URL shown in the `href` value of the `Link` shown in "Example: vApp Template with Download URL for OVF Descriptor," on page 65. The response includes the entire `Envelope` element, only part of which appears here.

Request:

```
GET https://vcloud.example.com/transfer/..../descriptor.ovf
```

Response:

```
200 OK
Content-Type text/xml
...
<Envelope
   xmlns="http://schemas.dmtf.org/ovf/envelope/1"
   ... >
   ...
</Envelope>
```

Download a Referenced File

After you download the OVF descriptor of a vApp template, you can examine the contents of the descriptor to discover download URLs for files that the template references.

The OVF descriptor that you download from a vApp template includes the names of files that the template references. To retrieve one of these files, you must create a download URL for it by combining its name with a URL derived from the download URL that you used to retrieve the descriptor. You must retrieve all of the files to create a valid OVF package.

Prerequisites

- Verify that you are logged in as a system administrator or member of an organization in the cloud.
- Retrieve the OVF descriptor of a vApp template that was enabled for download.

Procedure

1 For each `File` element in the `References` element of the descriptor, construct a download URL.

   a Start with the URL that you used to download the descriptor.

   This URL is the `href` value of the `download:default` link that the template contains.

   b Replace the final component of that URL with the value of the `href` attribute of the `File` element.

2 Use the constructed URLs to download each file.

   See "Example: Downloading a Referenced File," on page 66.

Example: Downloading a Referenced File

The request URL shown in this example combines the URL used in the request portion of "Example: Downloading the OVF Descriptor," on page 66 with the file name shown in this `File` element:

```
<File
   ovf:href="disk0.vmdk"
   ovf:id="file1"
   ovf:size="1950489088"/>
```

Request:

```
GET https://vcloud.example.com/transfer/..../disk0.vmdk
```

Response:

```
200 OK
...
   ...serialized contents of file disk0.vmdk...

EOF
```

Note The downloaded package is valid only if the descriptor and all of its referenced files maintain the same relationship in the local file system that they had on the transfer server file system. In this case, the descriptor and `disk0.vmdk` were both in the same directory, which is the default arrangement.

Upload a Media Image

The vCloud API supports uploading virtual media such as CD-ROM and floppy disk images.

The workflow for uploading media images is similar to the one described in "Upload an OVF Package to Create a vApp Template," on page 55.

Note You cannot download media images.

Prerequisites

Verify that the following conditions are met:

- You have a media image to upload.
- You are logged in as a user who has permission to upload media images.
- You know the URL of the target vDC that will receive the upload. Retrieve the XML representation of your organization to see a list of the vDCs that it contains.

Procedure

1 Find the `media` link in the target vDC.

Retrieve the XML representation of the vDC contains a `media` link, which has the following form:

```
<Link
      rel="add"
      type="application/vnd.vmware.vcloud.media+xml"
      href="https://vcloud.example.com/api/vdc/5/media" />
```

2 POST a `media` request to the server.

The request body is a `Media` element that includes information about the virtual media item to upload. See the request portion of "Example: Upload a Media Image," on page 68.

The server uses this information to create a `Media` object, then returns a representation of the object that contains an upload URL. See the response portion of "Example: Upload a Media Image," on page 68.

3 PUT the media file contents to the `upload:default` link in the response.

The procedure is the same as the one shown in "Uploading Referenced Files," on page 61.

Example: Upload a Media Image

Request:

```
POST https://vcloud.example.com/api/vdc/5/media
Content-Type: application/vnd.vmware.vcloud.media+xml
...
<?xml version="1.0" encoding="UTF-8"?>
<Media
   xmlns="http://www.vmware.com/vcloud/v1.5"
   name="database.iso"
   size="51242131"
   imageType="iso">
   <Description>ISO database image</Description>
</Media>
```

In addition to the `File` element that contains the `upload:default` URL, the response includes an `Owner` element and several `Link` elements that the server creates.

Response:

```
Content-Type: application/vnd.vmware.vcloud.media+xml
201 Created
...
<Media
   xmlns="http://www.vmware.com/vcloud/v1.5"
   size="3121215488"
   imageType="iso"
   status="0"
   name="database.iso"
   id="urn:vcloud:media:254"
   type="application/vnd.vmware.vcloud.media+xml"
   href="https://vcloud.example.com/api/media/254" ... >
   <Link
      rel="up"
      type="application/vnd.vmware.vcloud.vdc+xml"
      href="https://vcloud.example.com/api/vdc/5" />
   <Link
      rel="remove"
      href="https://vcloud.example.com/api/media/254" />
   <Description>ISO database image</Description>
   <Files>
      <File
         size="51242131"
         bytesTransferred="0"
         name="file">
         <Link
            rel="upload:default"
            href="https://vcloud.example.com/transfer/.../file" />
      </File>
   </Files>
   <Owner
      ...
   </Owner>
</Media>
```

Copying and Moving with the vCloud API

The vCloud API provides object-specific copy operations, implemented by `clone` actions, for media images, vApp templates, and vApps. With these operations, you can create a copy of the object in the same vDC or in another vDC in the same organization.

vCloud API copy operations support an option to delete the source object after the copy is complete. Doing so when the source and target objects are in different vDCs moves the source object to the target vDC. Doing so when the source and target objects are in the same vDC renames the source object. The vCloud API does not include an explicit move operation. When you move an object by copying it and deleting its source, an intermediate object is created in the target vDC, as part of the following sequence of events:

1. The source object is copied to an intermediate object whose name is a combination of the object name and a UUID.
2. The source object is deleted.
3. The intermediate object is renamed with the name specified for the target object in the copy request.

Copy or Move a Media Image

The `cloneMedia` request makes a copy of the media image referenced in the `Source` element of the request body. The request specifies a new name and, optionally, a new description for the copy. The request can optionally include an `IsSourceDelete` element whose value specifies whether the source media image is deleted after the copy is complete. If `IsSourceDelete` is missing from the request body or present with a value of `false`, the source object remains in place after the copy is complete. Setting `IsSourceDelete` to `true` effectively moves the media image.

For more information and an example, see the `cloneMedia` operation in the schema reference.

Copy or Move a vApp Template

The `cloneVAppTemplate` request makes a copy of the vApp template referenced in the `Source` element of the request body. The request specifies a new name and, optionally, a new description for the copy. The request can optionally include an `IsSourceDelete` element whose value specifies whether the source vApp template is deleted after the copy is complete. If `IsSourceDelete` is missing from the request body or present with a value of `false`, the source object remains in place after the copy is complete. Setting `IsSourceDelete` to `true` effectively moves the vApp template.

For more information and an example, see the `cloneVAppTemplate` operation in the schema reference.

Copy or Move a vApp

The `cloneVApp` request makes a copy of the vApp referenced in the `Source` element of the request body. The request specifies a new name and, optionally, a new description for the copy. The request can optionally include an `IsSourceDelete` element whose value specifies whether to delete the source vApp after the copy is complete. If `IsSourceDelete` is missing from the request body, or present with a value of `false`, the source object remains in place after the copy is complete. Setting `IsSourceDelete` to `true` effectively moves the vApp.

NOTE You cannot copy or move a vApp that is deployed.

For more information and an example, see the `cloneVApp` operation in the schema reference.

Capturing and Importing vApps

You can capture a vApp to create a vApp template from it. If you are a system administrator, you can also import vApps and vApp templates from vSphere.

Every vDC includes a `Link` of the following form:

```
<Link
   rel="add"
   type="application/vnd.vmware.vcloud.captureVAppParams+xml"
   href="https://vcloud.example.com/api/vdc/5/action/captureVApp"/>
```

You can POST a `CaptureVAppParams` request to this link to capture an undeployed vApp. The operation returns a `VAppTemplate` that you can add to a catalog. Instantiating this template recreates the vApp from which it was captured. Most elements of a vApp template are read-only, but you can instantiate a template, modify the resulting vApp, and then capture it to create a modified version of the template.

For more information and an example, see the `captureVApp` operation in the schema reference.

Importing vApps or vApp Templates from vSphere

A system administrator can import vApps and vApp templates from vSphere. See "Import a Virtual Machine from vCenter," on page 189.

Cataloging vApp Templates and Media Images

Catalogs can contain references to vApp templates and media images. A system administrator or a privileged member of the organization that owns the catalog can create and remove these references.

Although you can retrieve references to vApp templates and media images directly from the vDC to which they were uploaded, it is common practice to place references to such assets in one of an organization's catalogs. When you place the references in a catalog, the assets are easier to discover, because a catalog can include assets from all vDCs in an organization. You also have more flexible administrative control over them, because you can restrict access to catalogs and the items in them to specific users and groups. Assets such as vApp templates are not enabled for most uses until you include them in a catalog. For example, you cannot instantiate a vApp template that is not included in a catalog. A media image that is not included in a catalog cannot be copied or inserted by anyone but its owner.

Add an Item to a Catalog

A catalog can contain references to vApp templates and media images from any vDC in an organization. A vApp template or media image can be listed in at most one catalog.

Prerequisites

Verify that you are logged in as a user with the Catalog Author role, as an organization administrator, or as a system administrator.

Procedure

1 Retrieve the XML representation of your organization and look for links to the catalogs and vDCs it contains.

  Links to catalogs have the following form:

```
<Link rel="down"
   type="application/vnd.vmware.vcloud.catalog+xml"
   name="CatalogName"
   href="https://vcloud.example.com/api/catalog/32"/>
```

Links to vDCs have the following form:

```
<Link rel="down"
   type="application/vnd.vmware.vcloud.vdc+xml"
   name="vDCName"
   href="https://vcloud.example.com/api/vdc/5"/>
```

2 Browse the vDCs in your organization to find the `ResourceEntity` element that represents the item to add to the catalog.

Resource entities are contained in the `ResourceEntities` element of a `Vdc` .

```
<Vdc
   xmlns="http://www.vmware.com/vcloud/v1.5"
   name="ExampleVdc01"
   type="application/vnd.vmware.vcloud.vdc+xml"
   href="https://vcloud.example.com/api/vdc/5">
   ...
   <ResourceEntities>
      <ResourceEntity
         href="https://vcloud.example.com/api/vAppTemplate/vappTemplate-111"
         type="application/vnd.vmware.vcloud.vAppTemplate+xml"
         name="Ubuntu Template with vsftpd"/>
      <ResourceEntity
         type="application/vnd.vmware.vcloud.vAppTemplate+xml"
         name="NewVAppTemplate"
         href="https://vcloud.example.com/api/vAppTemplate/vappTemplate-559" />
      ...
   </ResourceEntities>
   ...
</Vdc>
```

3 Retrieve the XML representation of the catalog to which to add the item.

Use a request like this one:

```
GET https://vcloud.example.com/api/catalog/32
```

4 Examine the response to locate the `Link` element that contains the URL for adding items to the catalog.

This element has a `rel` attribute value of `add` and a `type` attribute value of `application/vnd.vmware.admin.catalogItem+xml`, as this example shows:

```
<Link
   rel="add"
   type="application/vnd.vmware.admin.catalogItem+xml"
   href="https://vcloud.example.com/api/catalog/32/catalogItems"/>
```

5 Create a `CatalogItem` element that contains a reference to the `ResourceEntity`

See "Example: ResourceEntity and Corresponding CatalogItem," on page 72 for an example.

6 POST the `CatalogItem` body to the catalog's `rel="add"` URL.

Step 4 explains how to find this URL.

Example: ResourceEntity and Corresponding CatalogItem

This example starts with this `ResourceEntity`, which references a vApp template.

```
<ResourceEntity
   href="https://vcloud.example.com/api/vAppTemplate/vappTemplate-111"
   type="application/vnd.vmware.vcloud.vAppTemplate+xml"
   name="Ubuntu Template with vsftpd"/>
```

Follow these guidelines to create a `CatalogItem` from a `ResourceEntity`.

- The `name` attribute of the `CatalogItem` can be the same as the one in the `ResourceEntity`, or you can make up a new value for `name`.
- The `Description` element of the `CatalogItem` can be the same as the `Description` element for the object that the `ResourceEntity` element references, or you can create a `Description` element.
- The `href` attribute of the `Entity` element in the `CatalogItem` must have the same value as the `href` attribute of the `ResourceEntity` that the `CatalogItem` references.

This request creates the catalog item that is retrieved in "Example: Retrieve a Catalog Item," on page 29. The response echoes the request, and includes links that an administrator or catalog owner can use to manage the `CatalogItem` and its metadata.

Request:

```
POST https://vcloud.example.com/api/catalog/32/catalogItems
Content-Type: application/vnd.vmware.vcloud.catalogItem+xml
...
<?xml version="1.0" encoding="UTF-8"?>
<CatalogItem
   name="Ubuntu Template with vsftpd"
   type="application/vnd.vmware.vcloud.catalogItem+xml"
   xmlns="http://www.vmware.com/vcloud/v1.5">
   <Description>Approved template for public FTP sites</Description>
   <Entity
      href="https://vcloud.example.com/api/vAppTemplate/vappTemplate-111"/>
</CatalogItem>
```

Response:

```
200 OK
Content-Type: application/vnd.vmware.vcloud.catalogItem+xml
...
<CatalogItem
   xmlns="http://www.vmware.com/vcloud/v1.5"
   name="Ubuntu Template with vsftpd"
   id="urn:vcloud:catalogitem:221"
   href="https://vcloud.example.com/api/catalogItem/221" ... >
  <Link
      rel="up"
      type="application/vnd.vmware.vcloud.catalog+xml"
      href="https://vcloud.example.com/api/catalog/32" />
   <Link
      rel="down"
      type="application/vnd.vmware.vcloud.metadata+xml"
      href="https://vcloud.example.com/api/catalogItem/221/metadata" />
   <Link
      rel="edit"
      type="application/vnd.vmware.vcloud.catalogItem+xml"
      href="https://vcloud.example.com/api/catalogItem/221" />
   <Link
      rel="remove"
      href="https://vcloud.example.com/api/catalogItem/221" />
   <Description>Approved template for public FTP sites</Description>
   <Entity
      href="https://vcloud.example.com/api/vAppTemplate/vappTemplate-111"
      type="application/vnd.vmware.vcloud.vAppTemplate+xml"
      name="Ubuntu Template with vsftpd"/>
</CatalogItem>
```

Remove an Item from a Catalog

An organization administrator or a user with adequate permissions can remove a `CatalogItem` by making a DELETE request to its `rel="remove"` link.

For more information and an example, see the `catalogItem` operation in the schema reference.

Prerequisites

Verify that you are logged in as a user with the Catalog Author role, as an organization administrator of the organization that owns the catalog, or as a system administrator.

Procedure

1 Retrieve the catalog item from the catalog.

2 Locate the `rel="remove"` link in the `CatalogItem` element.

3 Make a DELETE request to the `href` value of the `rel="remove"` link in the `CatalogItem`.

View or Change the Owner of an Object

You can view the owner of a `VApp`, `VAppTemplate`, or `Media` object by making a GET request to the object's `owner` link. You can change the owner of a `VApp`, but not that of a `VAppTemplate` or `Media` object. An administrator can view or change the owner of a `Catalog` object.

The initial owner of a `VApp`, `VAppTemplate`, `Catalog`, or `Media` object is the user who created it. Ownership is expressed in an `Owner` element that the object representation contains. This element includes a `User` element that references the owner.

Prerequisites

- To change the owner of a vApp, you must be the vApp owner, an organization administrator, or the system administrator.
- To change the owner of a catalog, you must be an organization administrator or the system administrator.

Procedure

1 Retrieve the `Owner` element from the object.

This element includes a reference to the current owner and an `edit` URL you can use to change the owner. This example retrieves the owner of a vApp.

```
GET https://vcloud.example.com/api/vApp/vapp-7/owner
```

2 Modify the Owner element to specify a different User.

The user must be a member of the organization that contains the object.

NOTE You cannot modify the Owner of a Media or VAppTemplate object.

3 To change the owner of a vApp, make a PUT request to the Owner element's rel="edit" URL and supply an Owner element in the request body.

The User element in the Owner element references the new owner. See "Example: Change the Owner of a vApp," on page 74.

Example: Change the Owner of a vApp

Request:

```
PUT https://vcloud.example.com/api/vApp/vapp-7/owner
Content-type: application/vnd.vmware.vcloud.owner+xml
...
<?xml version="1.0" encoding="UTF-8"?>
<Owner
   xmlns="http://www.vmware.com/vcloud/v1.5">
   <User
      type="application/vnd.vmware.admin.user+xml"
      href="https://vcloud.example.com/api/admin/user/120" />
</Owner>
```

Response:

```
204 No Content
```

Deploying and Operating vApps 5

The vCloud API supports programmatic access to a range of self-service datacenter operations that allow users to create, configure, deploy, and operate vApps.

The initial configuration of a vApp is established in the OVF package on which its source template is based. In the vCloud API, vApp templates are based on OVF 1.0. These templates can be retrieved from catalogs and transformed into virtual systems, called vApps, through a process called instantiation, which binds a template's abstract resource requirements to resources available in a vDC.

About OVF

OVF is a widely accepted standard format that applies to many virtualization technologies.

- Virtual machines and appliances are distributed as OVF packages by many vendors.
- Many vendors, including VMware, offer tools that simplify creating and customizing OVF, support converting virtual machines on existing virtualization platforms to OVF, or both.
- OVF can express the complex relationships between virtual appliances in enterprise applications. The author of the appliance can handle most of the complexity, rather than the user who deploys it.
- OVF is extensible, allowing new policies and requirements to be inserted by ISVs and implemented by the virtualization platforms that support them without requiring changes to other clients, other platforms, or the vCloud API itself.

Administrators and advanced users should become familiar with the details of the OVF standard before developing applications with the vCloud API. The complete OVF specification document is available at http://www.dmtf.org/standards/published_documents/DSP0243_1.0.0.pdf. An informative white paper on OVF is available at http://www.dmtf.org/standards/published_documents/DSP2017_1.0.0.pdf.

A virtual machine is typically made up of one or more virtual disk files that contain the operating system and applications that run on the virtual machine, and a configuration file containing metadata that describe how the virtual machine is configured and deployed. An OVF package includes these components, as well as optional certificate and manifest files. The package can be distributed and stored as a collection of individual files, or as a single archive (OVA) file. The vCloud API does not support uploading or downloading OVA files.

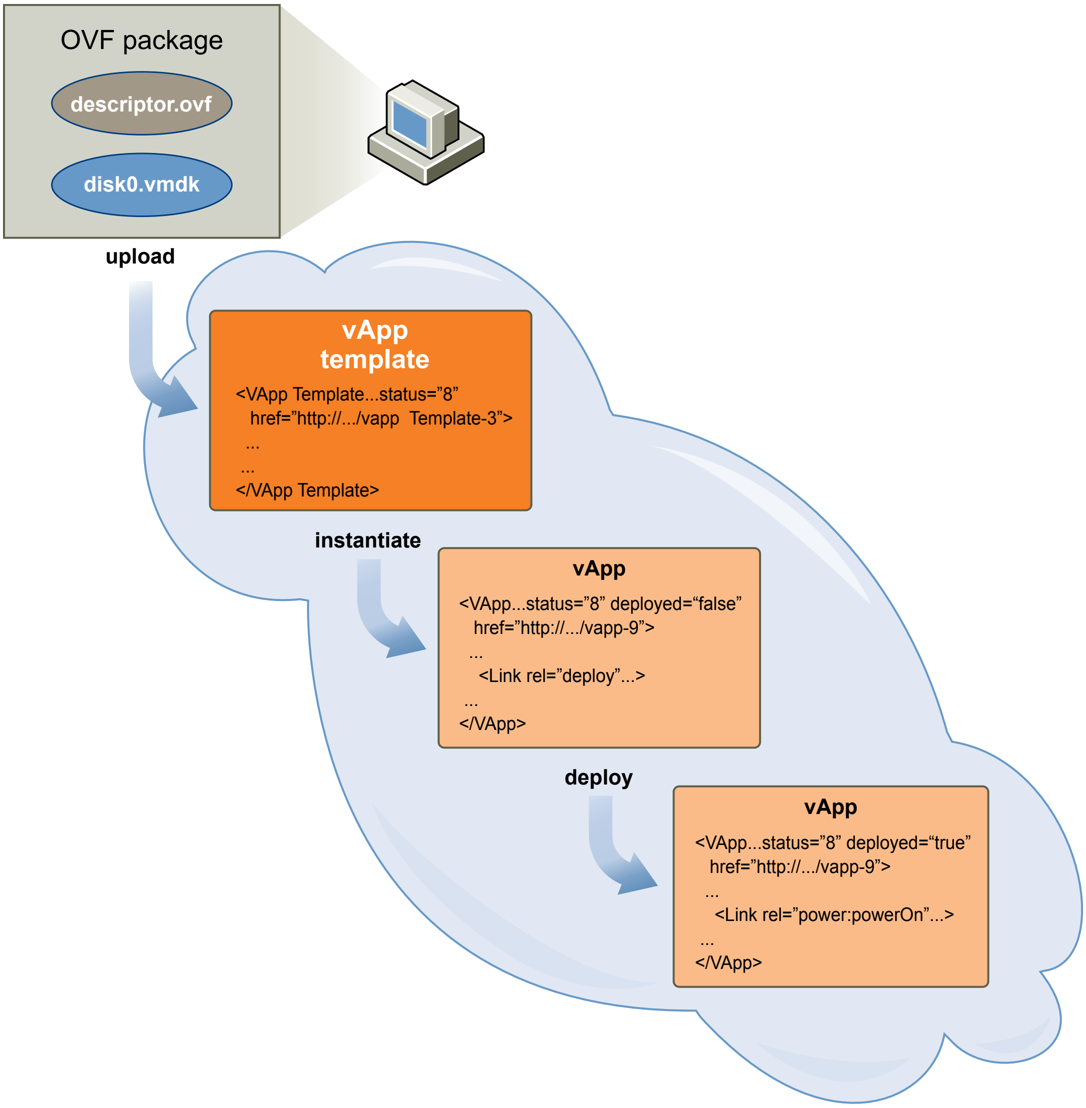
vApp Life Cycle

A vApp contains one or more `Vm` elements, which represent individual virtual machines. It also contains information that defines operational details for the vApp and the virtual machines that it contains. The vApp lifecycle includes several distinct states:

- An OVF package, the form in which vApps are typically distributed.
- A vApp template, created when a client uploads an OVF package to a vDC.

- An undeployed vApp, created when a vApp template is instantiated without also being deployed, or a deployed vApp is undeployed.
- A deployed vApp, ready to be powered on and operated. Instantiation can include deployment, power-on, or both.

Figure 5-1. vApp State Transitions


OVF package
descriptor.ovf
disk0.vmdk
upload
vApp template
<VApp Template...status="8"
href="http://.../vapp Template-3">
...
...
</VApp Template>
instantiate
vApp
<VApp...status="8" deployed="false"
href="http://.../vapp-9">
...
<Link rel="deploy"...>
...
</VApp>
deploy
vApp
<VApp...status="8" deployed="true"
href="http://.../vapp-9">
...
<Link rel="power:powerOn"...>
...
</VApp>

This chapter includes the following topics:

- "Summary of vCloud API vApp and Virtual Machine Operations Requests," on page 77
- "Create a vApp From a Template," on page 78
- "Compose a vApp From Existing Virtual Machines," on page 80
- "Recompose a vApp to Add or Remove Virtual Machines," on page 83
- "Operate a vApp," on page 85
- "Configuring vApps and Virtual Machines," on page 86

Summary of vCloud API vApp and Virtual Machine Operations Requests

These vCloud API operations requests create, manage, operate, and delete vApps and the virtual machines that they contain.

- *API-URL* is a URL of the form `https://vcloud.example.com/api`.
- *id* is a unique identifier in the form of a UUID, as defined by RFC 4122.
- *VApp-or-Vm-URL* is a URL of the form *API-URL*/vApp/vapp-*id* (for a `VApp` object) or *API-URL*/vApp/vm-*id* (for a `Vm` object)
- *Vm-URL* is a URL of the form *API-URL*/vApp/vm-*id*

Request URLs are always available when you GET the representation of the object on which they operate. URL forms shown in the table are for reference purposes only. For more information about the requests and responses for each operation, see "About the Schema Reference Downloadable Archive," on page 19.

Table 5-1. Summary of vCloud API Operations Requests

Operation	Request	Request Body	Response
Instantiate a vApp template	POST *API-URL*/vdc/*id*/action/instantiateVAppTemplate	`InstantiateVAppTemplateParams`	`VApp`
Compose a vApp	POST *API-URL*/vdc/*id*/action/composeVApp	`ComposeVAppParams`	`VApp`
Recompose a vApp to add or remove virtual machines	POST *API-URL*/vApp/vapp-*id*/action/recomposeVApp	`RecomposeVAppParams`	`Task`
Deploy a vApp or Virtual Machine	POST *VApp-or-Vm-URL*/action/deploy	`DeployVAppParams`	`Task`
Undeploy a vApp or Virtual Machine	POST *VApp-or-Vm-URL*/action/undeploy	`UndeployVAppParams`	`Task`
Power On a vApp or Virtual Machine	POST *VApp-or-Vm-URL*/action/powerOn	None	`Task`
Power Off a vApp or Virtual Machine	POST *VApp-or-Vm-URL*/action/powerOff	None	`Task`
Reset a vApp or Virtual Machine	POST *VApp-or-Vm-URL*/action/reset	None	`Task`
Suspend a vApp or Virtual Machine	POST *VApp-or-Vm-URL*/action/suspend	None	`Task`
Discard the Suspended State of a vApp or Virtual Machine	POST *VApp-or-Vm-URL*/action/discardSuspendedState	None	`Task`
Shut Down a vApp or Virtual Machine	POST *VApp-or-Vm-URL*/action/shutdown	None	`Task`
Reboot a vApp or Virtual Machine	POST *VApp-or-Vm-URL*/action/reboot	None	`Task`
Retrieve product sections of a vApp or virtual machine [NEW]	GET *VApp-or-Vm-URL*/productSections	None	`ProductSectionList`
Update product sections of a vApp or virtual machine [NEW]	PUT *VApp-or-Vm-URL*/productSections	`ProductSectionList`	`Task`

Table 5-1. Summary of vCloud API Operations Requests (Continued)

Operation	Request	Request Body	Response
Retrieve product sections of a vApp template [NEW]	GET *API-URL*/vAppTemplate/vappTemplate-*id*/productSections	None	`ProductSectionList`
Update product sections of a vApp template [NEW]	PUT *API-URL*/vAppTemplate/vappTemplate-*id*/productSections	`ProductSectionList`	`Task`
Retrieve version of VMware Tools installed on a virtual machine [NEW]	GET *Vm-URL*/runtimeInfoSection	None	`RuntimeInfoSection`
Install VMware Tools on a virtual machine [NEW]	POST *Vm-URL*/action/installVMwareTools	None	`Task`
Consolidate a virtual machine [NEW]	POST *Vm-URL*/action/consolidate	None	`Task`
Upgrade the hardware version of a virtual machine [NEW]	POST *Vm-URL*/action/upgradeHardwareVersion	None	`Task`
Insert Media Into a Virtual Machine	POST *Vm-URL*/action/insertMedia	`MediaInsertOrEjectParams`	`Task`
Eject Media from a Virtual Machine	POST *Vm-URL*/action/ejecttMedia	`MediaInsertOrEjectParams`	`Task`
List Media Devices of a Virtual Machine	GET *Vm-URL*/virtualHardwareSection/media	None	`RasdItemsList`
Get a Request for User Input	GET *Vm-URL*/question	None	`VmPendingQuestion`
Provide Requested User Input	POST *Vm-URL*/question/action/answer	`VmQuestionAnswer`	204 No Content
Get a Screen Thumbnail for a Virtual Machine	GET *Vm-URL*/screen	None	Returns a screen thumbnail (Content-type: image/png) if one is available. Otherwise returns null (Content-Length: 0).
Get a Screen Ticket for a Virtual Machine	POST *Vm-URL*/screen/action/acquireTicket	None	`ScreenTicket`

Create a vApp From a Template

An `instantiateVAppTemplate` request creates a vApp from a vApp template. By default, this operation also deploys the vApp and powers it on.

To create a vApp from a vApp template, you must bind the template's abstract resource requirements, such as network connections, storage resources, memory, and CPU capacity, to appropriate resources in the target vDC. This binding operation is called instantiation.

For an example of a simple instantiation request, see "Deploy the vApp," on page 31. You can also specify additional parameters as part of instantiation.

Template contents that might influence composition of the request body include the following sections:

- A `NetworkConnectionSection` that specifies network connection details for a virtual machine. Unless you want to create a vApp in which none of the virtual machines are connected to a network, your instantiation parameters must include at least one `NetworkConfigSection` that defines a vApp network, and that section must include a `NetworkConfig` element whose `networkName` attribute value matches the value of the `network` attribute of the `NetworkConnection` of each `Vm` in the template. If this attribute has the value `none` or is missing, the `Vm` can connect to any network. If the template contains `Vm` elements that specify different names for their network connections, you must create a vApp network for each.
- One or more `EulaSection` elements that specify licensing terms or other conditions that you must accept before creating the vApp. The `InstantiateVAppTemplateParams` element can include an `AllEULAsAccepted` element whose value indicates whether you accept all EULA terms included in the template. If a vApp template includes any `ovf:EulaSection` elements, `AllEULAsAccepted` must be set to a value of `true`. Otherwise, instantiation fails.
- A `LeaseSettingsSection`. If this section is present and specifies settings that are appropriate for the vApp, you do not need to modify it. If it is absent or empty, the vApp is created with your organization's default lease settings. If you specify new lease settings in a `LeaseSettingsSection` that you provide as part of instantiation, those settings replace any existing settings and override your organization's defaults.

Prerequisites

Verify that you are logged in as a system administrator or member of an organization in the cloud.

Procedure

1 Retrieve the XML representation of the vApp template.

Make a GET request to the URL provided in the `href` attribute of the `Entity` contained by the `CatalogItem` that references the template.

2 Examine the template to determine the set of instantiation parameters that the request must include.

3 Create an `InstantiateVAppTemplateParams` element.

See "Example: Instantiate a vApp Template," on page 79 for guidelines.

4 Make a POST request to the `action/instantiateVAppTemplate` URL of the vDC.

Supply the `InstantiateVAppTemplateParams` element as the request body.

The server takes the requested action and returns a `VApp` element. The element has a `status` attribute value of `0`, meaning it is unresolved because it is still being constructed. It also contains a `Task` element that tracks the progress of the request.

Example: Instantiate a vApp Template

This `InstantiateVAppTemplateParams` request extends the request shown in "Example: Deploying a vApp," on page 32 to include additional elements in its `InstantiationParams`:

- A `LeaseSettingsSection` that specifies custom lease settings, overriding the settings that would otherwise be inherited from the organization.
- An acknowledgement of `EulaSection` acceptance, supplied in the `AllEULAsAccepted` element. If the template does not include `EulaSection` elements, you can omit this acknowledgement.

For more information and a list of sections that you can include in `InstantiationParams`, see "Configuring a vApp," on page 87.

Request:

```
POST https://vcloud.example.com/api/vdc/5/action/instantiateVAppTemplate
Content-Type: application/vnd.vmware.vcloud.instantiateVAppTemplateParams+xml
...
<?xml version="1.0" encoding="UTF-8"?>
<InstantiateVAppTemplateParams
   xmlns="http://www.vmware.com/vcloud/v1.5"
   name="Linux FTP server"
   deploy="true"
   powerOn="true"
   xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
   xmlns:ovf="http://schemas.dmtf.org/ovf/envelope/1">
   <Description>Example FTP Server</Description>
   <InstantiationParams>
      <NetworkConfigSection>
         <ovf:Info>Configuration parameters for logical networks
         </ovf:Info>
         <NetworkConfig
            networkName="vAppNetwork">
            <Configuration>
               <ParentNetwork
                  href="https://vcloud.example.com/api/network/54" />
               <FenceMode>bridged</FenceMode>
            </Configuration>
         </NetworkConfig>
      </NetworkConfigSection>
      <LeaseSettingsSection
         type="application/vnd.vmware.vcloud.leaseSettingsSection+xml">
         <ovf:Info>Lease Settings</ovf:Info>
         <StorageLeaseInSeconds>172800</StorageLeaseInSeconds>
         <StorageLeaseExpiration>2010-04-11T08:08:16.438-07:00</StorageLeaseExpiration>
      </LeaseSettingsSection>
   </InstantiationParams>
   <Source
      href="https://vcloud.example.com/api/vAppTemplate/vappTemplate-111" />
   <AllEULAsAccepted>true</AllEULAsAccepted>
</InstantiateVAppTemplateParams>
```

The response is a sparsely populated VApp element, as shown in the response portion of "Example: Deploying a vApp," on page 32.

Compose a vApp From Existing Virtual Machines

With the vCloud API composeVApp operation, you can build a vApp from existing virtual machines, including virtual machines contained by vApps and vApp templates to which you have access.

Every vDC includes a link to a composeVApp operation, which creates a new vApp in it. To compose a vApp, POST a composeVApp request to this link. The request body is a ComposeVAppParams element, which includes the following information:

- An InstantiationParams element that can include any of the section types listed under "Configuring a vApp," on page 87. This is where you define the vApp network to which all the virtual machines in the composed vApp connect, and custom vApp lease settings and startup parameters for the virtual machines.
- An optional Description of the composed vApp.

- Zero or more SourcedItem elements, each of which defines a source of virtual machines to include in the composition. Each SourcedItem must contain a Source element that references a composition source, the href of a Vm, VApp, or VAppTemplate. If the Source element references a virtual machine, the SourcedItem can include any of the following elements:
  - An InstantiationParams element specific to that virtual machine. This element can include any of the section types listed under "Configuring a Virtual Machine," on page 87.
  - A NetworkAssignment element that specifies how the network connections in the virtual machine are mapped to vApp networks defined in the InstantiationParams element that applies to the composed vApp.
  - A VAppScopedLocalId element that provides a unique identifier for the virtual machine in the scope of the composed vApp.

  If the Source element references a vApp or vApp template, all Vm elements from each composition source become peers in the Children collection of the composed vApp.
- If any of the composition items is subject to a EULA, the ComposeVAppParams element must include an AllEULAsAccepted element that has a value of true, indicating that you accept the EULA. Otherwise, composition fails.

The composed vApp must be deployed and powered on before you can use it.

Prerequisites

Verify that you are logged in as a system administrator or member of an organization in the cloud.

Procedure

1 Find the composeVApp link in the target vDC.

The XML representation of the vDC contains a composeVapp link, which has the following form:

```
<Link
   rel="add"
   type="application/vnd.vmware.vcloud.composeVAppParams+xml"
   href="https://vcloud.example.com/api/vdc/5/action/composeVApp" />
```

2 Create a ComposeVappParams element that specifies the details of the composition.

3 POST the ComposeVappParams element to the composeVapp link of the target vDC.

See the Request portion of "Example: Compose a vApp," on page 81.

Example: Compose a vApp

This request specifies two vAppTemplate items and one Vm item. The Vm item requires InstantiationParams that modify its NetworkConnectionSection to specify the vApp network created for this vApp. The vAppTemplate items inherit this setting from the base InstantiationParams element that appears before the first SourcedItem is specified.

NOTE Virtual machines specified in Source elements must be powered off or this operation will fail. You can use a query like this one to return a list of references to powered-off virtual machines that you have access to.

```
https://vcloud.example.com/api/query?type=adminVM&format=references&filter=status==POWERED_OFF
```

See Chapter 9, "Using the Query Service," on page 201.

Request:

```
POST https://vcloud.example.com/api/vdc/5/action/composeVApp
Content-Type: application/vnd.vmware.vcloud.composeVAppParams+xml
...
<?xml version="1.0" encoding="UTF-8"?>
<ComposeVAppParams
   name="Example Corp’s CRM Appliance"
   xmlns="http://www.vmware.com/vcloud/v1.5"
   xmlns:ovf="http://schemas.dmtf.org/ovf/envelope/1">
   <Description>Composed CRM Appliance</Description>
   <InstantiationParams>
      <NetworkConfigSection>
         <ovf:Info>Configuration parameters for logical networks</ovf:Info>
         <NetworkConfig
            networkName="CRMApplianceNetwork">
            <Configuration>
               <ParentNetwork
                  href="https://vcloud.example.com/api/network/54" />
               <FenceMode>natRouted</FenceMode>
            </Configuration>
         </NetworkConfig>
      </NetworkConfigSection>
   </InstantiationParams>
   <SourcedItem>
      <Source
         href="https://vcloud.example.com/api/vApp/vm-4" />
      <InstantiationParams>
         <NetworkConnectionSection
            xmlns:ovf="http://schemas.dmtf.org/ovf/envelope/1"
            type="application/vnd.vmware.vcloud.networkConnectionSection+xml"
            href="https://vcloud.example.com/api/vApp/vm-4/networkConnectionSection/"
            ovf:required="false">
            <ovf:Info />
            <PrimaryNetworkConnectionIndex>0</PrimaryNetworkConnectionIndex>
            <NetworkConnection
               network="CRMApplianceNetwork">
               <NetworkConnectionIndex>0</NetworkConnectionIndex>
               <IsConnected>true</IsConnected>
               <IpAddressAllocationMode>DHCP</IpAddressAllocationMode>
            </NetworkConnection>
         </NetworkConnectionSection>
      </InstantiationParams>
   </SourcedItem>
   <SourcedItem>
      <Source
         href="https://vcloud.example.com/api/vAppTemplate/vappTemplate-114" />
   </SourcedItem>
   <SourcedItem>
      <Source
         href="https://vcloud.example.com/api/vAppTemplate/vappTemplate-190" />
   </SourcedItem>
   <AllEULAsAccepted>true</AllEULAsAccepted>
</ComposeVAppParams>
```

Response:

```
201 Created
Content-Type: application/vnd.vmware.vcloud.vApp+xml
...
<VApp
   name="Example Corp’s CRM Appliance"
   type="application/vnd.vmware.vcloud.vApp+xml"
   status="8"
   href="https://vcloud.example.com/api/vApp/vapp-33" ...>
   <Link
      rel="up" type="application/vnd.vmware.vcloud.vdc+xml"
      href="https://vcloud.example.com/api/vdc/5"/>
   ...
   <Description>Composed CRM Appliance</Description>
   ...
   <Tasks>
      <Task operation="Composing Virtual Application Example Corp’s CRM Appliance (33)" ...>
         ...
      </Task>
   </Tasks>
</VApp>
```

Recompose a vApp to Add or Remove Virtual Machines

The vCloud API supports recomposition of a vApp to add or remove virtual machines. To recompose a vApp, make a `recomposeVApp` request, supplying a `RecomposeVAppParams` element as the request body.

The `RecomposeVAppParams` element allows an arbitrary number of `DeleteItem` elements, but is otherwise identical to `ComposeVAppParams`. This means that in addition to adding or removing virtual machines, a `recomposeVApp` request can also change the name and description of the vApp, and can supply new `InstantiationParams` to change various sections of the composed vApp or any of the added virtual machines.

Unlike a `composeVapp` request, which operates on a vDC and creates a new vApp, a `recomposeVapp` request operates on (and modifies) an existing vApp.

Prerequisites

Verify that you are logged in as a system administrator or member of an organization in the cloud.

Procedure

1 Find the `recomposeVApp` link in the target vApp.

   The XML representation of a vApp contains a `recomposeVapp` link, which has the following form:

```
<Link
      rel="recompose"
      type="application/vnd.vmware.vcloud.recomposeVAppParams+xml"
      href="https://vcloud.example.com/api/vdc/5/action/recomposeVApp" />
```

2 Create a `RecomposeVappParams` element that specifies the details of the recomposition.

   See “Example: Recompose a vApp,” on page 84.

3 POST the `RecomposeVappParams` element to the `recomposeVapp` link of the target vApp.

Example: Recompose a vApp

This example uses the `recomposeVApp` operation to modify this vApp, which contains three virtual machines. Only a few of the elements in the vApp appear here.

```
<VApp
   name="Example Corp’s CRM Appliance"
   type="application/vnd.vmware.vcloud.vApp+xml"
   status="8"
   href="https://vcloud.example.com/api/vApp/vapp-33" ...>
   <Link
      rel="up"
      type="application/vnd.vmware.vcloud.vdc+xml"
      href="https://vcloud.example.com/api/vdc/5"/>
   ...
   <Children>
      <Vm
         status="8"
         name="CRM-DB"
         href="https://vcloud.example.com/api/vApp/vm-7" ...>
         ...
      </Vm>
      <Vm
         status="8"
         name="CRM-CRM"
         href="https://vcloud.example.com/api/vApp/vm-44" ...>
         ...
      </Vm>
      <Vm
         status="8"
         name="CRM-HTTP"
         href="https://vcloud.example.com/api/vApp/vm-45" ...>
         ...
      </Vm>
   </Children>
...
</VApp>
```

The request removes one of the virtual machines from the vApp and creates a `StartupSection` that specifies a startup order for the remaining virtual machines.

Request:

```
POST https://vcloud.example.com/api/vApp/vapp-33/action/recomposeVApp
Content-Type: application/vnd.vmware.vcloud.recomposeVAppParams+xml
...
<?xml version="1.0" encoding="UTF-8"?>
<RecomposeVAppParams
   name="Example Corp’s CRM Appliance"
   xmlns="http://www.vmware.com/vcloud/v1.5"
   xmlns:ovf="http://schemas.dmtf.org/ovf/envelope/1">
   <Description>Composed CRM Appliance</Description>
   <InstantiationParams>
      <NetworkConfigSection>
         <ovf:Info>Configuration parameters for logical networks</ovf:Info>
         <NetworkConfig
```

```
            networkName="CRMApplianceNetwork">
            <Configuration>
               <ParentNetwork
                  href="https://vcloud.example.com/api/network/54" />
               <FenceMode>natRouted</FenceMode>
            </Configuration>
         </NetworkConfig>
      </NetworkConfigSection>
      <ovf:StartupSection
         xmlns:vcloud="http://www.vmware.com/vcloud/v1.5"
         vcloud:type="application/vnd.vmware.vcloud.startupSection+xml">
         <ovf:Info>VApp startup section</ovf:Info>
         <ovf:Item
            ovf:order="0"
            ovf:id="CRM-DB" />
         <ovf:Item
            ovf:order="1"
            ovf:id="CRM-CRM" />
      </ovf:StartupSection>
   </InstantiationParams>
   <AllEULAsAccepted>true</AllEULAsAccepted>
   <DeleteItem
      href="https://vcloud.example.com/api/vApp/vm-45" />
</RecomposeVAppParams>
```

Response:

```
202 Accepted
Content-Type: application/vnd.vmware.vcloud.task+xml
...
<Task ...
   operation="Updating Virtual Application Example Corp’s CRM Appliance (33)" ...>
...
</Task>
```

Note Virtual machines specified in `Source` elements must be powered off or this operation will fail. You can use a query like this one to return a list of references to powered-off virtual machines that you have access to.

```
https://vcloud.example.com/api/query?type=adminVM&format=references&filter=status==POWERED_OFF
```

See Chapter 9, “Using the Query Service,” on page 201.

Operate a vApp

`vApp` and `Vm` elements include a number of action links. You can use these links to operate a vApp or one of its virtual machines by making requests such as power on, suspend, power off, undeploy, and so on.

Only those action links that are valid for the vApp or virtual machine in its current state are returned. For example, if a vApp is instantiated but not deployed, only the links for `deploy` and `remove` are returned. For a vApp that is powered on, links for all actions except `powerOn` are returned. Some requests apply only to vApps, some apply only to virtual machines (`Vm` objects), and some apply to both.

- A request made to a vApp `power` URL invokes the requested operation on each of the virtual machines in the `Children` element of the vApp, in the order specified in its `ovf:StartupSection` element. This element, if present, specifies a start order and related properties for each member of a `VirtualSystemCollection` (each `Vm` contained by the `Children` element). If the element is not present, all members are started up at the same time. The same logic applies to shutdown, reboot, and similar operations.

- A request made to a Vm power URL affects only that virtual machine.
- See "Summary of vCloud API vApp and Virtual Machine Operations Requests," on page 77 for details of each request.

Prerequisites

Verify that you are logged in as a system administrator or member of an organization in the cloud.

Procedure

1 Retrieve the XML representation of the vApp to find the action links it contains.

Use a request like this one:

```
GET https://vcloud.example.com/api/vApp/vapp-7
```

2 POST a request to the URL that implements the desired action.

Many of these requests do not require a body.

The server takes the requested action and returns a Task element that tracks the progress of the request.

Configuring vApps and Virtual Machines

You specify configuration details for vApps and the virtual machines they contain in ovf:SectionType elements contained in a VApp or Vm element. You can include most of these sections in the InstantiationParams that you supply when you instantiate a vApp template. You can also retrieve, modify, and update these sections to reconfigure a deployed vApp.

You establish the initial configuration of a vApp in the OVF package on which its source template is based. You cannot modify most sections of the template, but you can update the configuration during the following operations:

- When you create a vApp by making an instantiateVAppTemplate request, you can update its configuration by including modified sections in the request's InstantiationParams element.
- When you compose or recompose a vApp, you can update its configuration by including modified sections in the request's InstantiationParams element.
- When you update any of the modifiable sections in a deployed vApp or the virtual machines that it contains.

You can capture the reconfigured vApp to create a vApp template that preserves your modifications. See "Capturing and Importing vApps," on page 70.

Updating a vApp Template

You can update the following sections of a vApp template.

CustomizationSection You can use a PUT request to update this section, which contains a Boolean value, CustomizeOnInstantiate, that specifies whether guest customization should be run during instantiation. You cannot modify this element if the vApp has been added to a catalog.

Name and description You can use a PUT request to update the value of the template's name attribute or the contents of its Description element.

Owner

You can use a PUT request to update the value of the template's `Owner` element. See "View or Change the Owner of an Object," on page 73.

ProductSection elements

You can retrieve or update the template's `ProductSection` elements, which provide a way to pass runtime information to the virtual machines defined in the template. See "Retrieve or Modify ProductSection Elements," on page 102.

Except for the `Owner` element, all updates to a vApp template propagate to the vApp during instantiation. The owner of the vApp is reset to the identity of the user who instantiates the template.

Configuring a vApp

You can include the following sections in `InstantiationParams` for a vApp. You can also modify them after you deploy the vApp.

LeaseSettingsSection

Defines the terms of storage and deployment leases for the vApp. If this section is omitted, the vApp inherits the default lease settings of the containing organization.

NetworkConfigSection

Defines the properties of the vApp network and specifies how it is connected to a network in the vDC. Unless you intend to create a vApp that has no connection to any network, you must include this section in the `InstantiationParams` element of an `instantiateVappTemplate` request.

StartupSection

Defines the order in which the virtual machines in the vApp start up and shut down. If this section is omitted, the startup and shutdown order of virtual machines in the vApp is indeterminate.

Configuring a Virtual Machine

You can include the following sections in the `InstantiationParams` for a `Vm`. You can also modify them after you deploy the `Vm`. You can reconfigure a virtual machine by making changes to any of the following sections of a `Vm`:

VirtualHardwareSection

Contains a description of the virtual hardware supported by a virtual machine. You can reconfigure individual items in this section, or reconfigure groups of related items such as hard disks or network interface cards.

OperatingSystemSection

Specifies the guest operating system installed on the virtual machine.

NetworkConnectionSection

Specifies how the virtual NIC devices on the virtual machine are connected to the vApp network.

GuestCustomizationSection

Contains guest customization parameters for the virtual machine.

Reconfiguration Workflow

The workflow for reconfiguring a vApp or virtual machine is the same regardless of the section you are modifying.

1 Retrieve the `vApp` or `Vm` and examine the response to find the section that you want to modify.

2 Retrieve the section by making a GET request to the URL in the section's `href` attribute value.

3 Modify the section as needed.

4 Update the section by making a PUT request to the section's edit link, a `Link` element in the section where `rel="edit"`, and supplying the modified section in the request body.

Modified sections must contain all required elements, even if you are not changing their values. Because optional elements revert to default values if they are omitted or empty, it is a best practice to include optional elements in updates. `Link` elements and `href` attributes from responses do not need to be included in modified sections. Some elements and attributes might be read-only. See the schema reference for details.

Note You cannot make configuration changes to a vApp if it is in maintenance mode. A system administrator can put a vApp into maintenance mode to prevent metadata changes during administrative operations such as backup, restore, and upgrade. See "Summary of vSphere Platform Extension Requests," on page 161.

Retrieve the Configuration Links for a vApp

Each modifiable section of a vApp includes a `Link` element whose `rel` attribute has the value `edit`. You cannot modify sections that do not contain this `Link` element.

Any `ovf:SectionType` element can include an arbitrary number of `Link` elements. Sections that you can modify include a `Link` element where `rel="edit"`. To modify one of these sections, retrieve it by making a GET request to the URL in section's `href` attribute. Then make a PUT request to the `href` attribute value of `Link` where `rel="edit"` to update the section with your modifications.

Prerequisites

Verify that you are logged in as a system administrator or member of an organization in the cloud.

Procedure

1 Retrieve the XML representation of the vApp.

Use a GET request as shown in "Example: Configuration Links in a vApp," on page 88.

2 Examine the response for `edit` links to modifiable sections.

The response portion of "Example: Configuration Links in a vApp," on page 88 includes one of these links for each of the modifiable sections of the vApp. You cannot modify sections that do not contain a `Link` element where `rel="edit"`.

Example: Configuration Links in a vApp

In this example, the response was edited to show only the modifiable sections of the `VApp` element. Each `Vm` in the `Children` element of the `VApp` includes additional configuration links, shown in "Example: Configuration Links in a Vm Element," on page 90.

Request:

```
GET https://vcloud.example.com/api/vApp/vapp-7
```

Response:

```
200 OK
Content-Type: application/vnd.vmware.vcloud.vApp+xml
...
<VApp ... href="https://vcloud.example.com/api/vApp/vapp-7">
   ...
   <LeaseSettingsSection ...
      href="https://vcloud.example.com/api/vApp/vapp-7/leaseSettingsSection/" ...>
      ...
      <Link
         rel="edit"
         type="application/vnd.vmware.vcloud.leaseSettingsSection+xml"
         href="https://vcloud.example.com/api/vApp/vapp-7/leaseSettingsSection/" />
      ...
   </LeaseSettingsSection>
   <ovf:StartupSection ...
      href="https://vcloud.example.com/api/vApp/vapp-7/startupSection/" ... >
      ...
      <Link
         rel="edit"
         type="application/vnd.vmware.vcloud.startupSection+xml"
         href="https://vcloud.example.com/api/vApp/vapp-7/startupSection/" />
   ...
   </ovf:StartupSection>
   <NetworkConfigSection ...
      href="https://vcloud.example.com/api/vApp/vapp-7/networkConfigSection/" ... />
      ...
      <Link
         rel="edit"
         type="application/vnd.vmware.vcloud.networkConfigSection+xml"
         href="https://vcloud.example.com/api/vApp/vapp-7/networkConfigSection/" />
      ...
   </NetworkConfigSection>
   <Children>
      <Vm ...
         status="8"
         name="ubuntu10-x86"
         href="https://vcloud.example.com/api/vApp/vm-4">
         ...
      </Vm>
   </Children>
</VApp>
```

Summary of vApp Reconfiguration Requests

vApp reconfiguration requests retrieve or update modifiable sections of a vApp.

- *API-URL* is a URL of the form `https://vcloud.example.com/api`.
- *id* is a unique identifier in the form of a UUID, as defined by RFC 4122.

Request URLs are always available when you GET the representation of the object on which they operate. URL forms are for reference purposes only. For more information about the requests and responses for each operation, see "About the Schema Reference Downloadable Archive," on page 19.

Table 5-2. Summary of vApp Reconfiguration Requests

Operation	Request	Request Body	Response
Retrieve vApp `LeaseSettingsSection`	GET *API-URL*/vApp/vapp-*id*/ leaseSettingsSection/	None	`LeaseSettingsSection`
Update vApp `LeaseSettingsSection`	PUT *API-URL*/vApp/vapp-*id*/ leaseSettingsSection/	`LeaseSettingsSection`	`Task`
Retrieve vApp `StartupSection`	GET *API-URL*/vApp/vapp-*id*/ startupSection/	None	`StartupSection`
Update vApp `StartupSection`	PUT *API-URL*/vApp/vapp-*id*/ startupSection/	`StartupSection`	`Task`

Table 5-2. Summary of vApp Reconfiguration Requests (Continued)

Operation	Request	Request Body	Response
Retrieve vApp NetworkConfigSection	GET *API-URL*/vApp/vapp-*id*/ networkConfigSection/	None	NetworkConfigSection
Update vApp NetworkConfigSection	PUT *API-URL*/vApp/vapp-*id*/ networkConfigSection/	NetworkConfigSection	Task

Retrieve the Configuration Links for a Virtual Machine

A virtual machine is represented by a Vm element. Each modifiable section of a Vm element includes a Link element whose rel attribute has the value edit. You cannot modify sections that do not contain this Link element.

Any ovf:SectionType element can include an arbitrary number of Link elements. Sections that you can modify include a Link element where rel="edit". To modify one of these sections, retrieve it by making a GET request to the URL in section's href attribute. Then make a PUT request to the href attribute value of Link where rel="edit" to update the section with your modifications.

Prerequisites

Verify that you are logged in as a system administrator or member of an organization in the cloud.

Procedure

1 Retrieve the XML representation of the vApp that contains the virtual machine to reconfigure.

Use a GET request as shown in "Example: Configuration Links in a vApp," on page 88.

2 In the VApp element's Children element, find the Vm element that represents the virtual machine and retrieve it.

3 Examine the response for edit links to modifiable sections.

The response portion of "Example: Configuration Links in a Vm Element," on page 90 shows the links for each of the modifiable section of the Vm. You cannot modify sections that do not contain a link where rel="edit".

Example: Configuration Links in a Vm Element

This example retrieves a Vm element shown in "Example: Configuration Links in a vApp," on page 88. It expands that element to show its configuration links. It also shows the entire NetworkConnectionSection of that Vm, and additional information that is referenced in other examples. You cannot modify sections that do not have a Link where rel="edit", so they do not appear in this example. Modifiable sections of the parent vApp are shown in "Example: Configuration Links in a vApp," on page 88.

Request:

```
GET https://vcloud.example.com/api/vApp/vm-4
```

Response:

```
200 OK
Content-Type: application/vnd.vmware.vcloud.vm+xml
<Vm
   xmlns="http://www.vmware.com/vcloud/v1.5"
   xmlns:ovf="http://schemas.dmtf.org/ovf/envelope/1"
   status="8"
   name="ubuntu10-x86"
   href="https://vcloud.example.com/api/vApp/vm-4">
   <ovf:VirtualHardwareSection>
```

```
  <Link
      rel="edit"
      type="application/vnd.vmware.vcloud.virtualHardwareSection+xml"
      href="https://vcloud.example.com/api/vApp/vm-4/virtualHardwareSection/" />
   <Link
      rel="down"
      type="application/vnd.vmware.vcloud.rasdItem+xml"
      href="https://vcloud.example.com/api/vApp/vm-4/virtualHardwareSection/cpu" />
   <Link
      rel="edit"
      type="application/vnd.vmware.vcloud.rasdItem+xml"
      href="https://vcloud.example.com/api/vApp/vm-4/virtualHardwareSection/cpu" />
   <Link
      rel="down"
      type="application/vnd.vmware.vcloud.rasdItem+xml"
      href="https://vcloud.example.com/api/vApp/vm-4/virtualHardwareSection/memory" />
   <Link
      rel="edit"
      type="application/vnd.vmware.vcloud.rasdItem+xml"
      href="https://vcloud.example.com/api/vApp/vm-4/virtualHardwareSection/memory" />
   <Link
      rel="down"
      type="application/vnd.vmware.vcloud.rasdItemsList+xml"
      href="https://vcloud.example.com/api/vApp/vm-4/virtualHardwareSection/disks" />
   <Link
      rel="edit"
      type="application/vnd.vmware.vcloud.rasdItemsList+xml"
      href="https://vcloud.example.com/api/vApp/vm-4/virtualHardwareSection/disks" />
   <Link
      rel="down"
      type="application/vnd.vmware.vcloud.rasdItemsList+xml"
      href="https://vcloud.example.com/api/vApp/vm-4/virtualHardwareSection/media" />
   <Link
      rel="down"
      type="application/vnd.vmware.vcloud.rasdItemsList+xml"
      href="https://vcloud.example.com/api/vApp/vm-4/virtualHardwareSection/networkCards" />
   <Link
      rel="edit"
      type="application/vnd.vmware.vcloud.rasdItemsList+xml"
      href="https://vcloud.example.com/api/vApp/vm-4/virtualHardwareSection/networkCards" />
</ovf:VirtualHardwareSection>
<ovf:OperatingSystemSection
   href="https://vcloud.example.com/api/vApp/vm-4/operatingSystemSection/" ... >
   <Link
      rel="edit"
      type="application/vnd.vmware.vcloud.operatingSystemSection+xml"
      href="https://vcloud.example.com/api/vApp/vm-4/operatingSystemSection/" />
   ...
</ovf:OperatingSystemSection>
<NetworkConnectionSection>
   <Link
      rel="edit"
      type="application/vnd.vmware.vcloud.networkConnectionSection+xml"
      href="https://vcloud.example.com/api/vApp/vm-4/networkConnectionSection/" />
   <ovf:Info>Specifies the available VM network connections</ovf:Info>
```

```
      <PrimaryNetworkConnectionIndex>0</PrimaryNetworkConnectionIndex>
      <NetworkConnection
         network="vAppNetwork">
         <NetworkConnectionIndex>0</NetworkConnectionIndex>
         <IpAddress>10.147.122.134</IpAddress>
         <IsConnected>false</IsConnected>
         <MACAddress>00:50:56:01:01:49</MACAddress>
         <IpAddressAllocationMode>POOL</IpAddressAllocationMode>
      </NetworkConnection>
   </NetworkConnectionSection>
   <GuestCustomizationSection>
      <Link
         rel="edit"
         href="https://vcloud.example.com/api/vApp/vm-4/guestCustomizationSection+xml/">
      </Link>
      ...
   </GuestCustomizationSection>
   ...
</Vm>
```

Summary of Vm Reconfiguration Requests

Vm reconfiguration requests retrieve or update modifiable sections of a virtual machine (`Vm` object)

- *API-URL* is a URL of the form `https://vcloud.example.com/api`.
- *id* is a unique identifier in the form of a UUID, as defined by RFC 4122.

Request URLs are always available when you GET the representation of the object on which they operate. URL forms are for reference purposes only. For more information about the requests and responses for each operation, see "About the Schema Reference Downloadable Archive," on page 19.

Table 5-3. Summary of Vm Reconfiguration Requests

Operation	Request	Request Body	Response
Retrieve the `NetworkConnectionSection` of a virtual machine	GET *API-URL*/vApp/vm-*id*/networkConnectionSection/	None	`NetworkConnectionSection`
Update the `NetworkConnectionSection` of a virtual machine	PUT *API-URL*/vApp/vm-*id*/networkConnectionSection/	`NetworkConnectionSection`	`Task`
Retrieve the `GuestCustomizationSection` of a virtual machine	GET *API-URL*/vApp/vm-*id*/guestCustomizationSection/	None	`GuestCustomizationSectio`
Update the `GuestCustomizationSection` of a virtual machine	PUT *API-URL*/vApp/vm-*id*/guestCustomizationSection/	`GuestCustomizationSection`	`Task`
Retrieve the `OperatingSystemSection` of a virtual machine	GET *API-URL*/vApp/vm-*id*/operatingSystemSection/	None	`OperatingSystemSection`
Update the `OperatingSystemSection` of a virtual machine	PUT *API-URL*/vApp/vm-*id*/operatingSystemSection/	`OperatingSystemSection`	`Task`
Retrieve the `VirtualHardwareSection` of a virtual machine	GET *API-URL*/vApp/vm-*id*/virtualHardwareSection/	None	`VirtualHardwareSection`

Table 5-3. Summary of Vm Reconfiguration Requests (Continued)

Operation	Request	Request Body	Response
Update the `VirtualHardwareSection` of a virtual machine	PUT *API-URL*/vApp/vm-*id*/ virtualHardwareSection/	`VirtualHardwareSection`	`Task`
Retrieve the CPU configuration of a virtual machine	GET *API-URL*/vApp/vm-*id*/ virtualHardwareSection/ cpu	None	`ovf:Item`
Update the CPU configuration of a virtual machine	PUT *API-URL*/vApp/vm-*id*/ virtualHardwareSection/ cpu	`ovf:Item`	`Task`
Retrieve the `memory` item from the `VirtualHardwareSection` of a virtual machine	GET *API-URL*/vApp/vm-*id*/ virtualHardwareSection/ memory	None	`ovf:Item`
Update the `memory` item in the `VirtualHardwareSection` of a virtual machine	PUT *API-URL*/vApp/vm-*id*/ virtualHardwareSection/ memory	`ovf:Item`	`Task`
Retrieve virtual disk items from the `VirtualHardwareSection` of a virtual machine	GET *API-URL*/vApp/vm-*id*/ virtualHardwareSection/ disks	None	`RasdItemsList`
Update virtual disk items in the `VirtualHardwareSection` of a virtual machine	PUT *API-URL*/vApp/vm-*id*/ virtualHardwareSection/ disks	`RasdItemsList`	`Task`
Retrieve network card items from the `VirtualHardwareSection` of a virtual machine	GET *API-URL*/vApp/vm-*id*/ virtualHardwareSection/ networkCards	None	`RasdItemsList`
Update network card items in the `VirtualHardwareSection` of a virtual machine	PUT *API-URL*/vApp/vm-*id*/ virtualHardwareSection/ networkCards	`RasdItemsList`	`Task`

vCloud API Custom Attributes

vCloud API custom attributes extend several elements in the OVF and RASD namespaces. You can use these attributes to provide additional detail about virtual NIC and hard disk controller devices, or to specify the guest operating system type.

With the exception of `osType`, custom attributes are scoped to `ovf:Item` elements based on the elements' RASD resource type. The `osType` attribute applies to the `ovf:OperatingSystemSection` element. All of the elements to which these custom attributes apply are contained in the `VirtualHardwareSection` of a `Vm`.

Table 5-4. vCloud API Custom Attributes for OVF and RASD Elements

Element Name	RASD Resource Type	Attribute Name	Attribute Type	Description
rasd:Connection	10 (Network adapters)	ipAddressingMode	xs:string	IP addressing mode to use for this connection. One of `NONE`, `MANUAL`, `DHCP`, `POOL`.
rasd:Connection	10 (Network adapters)	ipAddress	xs:string	If `ipAddressingMode="MANUAL"`, set the IP address here

Table 5-4. vCloud API Custom Attributes for OVF and RASD Elements (Continued)

Element Name	RASD Resource Type	Attribute Name	Attribute Type	Description
rasd:Connection	10 (Network adapters)	primaryNetworkConnection	xs:boolean	True if this is the primary network connection of the virtual machine
rasd:HostResource	17 (Hard disks)	capacity	xs:string	Hard disk capacity in megabytes. See "Retrieve or Modify the Hard Disk Configuration of a Virtual Machine," on page 106
rasd:HostResource	17 (Hard disks)	busType	xs:string	read-only
rasd:HostResource	17 (Hard disks)	busSubType	xs:string	read-only
ovf:OperatingSystemSection	N/A	osType	xs:string	Internal VMware identifier for the guest operating system. See https://www.vmware.com/support/developer/vc-sdk/visdk41pubs/ApiReference/vim.vm.GuestOsDescriptor.GuestOsIdentifier.html

For more information about OVF and RASD (`CIM_ResourceAllocationSettingData`) elements, see the OVF specification, available at http://www.dmtf.org/standards/published_documents/DSP0243_1.0.0.pdf.

Retrieve or Update a Modifiable Section

You can make a GET request to the URL of any modifiable section to retrieve it for modification. After you modify the section, you can make a PUT request to its `edit` link to update the section with your modifications.

Prerequisites

Verify that you are logged in as a system administrator or the object owner.

Procedure

1. Retrieve the section to modify.

   Make a GET request to the URL in the section's `href` attribute value.

2. Modify the retrieved section.

   Modified sections must contain all required elements, even if you are not changing their values. Because optional elements revert to default values if they are omitted or empty, it is a best practice to include optional elements in updates. `Link` elements and `href` attributes from responses do not need to be included in modified sections. Some elements and attributes might be read-only. See the schema reference for details.

3. Update the section with your modifications.

   Find the `Link` element in the section where `rel="edit"`. Make a PUT request to the URL in that link's `href` attribute value, and supply the modified section as the request body.

   For most section types, the response to this request is a `Task` element that tracks the update operation. When the task completes, the section is updated.

The modified section replaces the contents of the original section. For some section types, modifications take effect immediately. For others, modifications take effect only after a power or deployment state change.

Example: Retrieve a NetworkConfigSection

This example retrieves the `NetworkConfigSection` of the vApp shown in "Example: Configuration Links in a vApp," on page 88.

Request:

```
GET https://vcloud.example.com/api/vApp/vapp-7/networkConfigSection
```

Response:

```
200 OK
Content-type: application/vnd.vmware.vcloud.networkConfigSection+xml
...
<NetworkConfigSection
   xmlns="http://www.vmware.com/vcloud/v1.5"
   xmlns:ovf="http://schemas.dmtf.org/ovf/envelope/1"
   href="https://vcloud.example.com/api/vApp/vapp-7/networkConfigSection/"
   ovf:required="false">
   <ovf:Info>Configuration parameters for logical networks</ovf:Info>
   <Link
      rel="edit"
      type="application/vnd.vmware.vcloud.networkConfigSection+xml"
      href="https://vcloud.example.com/api/vApp/vapp-7/networkConfigSection/" />
   <NetworkConfig
      networkName="vAppNetwork">
      <Configuration>
         <IpScope>
            <IsInherited>true</IsInherited>
            <Gateway>10.147.56.253</Gateway>
            <Netmask>255.255.255.0</Netmask>
            <Dns1>10.147.115.1</Dns1>
            <Dns2>10.147.115.2</Dns2>
            <DnsSuffix>example.com</DnsSuffix>
            <IpRanges>
               <IpRange>
                  <StartAddress>10.147.56.1</StartAddress>
                  <EndAddress>10.147.56.255</EndAddress>
               </IpRange>
            </IpRanges>
         </IpScope>
         <ParentNetwork
            type="application/vnd.vmware.vcloud.network+xml"
            name="Internet"
            href="https://vcloud.example.com/api/network/54" />
         <FenceMode>bridged</FenceMode>
      </Configuration>
      <IsDeployed>false</IsDeployed>
   </NetworkConfig>
</NetworkConfigSection>
```

For an example that updates this section, see "Example: Update a NetworkConfigSection," on page 96 .

Update a vApp Network Configuration

To change the configuration of a vApp network, you retrieve the NetworkConfigSection element of the vApp, modify it, and use it with a PUT request to update the section.

Prerequisites

Verify that you are logged in as a system administrator or the object owner.

Procedure

1 Retrieve the vApp's NetworkConfigSection.

2 Modify the returned NetworkConfigSection as needed.

Modified sections must contain all required elements, even if you are not changing their values. Because optional elements revert to default values if they are omitted or empty, it is a best practice to include optional elements in updates. Link elements and href attributes from responses do not need to be included in modified sections. Some elements and attributes might be read-only. See the schema reference for details.

3 Update the NetworkConfigSection in the vApp.

Find the Link element in the section where rel="edit". Make a PUT request to the URL in that link's href attribute value, and supply the modified section as the request body.

Example: Update a NetworkConfigSection

This example modifies the NetworkConfigSection that was retrieved in "Example: Retrieve a NetworkConfigSection," on page 95. The modifications change the FenceMode value to natRouted and add a Features element that defines several network features that are useful to an FTP server that must be reachable from the public Internet, but only at the FTP and SSH ports. The modifications add the following items:

- A set of FirewallRules that allow TCP traffic to ports 21 and 22. Because these rules require you to specify a single IP address on the inside of the firewall, the IpScope element is modified to limit the range of IP addresses available on the vApp network to a single address. Any virtual machine that connects to the vApp network defined in this NetworkConfigSection is given this address.
- A NatService element that maps a routable external IP address to the internal IP address allocated to the Vm by the vApp network. The VAppScopedVmId value in this element is taken from the VAppScopedLocalId element of the Vm and the VmNicId value is taken from its PrimaryNetworkConnectionIndex. See "Example: Configuration Links in a Vm Element," on page 90.

This request, like all request bodies derived from a response, omits the Link elements and href attributes that were part of the retrieved NetworkConfigurationSection. It also omits the IsDeployed element of the NetworkConfig. These elements and attributes are created by the server and are read-only. They are ignored if you include them in a request. Read-only elements are noted in the schema references.

Request:

```
PUT https://vcloud.example.com/api/vApp/vapp-7/networkConfigSection/
Content-type: application/vnd.vmware.vcloud.networkConfigSection+xml
...
<?xml version="1.0" encoding="UTF-8"?>
<NetworkConfigSection
   xmlns="http://www.vmware.com/vcloud/v1.5"
   xmlns:ovf="http://schemas.dmtf.org/ovf/envelope/1">
   <ovf:Info>Configuration parameters for logical networks</ovf:Info>
   <NetworkConfig
      networkName="vAppNetwork">
```

```
<Configuration>
   <IpScope>
      <IsInherited>false</IsInherited>
      <Gateway>10.147.56.253</Gateway>
      <Netmask>255.255.255.0</Netmask>
      <Dns1>10.147.115.1</Dns1>
      <Dns2>10.147.115.2</Dns2>
      <DnsSuffix>example.com</DnsSuffix>
      <IpRanges>
         <IpRange>
            <StartAddress>10.147.56.1</StartAddress>
            <EndAddress>10.147.56.1</EndAddress>
         </IpRange>
      </IpRanges>
   </IpScope>
   <ParentNetwork
      type="application/vnd.vmware.vcloud.network+xml"
      name="Internet"
      href="https://vcloud.example.com/api/network/54" />
   <FenceMode>natRouted</FenceMode>
   <Features>
      <FirewallService>
         <IsEnabled>true</IsEnabled>
         <FirewallRule>
            <IsEnabled>true</IsEnabled>
            <Description>FTP Rule</Description>
            <Policy>allow</Policy>
            <Protocols>
               <Tcp>true</Tcp>
            </Protocols>
            <Port>21</Port>
            <DestinationIp>10.147.115.1</DestinationIp>
            <SourcePort>-1</SourcePort>
            <SourceIp>Any</SourceIp>
         </FirewallRule>
         <FirewallRule>
            <IsEnabled>true</IsEnabled>
            <Description>SSH Rule</Description>
            <Policy>allow</Policy>
            <Protocols>
               <Tcp>true</Tcp>
            </Protocols>
            <Port>22</Port>
            <DestinationIp>10.147.115.1</DestinationIp>
            <SourcePort>-1</SourcePort>
            <SourceIp>Any</SourceIp>
         </FirewallRule>
      </FirewallService>
      <NatService>
         <IsEnabled>true</IsEnabled>
         <NatType>ipTranslation</NatType>
         <Policy>allowTraffic</Policy>
         <NatRule>
            <OneToOneVmRule>
               <MappingMode>manual</MappingMode>
```

```
                     <ExternalIpAddress>64.100.10.1</ExternalIpAddress>
                     <VAppScopedVmId>20ea086f-1a6a-4fb2-8e2e-23372facf7de
                     </VAppScopedVmId>
                     <VmNicId>0</VmNicId>
                  </OneToOneVmRule>
               </NatRule>
            </NatService>
         </Features>
      </Configuration>
   </NetworkConfig>
</NetworkConfigSection>
```

Response:

```
202 Accepted
Content-Type: application/vnd.vmware.vcloud.task+xml
...
<Task ... operation="Updating Virtual Application Linux FTP server (7)" ...>
   ...
</Task>
```

IMPORTANT Whenever you modify a vApp network, as we do in this example, you must be sure that the modifications are consistent with the network connection requirements of the virtual machines in the vApp. The vApp in this example contains a single virtual machine. Its `NetworkConnection` element, shown in "Example: Configuration Links in a Vm Element," on page 90, specifies an IP address that will not be available after the vApp network is reconfigured as shown here. "Example: Update a NetworkConnectionSection," on page 99 corrects this problem. This example uses the `IpScope` element to restrict the IP addresses available on a vApp network. It is usually more practical to use a wide range of addresses available on a vApp network and apply any firewall-related IP address restrictions by modifying the `NetworkConnectionSection` of the `Vm` to which the `FirewallRules` apply, as shown in "Example: Update a NetworkConnectionSection," on page 99. A wider range of IP addresses allows you to modify this vApp to include additional virtual machines, and the IP address restriction applied in "Example: Update a NetworkConnectionSection," on page 99 allows the `FirewallRules` in this example to remain valid.

Update the NetworkConnectionSection of a Virtual Machine

Whenever you create a vApp network or update its configuration, you might also need to update the `NetworkConnectionSection` elements of the virtual machines in the vApp.

Prerequisites

Verify that you are logged in as a system administrator or the object owner.

Procedure

1 Retrieve the virtual machine's `NetworkConnectionSection`.

2 Modify the returned `NetworkConnectionSection` as needed.

   Modified sections must contain all required elements, even if you are not changing their values. Because optional elements revert to default values if they are omitted or empty, it is a best practice to include optional elements in updates. `Link` elements and `href` attributes from responses do not need to be included in modified sections. Some elements and attributes might be read-only. See the schema reference for details.

3 Update the `NetworkConnectionSection` in the virtual machine.

   Find the `Link` element in the section where `rel="edit"`. Make a PUT request to the URL in that link's `href` attribute value, and supply the modified section as the request body.

Example: Update a NetworkConnectionSection

This example modifies the `NetworkConnectionSection` shown in "Example: Configuration Links in a Vm Element," on page 90 so that this network connection is compatible with the reconfigured vApp network to which it must connect. See "Example: Update a NetworkConfigSection," on page 96. The modified `NetworkConnectionSection` in the request changes two values:

- The `IpAddress` now specifies the address to which the vApp network's firewall allows access.
- Because it specifies an IP address, the modified `NetworkConnectionSection` also changes the value of the `IpAddressAllocationMode` from `DHCP` to `STATIC`.

NOTE The `ovf:Info` element is a required member of `NetworkConnectionSection` and all other sections that are derived from `ovf:SectionType`. Element content is ignored, but the element itself must be present. In this example, we use the content to explain why the connection is configured this way.

Request:

```
PUT "https://vcloud.example.com/api/vApp/vm-4/networkConnectionSection/
Content-type: application/vnd.vmware.vcloud.networkConnectionSection+xml
...
<?xml version="1.0" encoding="UTF-8"?>
<NetworkConnectionSection
   type="application/vnd.vmware.vcloud.networkConnectionSection+xml"
   xmlns="http://www.vmware.com/vcloud/v1.5"
   xmlns:ovf="http://schemas.dmtf.org/ovf/envelope/1">
   <ovf:Info>Firewall allows access to this address.</ovf:Info>
   <PrimaryNetworkConnectionIndex>0</PrimaryNetworkConnectionIndex>
   <NetworkConnection
      network="vAppNetwork">
      <NetworkConnectionIndex>0</NetworkConnectionIndex>
      <IpAddress>10.147.115.1</IpAddress>
      <IsConnected>true</IsConnected>
      <MACAddress>00:50:56:01:01:49</MACAddress>
      <IpAddressAllocationMode>STATIC</IpAddressAllocationMode>
   </NetworkConnection>
</NetworkConnectionSection>
```

Response:

```
202 Accepted
Content-Type: application/vnd.vmware.vcloud.task+xml
...
<Task ... operation="Updating Virtual Application Linux FTP server (7)" ...>
   ...
</Task>
```

Retrieve or Modify the CPU Configuration of a Virtual Machine

The CPU configuration of a virtual machine is represented by an `Item` in its `VirtualHardwareSection` element.

Prerequisites

Verify that you are logged in as a system administrator or the object owner.

Procedure

1 Retrieve the CPU section to modify.

Make a GET request to the URL in the section's href attribute value:

```
GET https://vcloud.example.com/api/vApp/vm-4/virtualHardwareSection/cpu
```

2 Modify the retrieved section.

Modified sections must contain all required elements, even if you are not changing their values. Because optional elements revert to default values if they are omitted or empty, it is a best practice to include optional elements in updates. Link elements and href attributes from responses do not need to be included in modified sections. Some elements and attributes might be read-only. See the schema reference for details.

3 Update the section with your modifications.

Find the Link element in the section where rel="edit". Make a PUT request to the URL in that link's href attribute value, and supply the modified section as the request body.

The response to this request is a Task element that tracks the update operation. When the task is complete, the section is updated.

The modified section replaces the contents of the original section. For some section types, modifications take effect immediately. For others, modifications take effect only after a power or deployment state change.

Example: Modify the CPU Configuration of a Virtual Machine

This request modifies the CPU section of the virtual machines shown in "Example: Configuration Links in a Vm Element," on page 90. The modified Item in the request body adds a second CPU to the Vm by changing the rasd:VirtualQuantity value of the Item to 2.

Request:

```
PUT https://vcloud.example.com/api/vApp/vm-4/virtualHardwareSection/cpu
Content-type: application/vnd.vmware.vcloud.rasdItem+xml
...
<?xml version="1.0" encoding="UTF-8"?>
<Item xmlns="http://schemas.dmtf.org/ovf/envelope/1"
   xmlns:vcloud="http://www.vmware.com/vcloud/v1.5"
   xmlns:rasd="http://schemas.dmtf.org/wbem/wscim/1/cim-
schema/2/CIM_ResourceAllocationSettingData"
   vcloud:type="application/vnd.vmware.vcloud.rasdItem+xml">
   <rasd:AllocationUnits>hertz * 10^6</rasd:AllocationUnits>
   <rasd:Description>Number of Virtual CPUs</rasd:Description>
   <rasd:ElementName>2 virtual CPU(s)</rasd:ElementName>
   <rasd:InstanceID>4</rasd:InstanceID>
   <rasd:Reservation>0</rasd:Reservation>
   <rasd:ResourceType>3</rasd:ResourceType>
   <rasd:VirtualQuantity>2</rasd:VirtualQuantity>
   <rasd:Weight>0</rasd:Weight>
</Item>
```

Response:

```
202 Accepted
Content-Type: application/vnd.vmware.vcloud.task+xml
...
<Task ... operation="Updating Virtual Application Linux FTP server (7)" ...>
...
</Task>
```

Retrieve or Modify the GuestCustomizationSection of a Virtual Machine

The `GuestCustomizationSection` element includes a customization script and other parameters that are applied when you customize a virtual machine.

The `GuestCustomizationSection` includes predefined property names that VMware guest customization tools recognize. In addition, the vCloud API supports use of the `ovf:ProductSection` to pass an arbitrary set of *key*=*value* pairs to a vApp or virtual machine through the `ovf:Environment` element. See "Retrieve or Modify ProductSection Elements," on page 102.

Prerequisites

Verify that you are logged in as a system administrator or the object owner.

Procedure

1 Retrieve the `GuestCustomizationSection` to modify.

Make a GET request to the URL in the section's `href` attribute value.

```
GET https://vcloud.example.com/api/vApp/vm-12/guestCustomizationSection/
```

2 Modify the retrieved section.

Modified sections must contain all required elements, even if you are not changing their values. Because optional elements revert to default values if they are omitted or empty, it is a best practice to include optional elements in updates. `Link` elements and `href` attributes from responses do not need to be included in modified sections. Some elements and attributes might be read-only. See the schema reference for details.

3 Update the section with your modifications.

In the section, find the `Link` element where `rel="edit"`. Make a PUT request to the URL in that link's `href` attribute value, and supply the modified section as the request body.

The response to this request is a `Task` element that tracks the update operation. When the task completes, the section is updated.

The modified section replaces the contents of the original section. For some section types, modifications take effect immediately. For others, modifications take effect only after a power or deployment state change.

Example: Modify the Guest Customization Section of a Virtual Machine

Request:

```
PUT https://vcloud.example.com/api/vApp/vm-12/guestCustomizationSection/
Content-type: application/vnd.vmware.vcloud.guestcustomizationsection+xml
...
<?xml version="1.0" encoding="UTF-8"?>
<GuestCustomizationSection
   xmlns="http://www.vmware.com/vcloud/v1.5"
   xmlns:ovf="http://schemas.dmtf.org/ovf/envelope/1"
   ovf:required="false">
   <ovf:Info>Specifies Guest OS Customization Settings</ovf:Info>
   <Enabled>true</Enabled>
   <ChangeSid>true</ChangeSid>
   <VirtualMachineId>12</VirtualMachineId>
   <JoinDomainEnabled>false</JoinDomainEnabled>
   <UseOrgSettings>false</UseOrgSettings>
   <DomainName />
   <DomainUserName />
   <DomainUserPassword />
   <AdminPasswordEnabled>true</AdminPasswordEnabled>
   <AdminPasswordAuto>true</AdminPasswordAuto>
   <AdminPassword />
   <ResetPasswordRequired>false</ResetPasswordRequired>
   <CustomizationScript />
   <ComputerName>Win2K3</ComputerName>
</GuestCustomizationSection>
```

Response:

```
202 Accepted
Content-Type: application/vnd.vmware.vcloud.task+xml
...
<Task ... operation="Updating Virtual Application Win2K3 (12)" ...>
...
</Task>
```

Retrieve or Modify ProductSection Elements

`ProductSection` elements allow you to pass runtime information to vApps and virtual machines. The *`key=value`* pairs in this section are made available in the in the OVF `Environment` of a powered-on vApp or virtual machine.

A vApp or virtual machine can get runtime information from its `ovf:Environment` element. This read-only element is populated with information from a `ProductSection` element when the vApp or virtual machine is powered on. A `Vm` can use VMware Tools to read these values from its `ovf:Environment`. A `Vm` can also read the values by mounting a special media object. To make a *`key=value`* pair available in the `ovf:Environment`, add it to the appropriate `ProductSection` of a vApp template or powered-off vApp or virtual machine.

NOTE All `ProductSection` elements in a vApp template, vApp, or virtual machine are returned as members of a `ProductSectionList`. You cannot retrieve or update an individual `ProductSection`. You must retrieve the `ProductSectionList` and update the individual `ProductSection` elements it contains.

Prerequisites

Verify that you are logged in as a system administrator or the object owner.

Procedure

1 Retrieve the `ProductSectionList` from the vApp or virtual machine.

Use a request like this one, which targets a vApp.

```
GET https://vcloud.example.com/api/vApp/vapp-123/productSections
```

The response is a `ProductSectionList` element, which contains all the `ProductSection` elements in the vApp, along with a `Link` element that contains the `rel="edit"` URL to use when updating the `ProductSectionList`. If the vApp contains no `ProductSection` elements, the response contains only the `Link` element.

2 Modify the retrieved `ProductSectionList`.

You can modify existing `ProductSection` elements, create new ones, or both. `ProductSection` has no required contents. Unlike updates to other sections, updates to a `ProductSection` merge new and existing values, subject to the following rules:

- `Property` elements that are present in the existing `ProductSection` but not in the update are removed.
- `Property` elements that are present in the update but not in the in the existing `ProductSection` are added to the `ProductSection` if they have a corresponding `Value` element.

- If a Property element that is present in the existing ProductSection has different attributes, qualifiers, or other details in the update, the Property in the update replaces the existing one.
- If a Property element that is present in the existing ProductSection has no Value in the update, the existing Property and Value remain unchanged.

3 Update the section with your modifications.

Find the Link element in the ProductSectionList where rel="edit". Make a PUT request to the URL in that link's href attribute value, and supply the modified ProductSectionList as the request body.

The response to this request is a Task element that tracks the update operation. When the task is complete, the section is updated.

The modified section replaces the contents of the original section, subject to the rules listed in Step 2.

Example: Update a ProductSection in a vApp

This request creates or updates a ProductSectionList that contains a single ProductSection. The ProductSection sets three properties. The response is a Task.

Request:

```
PUT https://vcloud.example.com/api/vApp/vapp-123/productSections
Content-Type: application/vnd.vmware.vcloud.productSections+xml
...
<?xml version="1.0" encoding="UTF-8"?>
<ProductSectionList
   xmlns="http://www.vmware.com/vcloud/v1.5"
   xmlns:ovf="http://schemas.dmtf.org/ovf/envelope/1">
   <ovf:ProductSection
      required="true">
      <ovf:Info>Information about the installed software</ovf:Info>
      <ovf:Property
         ovf:type="string"
         ovf:key="CRM_Database_Host"
         ovf:value="CRM.example.com">
         <ovf:Label>CRM Database Host</ovf:Label>
      </ovf:Property>
      <ovf:Property
         ovf:type="string"
         ovf:key="CRM_Database_Username"
         ovf:value="dbuser">
         <ovf:Label>CRM Database Usernname</ovf:Label>
      </ovf:Property>
      <ovf:Property
         ovf:type="string"
         ovf:key="CRM_Password"
         ovf:value="Pa55w0rd">
         <ovf:Label>CRM Database User Password</ovf:Label>
      </ovf:Property>
   </ovf:ProductSection>
</ProductSectionList>
```

Response:

```
202 Accepted
Content-Type: application/vnd.vmware.vcloud.task+xml
...
<Task ... operation="Updating Virtual Application ..." ...>
...
</Task>
```

After the vApp is powered on, a virtual machine can retrieve the `ovf:Environment` document in the following ways:

- It can use the default OVF `iso` transport type. This makes the environment document available as a file named `ovf-env.xml` on an ISO image that is mounted on the first available CD-ROM device on the virtual machine. You can use any convenient mechanism to read this file.

  ```
  [root@example-vm-RHEL ~] cat /media/cdrom/ovf-env.xml
  <?xml version="1.0" encoding="UTF-8"?>
  <Environment ...>
     ...
     <PropertySection>
        <Property
           oe:key="CRM_Database_Host"
           oe:value="CRM.example.com" />
        <Property
           oe:key="CRM_Database_Username"
           oe:value="dbuser" />
        <Property
           oe:key="CRM_Password"
           oe:value="Pa55w0rd" />
      </PropertySection>
     ...
  </Environment>
  ```

- If the virtual machine has VMware Tools installed, it can use the `vmtoolsd` program, as shown here.

  ```
  [root@example-vm-RHEL ~] /usr/sbin/vmtoolsd --cmd 'info-get guestinfo.ovfEnv'
  <?xml version="1.0" encoding="UTF-8"?>
  <Environment ...>
     ...
     <PropertySection>
        <Property
           oe:key="CRM_Database_Host"
           oe:value="CRM.example.com" />
        <Property
           oe:key="CRM_Database_Username"
           oe:value="dbuser" />
        <Property
           oe:key="CRM_Password"
           oe:value="Pa55w0rd" />
      </PropertySection>
     ...
  </Environment>
  ```

  On Windows, the `vmtoolsd` executable file is typically installed in `C:\Program Files\VMware\VMware Tools\vmtoolsd.exe`

Retrieve or Modify Groups of Related Sections in a Virtual Machine

The vCloud API provides links that you can use to retrieve or update groups of sections that define related hardware items such as disks, media devices, and network cards in a Vm element.

As shown in "Example: Configuration Links in a Vm Element," on page 90, Link elements for disks, media devices, and network cards are grouped at the end of the VirtualHardwareSection. These links have content type application/vnd.vmware.vcloud.rasdItemsList+xml, and reference a RasdItemsList element in the VirtualHardwareSection of a Vm. The vCloud API uses the RasdItemsList element to aggregate related elements in a VirtualHardwareSection. This approach simplifies retrieval and modification of Item elements that are typically viewed or modified as a group.

Prerequisites

Verify that you are logged in as a system administrator or the object owner.

Procedure

1 Retrieve the RasdItemsList from a Vm.

   Make a GET request to the URL in the link where type="application/vnd.vmware.vcloud.rasdItemsList+xml" and rel="down". See "Example: Retrieve the Hard Disks and Controllers in a Virtual Machine," on page 105.

2 Modify the items in the retrieved list.

   Modified sections must contain all required elements, even if you are not changing their values. Because optional elements revert to default values if they are omitted or empty, it is a best practice to include optional elements in updates. Link elements and href attributes from responses do not need to be included in modified sections. Some elements and attributes might be read-only. See the schema reference for details.

3 Update the sections with your modifications.

   Make a PUT request to the URL in the link where type="application/vnd.vmware.vcloud.rasdItemsList+xml" and rel="edit", and supply the modified section as the request body.

   The response to this request is a Task element that tracks the update operation. When the task is complete, the section is updated.

Example: Retrieve the Hard Disks and Controllers in a Virtual Machine

This example uses the virtualHardwareSection/disks link shown in "Example: Configuration Links in a Vm Element," on page 90 to retrieve the list of hard disks and hard disk controllers for a virtual machine.

Request:

```
GET https://vcloud.example.com/api/vApp/vm-4/virtualHardwareSection/disks
```

Response:

```
Content-Type: application/vnd.vmware.vcloud.rasdItemsList+xml
...
<RasdItemsList
   xmlns="http://www.vmware.com/vcloud/v1.5"
   xmlns:rasd="http://schemas.dmtf.org/wbem/wscim/1/cim-
schema/2/CIM_ResourceAllocationSettingData"
   type="application/vnd.vmware.vcloud.rasdItemsList+xml"
   href="https://vcloud.example.com/api/vApp/vm-4/virtualHardwareSection/disks" ... >
   <Link
      rel="edit"
      type="application/vnd.vmware.vcloud.rasdItemsList+xml"
      href="https://vcloud.example.com/api/vApp/vm-4/virtualHardwareSection/disks" />
   <Item>
      <rasd:Address>0</rasd:Address>
      <rasd:Description>SCSI Controller</rasd:Description>
      <rasd:ElementName>SCSI Controller 0</rasd:ElementName>
      <rasd:InstanceID>2</rasd:InstanceID>
      <rasd:ResourceSubType>lsilogic</rasd:ResourceSubType>
      <rasd:ResourceType>6</rasd:ResourceType>
   </Item>
   <Item>
      <rasd:AddressOnParent>0</rasd:AddressOnParent>
      <rasd:Description>Hard disk</rasd:Description>
      <rasd:ElementName>Hard disk 1</rasd:ElementName>
      <rasd:HostResource
         xmlns:vcloud="http://www.vmware.com/vcloud/v1.5"
         vcloud:capacity="1024"
         vcloud:busSubType="lsilogic"
         vcloud:busType="6"></rasd:HostResource>
      <rasd:InstanceID>2000</rasd:InstanceID>
      <rasd:Parent>2</rasd:Parent>
      <rasd:ResourceType>17</rasd:ResourceType>
   </Item>
   <Item>
      <rasd:AddressOnParent>1</rasd:AddressOnParent>
      <rasd:Description>Hard disk</rasd:Description>
      <rasd:ElementName>Hard disk 2</rasd:ElementName>
      <rasd:HostResource
         xmlns:vcloud="http://www.vmware.com/vcloud/v1.5"
         vcloud:capacity="2048"
         vcloud:busSubType="lsilogic"
         vcloud:busType="6"></rasd:HostResource>
      <rasd:InstanceID>2001</rasd:InstanceID>
      <rasd:Parent>2</rasd:Parent>
      <rasd:ResourceType>17</rasd:ResourceType>
   </Item>
   <Item>
      <rasd:Address>0</rasd:Address>
      <rasd:Description>IDE Controller</rasd:Description>
      <rasd:ElementName>IDE Controller 0</rasd:ElementName>
      <rasd:InstanceID>3</rasd:InstanceID>
      <rasd:ResourceType>5</rasd:ResourceType>
   </Item>
</RasdItemsList>
```

Retrieve or Modify the Hard Disk Configuration of a Virtual Machine

The hard disk configuration of a virtual machine is represented by an `Item` element in its `VirtualHardwareSection`.

Prerequisites

Verify that you are logged in as a system administrator or the object owner.

Procedure

1 Retrieve the hard disk configuration from the virtual machine.

   Make a GET request to the virtual machine's `virtualHardwareSection/disks` link.

   ```
   GET https://vcloud.example.com/api/vApp/vm-4/virtualHardwareSection/disks
   ```

   The response to this kind of request is a `RasdItemsList` element that contains an `Item` element for each of the virtual machine's hard disks and hard disk controllers, as shown in "Example: Retrieve the Hard Disks and Controllers in a Virtual Machine," on page 105.

2 Modify the retrieved section.

   Modified sections must contain all required elements, even if you are not changing their values. Because optional elements revert to default values if they are omitted or empty, it is a best practice to include optional elements in updates. `Link` elements and `href` attributes from responses do not need to be included in modified sections. Some elements and attributes might be read-only. See the schema reference for details.

3 Update the section with your modifications.

   Find the `Link` element in the section where `rel="edit"`. Make a PUT request to the URL in that link's `href` attribute value, and supply the modified section as the request body.

   The response to this request is a `Task` element that tracks the update operation. When the task is complete, the section is updated.

The modified section replaces the contents of the original section. For some section types, modifications take effect immediately. For others, modifications take effect only after a power or deployment state change.

Example: Modify the Hard Disk Configuration of a Virtual Machine

The following request increases the capacity of the hard disk from 1GB to 10GB by changing the `vcloud:capacity` value of the `Item` that defines the disk. The capacity is raised from `1024` to `10240`. The request body includes the entire `RasdItemsList` returned by the request shown in Step 1, even though only one element is changed. `Link` elements from a response are ignored if you include them in a request, so they are omitted in this example.

Request:

```
PUT https://vcloud.example.com/api/vApp/vm-4/virtualHardwareSection/disks
Content-Type: application/vnd.vmware.vcloud.rasditemslist+xml
...
<?xml version="1.0" encoding="UTF-8"?>
<RasdItemsList
   xmlns="http://www.vmware.com/vcloud/v1.5"
   xmlns:rasd="http://schemas.dmtf.org/wbem/wscim/1/cim-
schema/2/CIM_ResourceAllocationSettingData"
   type="application/vnd.vmware.vcloud.rasdItemsList+xml" >
   <Item>
      <rasd:Address>0</rasd:Address>
      <rasd:Description>SCSI Controller</rasd:Description>
      <rasd:ElementName>SCSI Controller 0</rasd:ElementName>
      <rasd:InstanceID>2</rasd:InstanceID>
      <rasd:ResourceSubType>lsilogic</rasd:ResourceSubType>
      <rasd:ResourceType>6</rasd:ResourceType>
   </Item>
   <Item>
      <rasd:AddressOnParent>0</rasd:AddressOnParent>
      <rasd:Description>Hard disk</rasd:Description>
```

```
      <rasd:ElementName>Hard disk 1</rasd:ElementName>
      <rasd:HostResource
         xmlns:vcloud="http://www.vmware.com/vcloud/v1.5"
         vcloud:capacity="10240"
         vcloud:busSubType="lsilogic"
         vcloud:busType="6"></rasd:HostResource>
      <rasd:InstanceID>2000</rasd:InstanceID>
      <rasd:Parent>2</rasd:Parent>
      <rasd:ResourceType>17</rasd:ResourceType>
   </Item>
   <Item>
      <rasd:AddressOnParent>1</rasd:AddressOnParent>
      <rasd:Description>Hard disk</rasd:Description>
      <rasd:ElementName>Hard disk 2</rasd:ElementName>
      <rasd:HostResource
         xmlns:vcloud="http://www.vmware.com/vcloud/v1.5"
         vcloud:capacity="2048"
         vcloud:busSubType="lsilogic"
         vcloud:busType="6"></rasd:HostResource>
      <rasd:InstanceID>2001</rasd:InstanceID>
      <rasd:Parent>2</rasd:Parent>
      <rasd:ResourceType>17</rasd:ResourceType>
   </Item>
   <Item>
      <rasd:Address>0</rasd:Address>
      <rasd:Description>IDE Controller</rasd:Description>
      <rasd:ElementName>IDE Controller 0</rasd:ElementName>
      <rasd:InstanceID>3</rasd:InstanceID>
      <rasd:ResourceType>5</rasd:ResourceType>
   </Item>
</RasdItemsList>
```

The response is a task.

Response:

```
202 Accepted
Content-Type: application/vnd.vmware.vcloud.task+xml
...
<Task ... operation="Updating Virtual Application Linux FTP server (7)" ...>
...
</Task>
```

Creating, Provisioning, and Managing Organizations 6

The VMware vCloud API supports objects and operations that an organization or a system administrator can use to automate tasks associated with provisioning organizations with resources and users.

A successful login by an organization or system administrator returns a `Session` element, which contains a link that enables the administrator to retrieve a `VCloud` element. This element provides access to a cloud-wide namespace of administrative objects. See "Retrieve an Administrative View of a Cloud," on page 47.

This chapter includes the following topics:

Summary of Administrative Requests

Administrative requests retrieve or update administrative objects such as organizations, users, and vDCs.

- *API-URL* is a URL of the form `https://vcloud.example.com/api`.
- *id* is a unique identifier in the form of a UUID, as defined by RFC 4122.

Request URLs are always available when you GET the representation of the object on which they operate. URL forms are for reference purposes only. For more information about the requests and responses for each operation, see "About the Schema Reference Downloadable Archive," on page 19.

Table 6-1. Summary of Administrative Requests

Operation	Request	Request Body	Response
Retrieve an administrative view of an organization.	GET *API-URL*/admin/org/*id*	None	`AdminOrg`
Create an organization.	POST *API-URL*/admin/orgs	`AdminOrg`	`AdminOrg`
Enable an organization.	POST *API-URL*/admin/org/*id*/action/enable	None	204 No Content

Table 6-1. Summary of Administrative Requests (Continued)

Operation	Request	Request Body	Response
Disable an organization.	POST *API-URL*/admin/org/*id*/action/disable	None	204 No Content
Update an organization.	PUT *API-URL*/admin/org/*id*	`AdminOrg`	`AdminOrg`
Create a network in an organization.	POST *API-URL*/admin/org/*id*/networks	`OrgNetwork`	`OrgNetwork`
Update a network in an organization.	PUT *API-URL*/admin/network/*id*	`OrgNetwork`	`Task`
Delete a network from an organization.	DELETE *API-URL*/admin/network/*id*	None	`Task`
Synchronize syslog server settings for organization and vApp networks. [NEW]	POST *API-URL*/admin/network/*id*/action/syncSyslogServerSettings	None	`Task`
Reset networking services on a logical network. [NEW]	POST *API-URL*/admin/network/*id*/action/reset	None	`Task`
Retrieve a provider vDC.	GET *API-URL*/admin/providervdc/*id*	None	`ProviderVdc`
List the organization vDCs supported by a provider vDC.	GET *API-URL*/admin/providervdc/*id*/vdcReferences	None	`VdcReferences`
Create a vDC for an organization.	POST *API-URL*/admin/org/*id*/vdcs	`AdminVdc`	`AdminVdc`
Enable a vDC.	POST *API-URL*/admin/vdc/*id*/action/enable	None	204 No Content
Disable a vDC.	POST *API-URL*/admin/vdc/*id*/action/disable	None	204 No Content
Delete a vDC.	DELETE *API-URL*/admin/vdc/*id*	None	`Task`
Create a catalog.	POST *API-URL*/admin/org/*id*/catalogs	`AdminCatalog`	`AdminCatalog`
Publish a catalog.	POST *API-URL*/admin/catalog/*id*/action/publish	`PublishCatalogParams`	204 No Content
Update catalog properties.	PUT *API-URL*/admin/catalog/*id*	`AdminCatalog`	`AdminCatalog`
Delete a catalog.	DELETE *API-URL*/admin/catalog/*id*	None	204 No Content
Update the owner of a catalog. [NEW]	PUT *API-URL*/admin/catalog/*id*/owner	`Owner`	204 No Content
Create or import a user.	POST *API-URL*/admin/org/*id*/users	`User`	`User`
Update user properties.	PUT *API-URL*/admin/user/*id*	`User`	`User`

Table 6-1. Summary of Administrative Requests (Continued)

Operation	Request	Request Body	Response
Delete a user.	DELETE *API-URL*/admin/user/*id*	None	204 No Content
Import a group.	POST *API-URL*/admin/org/*id*/groups	`Group`	`Group`
Delete a group.	DELETE *API-URL*/admin/group/*id*	None	204 No Content
Update group properties.	PUT *API-URL*/admin/group/*id*	`Group`	`Group`
Create a role.	POST *API-URL*/admin/org/*id*/roles	`Role`	`Role`
Update role properties.	PUT *API-URL*/admin/role/*id*	`Role`	`Role`
Delete a role.	DELETE *API-URL*/admin/role/*id*	None	204 No Content
Retrieve a list of tasks owned by organization *id*.	GET *API-URL*/tasksList/*id*	None	`TasksList`
Retrieve a task.	GET *API-URL*/tasks/*id*	None	`Task`
Cancel a task.	POST *API-URL*/task/*id*/action/cancel	None	204 No Content

Administrator Credentials and Privileges

An administrator's privileges are scoped by the organization to which the administrator authenticates.

The vCloud API defines two levels of administrative privilege:

- Organization administrators, who have administrative privileges in a specific organization.
- System administrators, who have superuser privileges throughout the system. System administrators are members of the `System` organization, and can create, read, update, and delete all objects in a cloud. They have organization administrator rights in all organizations in a cloud, and can operate directly on vSphere resources to create and modify provider vDCs.

Some administrative operations, and all vSphere platform operations, are restricted to the system administrator. Before you attempt these operations, log in to the `System` organization with the user name and password of the system administrator account that was created when vCloud Director was installed. For example, if the system administrator's user name and password was defined as `administrator` and `Pa55w0rd`, the system administrator login credentials are the MIME Base64 encoding of the string `administrator@System:Pa55w0rd`.

The System Organization

The `System` organization is created automatically when vCloud Director is installed. It is not listed in an `OrgList`, but can be retrieved by a system administrator can retrieve it with a GET request to the `href` value of a special `rel="down"` link contained in the `VCloud` object, as shown in "Example: The System Organization," on page 112

Example: The System Organization

Every `VCloud` object contains a link, like the one shown here, to the `System` organization.

```
<VCloud ... >
   ...
   <Link
      rel="down"
      type="application/vnd.vmware.admin.systemOrganization+xml"
      name="System"
      href="https://vcloud.example.com/api/admin/org/123"/>
   ...
</VCloud>
```

Unlike `Org` and `AdminOrg` objects, the `System` organization cannot contain catalogs, vDCs, groups, or users who are not system administrators.

Organization Administration

System administrators work with `AdminOrg` elements to create and modify organizations. Organization administrators can modify most properties of these organizations.

A cloud can contain one or more organizations. Each organization is a unit of administration for a collection of users, groups, and computing resources. Users authenticate at the organization level, supplying credentials established when the user was created or imported.

An `AdminOrg` is an extended representation of an `Org` object, and is accessed at *`API-URL`*`/admin/org/`*`id`*, rather than *`API-URL`*`/org/`*`id`*. Although system administrators work with `AdminOrg` elements to create and modify organizations, most modifications to an `AdminOrg` also modify some property of the corresponding `Org` that is visible to users.

Create an Organization

To create an organization, a system administrator POSTs an `AdminOrg` element to the cloud's `add` URL for `orgs`.

When you create an organization, you can specify its properties in detail. You must specify a subset of those properties during creation. Properties that you do not specify in the creation request are initialized with default values that you can modify later.

Prerequisites

Verify that you are logged in as a system administrator.

Procedure

1 Retrieve the XML representation of the cloud.

Use a request like this one.

```
GET https://vcloud.example.com/api/admin
```

2 Examine the response to locate the `Link` element that contains the URL for adding organizations to the cloud.

This element has a `rel` attribute value of `add` and a `type` attribute value of `application/vnd.vmware.admin.organization+xml`, as shown here:

```
<Link href="https://vcloud.example.com/api/admin/orgs"
   rel="add"
   type="application/vnd.vmware.admin.organization+xml"/>
```

3 Create an AdminOrg element that specifies the properties of the organization.

   See the request portion of "Example: Create an Organization," on page 113.

4 POST the AdminOrg element you created in Step 3 to the URL described in Step 2.

   See the request portion of "Example: Create an Organization," on page 113.

The server creates and enables the organization, and returns an AdminOrg element that includes the contents you POSTed, along with a set of Link elements that you can use to access, remove, disable, or modify it. vCloud API users can log in to this organization using the URL specified in the href attribute of the Link where rel="alternate". Users of the vCloud Director Web console can log in to the organization at a URL of the form *cloud-url*/org/*name*, where *cloud-url* is a URL of the form https://vcloud.example.com/cloud and *name* is the value of the name attribute of the AdminOrg element. To log in to the organization created by "Example: Create an Organization," on page 113, a user opens a browser and navigates to https://vcloud.example.com/cloud/org/Finance.

Example: Create an Organization

This request creates an organization and specifies its required properties. For a list of all required and optional elements that an AdminOrg contains, see the schema reference.

Request:

```
POST https://vcloud.example.com/api/admin/orgs
Content-Type: application/vnd.vmware.admin.organization+xml
...
<?xml version="1.0" encoding="UTF-8"?>
<AdminOrg
   xmlns="http://www.vmware.com/vcloud/v1.5"
   name="Finance"
   type="application/vnd.vmware.admin.organization+xml">
   <Description>Example Corporation’s Finance Organization</Description>
   <FullName>Finance</FullName>
   <Settings>
      <GeneralOrgSettings>
         <CanPublishCatalogs>true</CanPublishCatalogs>
      </GeneralOrgSettings>
      <OrgLdapSettings>
         <OrgLdapMode>SYSTEM</OrgLdapMode>
      </OrgLdapSettings>
      <OrgEmailSettings>
         <IsDefaultSmtpServer>true</IsDefaultSmtpServer>
         <IsDefaultOrgEmail>true</IsDefaultOrgEmail>
         <FromEmailAddress>vcloud@example.com</FromEmailAddress>
         <DefaultSubjectPrefix>Attention</DefaultSubjectPrefix>
         <IsAlertEmailToAllAdmins>true</IsAlertEmailToAllAdmins>
      </OrgEmailSettings>
   </Settings>
</AdminOrg>
```

The response echoes the request, and includes links that an administrator can use to manage the organization and its settings, and to add resources such as vDCs, catalogs, and users. On creation, AdminOrg objects are disabled by default. A system administrator must enable the AdminOrg before users can log into it. See "Example: Enable an Organization," on page 118

The response also includes elements inherited from system defaults, such as a OrgPasswordPolicySettings, VAppLeaseSettings, and VAppTemplateLeaseSettings. The full content of these elements appears in the actual response. This example shows only place-holders.

Response:

```
201 Created
Content-Type: application/vnd.vmware.admin.organization+xml
...
<AdminOrg
   xmlns="http://www.vmware.com/vcloud/v1.5"
   name="Finance"
   id="urn:vcloud:org:26"
   type="application/vnd.vmware.admin.organization+xml"
   href="https://vcloud.example.com/api/admin/org/26" ...>
   <Link
      rel="add"
      type="application/vnd.vmware.admin.catalog+xml"
      href="https://vcloud.example.com/api/admin/org/26/catalogs" />
   <Link
      rel="add"
      type="application/vnd.vmware.admin.user+xml"
      href="https://vcloud.example.com/api/admin/org/26/users" />
   <Link
      rel="add"
      type="application/vnd.vmware.admin.group+xml"
      href="https://vcloud.example.com/api/admin/org/26/groups" />
   <Link
      rel="add"
      type="application/vnd.vmware.admin.vdc+xml"
      href="https://vcloud.example.com/api/admin/org/26/vdcs" />
   <Link
      rel="add"
      type="application/vnd.vmware.admin.orgNetwork+xml"
      href="https://vcloud.example.com/api/admin/org/26/networks" />
   <Link
      rel="edit"
      type="application/vnd.vmware.admin.organization+xml"
      href="https://vcloud.example.com/api/admin/org/26" />
   <Link
      rel="remove"
      href="https://vcloud.example.com/api/admin/org/26" />
   <Link
      rel="enable"
      href="https://vcloud.example.com/api/admin/org/26/action/enable" />
   <Link
      rel="alternate"
      type="application/vnd.vmware.vcloud.org+xml"
      href="https://vcloud.example.com/api/org/26" />
   <Description>Example Corporation’s Finance Organization</Description>
   <FullName>Finance</FullName>
   <IsEnabled>false</IsEnabled>
   <Settings
      type="application/vnd.vmware.admin.orgSettings+xml"
      href="https://vcloud.example.com/api/admin/org/26/settings">
      <Link
         rel="down"
         type="application/vnd.vmware.admin.vAppTemplateLeaseSettings+xml"
         href="https://vcloud.example.com/api/admin/org/26/settings/vAppTemplateLeaseSettings" />
      <Link
```

```
         rel="down"
         type="application/vnd.vmware.admin.organizationEmailSettings+xml"
         href="https://vcloud.example.com/api/admin/org/26/settings/email" />
      <Link
         rel="down"
         type="application/vnd.vmware.admin.vAppLeaseSettings+xml"
         href="https://vcloud.example.com/api/admin/org/26/settings/vAppLeaseSettings" />
      <Link
         rel="down"
         type="application/vnd.vmware.admin.organizationPasswordPolicySettings+xml"
         href="https://vcloud.example.com/api/admin/org/26/settings/passwordPolicy" />
      <Link
         rel="down"
         type="application/vnd.vmware.admin.organizationGeneralSettings+xml"
         href="https://vcloud.example.com/api/admin/org/26/settings/general" />
      <Link
         rel="down"
         type="application/vnd.vmware.admin.organizationLdapSettings+xml"
         href="https://vcloud.example.com/api/admin/org/26/settings/general/ldap" />
      <Link
         rel="edit"
         type="application/vnd.vmware.admin.orgSettings+xml"
         href="https://vcloud.example.com/api/admin/org/26/settings" />
      <GeneralOrgSettings
         type="application/vnd.vmware.admin.organizationGeneralSettings+xml"
         href="https://vcloud.example.com/api/admin/org/26/settings/general">
         <Link
            rel="edit"
            type="application/vnd.vmware.admin.organizationGeneralSettings+xml"
            href="https://vcloud.example.com/api/admin/org/26/settings/general" />
         <CanPublishCatalogs>false</CanPublishCatalogs>
         <DeployedVMQuota>0</DeployedVMQuota>
         <StoredVmQuota>0</StoredVmQuota>
         <UseServerBootSequence>false</UseServerBootSequence>
         <DelayAfterPowerOnSeconds>0</DelayAfterPowerOnSeconds>
      </GeneralOrgSettings>
      <VAppLeaseSettings
         ...
      </VAppLeaseSettings>
      <VAppTemplateLeaseSettings
         ...
      </VAppTemplateLeaseSettings>
      <OrgLdapSettings
         ...
      </OrgLdapSettings>
      <OrgEmailSettings
         ...
      </OrgEmailSettings>
      <OrgPasswordPolicySettings
         ...
      </OrgPasswordPolicySettings>
   </Settings>
</AdminOrg>
```

Retrieve or Update Organization Settings

A system or organization administrator can retrieve or update organization properties. These properties establish default values for leases on vApps and vApp templates, email service and message properties, and similar settings.

An `AdminOrg` element contains an `OrgSettings` element, which contains the following elements, each of which represents a group of related organization settings.

GeneralOrgSettings	Sets limits on stored and deployed virtual machines, specifies default behaviors for virtual machines in the organization, and controls the ability to publish catalogs
VAppLeaseSettings	Controls storage and deployment leases for vApps.
VAppTemplateLeaseSettings	Controls storage and deployment leases for vApp templates.
OrgLdapSettings	Defines whether this organization is connected to an LDAP service, and whether it uses the system default LDAP service or a custom LDAP service. Specifies configuration and connection parameters for a custom LDAP service if needed.
OrgEmailSettings	Defines whether this organization uses the system default email service, and specifies properties of email alerts sent by the organization. Specifies configuration and connection parameters for a custom email service if needed.
OrgPasswordPolicySettings	Specifies policies to be followed when a user in this organization enters an incorrect password.

Prerequisites

Verify that you are logged in as an organization administrator or system administrator.

Procedure

1 Retrieve the list of organization settings elements.

Use a request like this one:

```
GET https://vcloud.example.com/api/admin/org/26/settings
```

The response is an `OrgSettings` element.

2 Examine the `OrgSettings` element to find the links to the sections to view or modify.

Each section is represented in the `OrgSettings` element with a link where `rel="down"`. You can use that link to retrieve the section. The retrieved section includes a link where `rel="edit"`. You can use that link as the target of a PUT request that modifies the settings that the element represents. The `OrgSettings` element itself also has a `rel="edit"` link, which you can use to update multiple settings sections in one request.

3 Retrieve the settings element to modify.

Make a GET request to the URL in the element's `href` attribute value.

4 Modify the retrieved settings element.

Modified sections must contain all required elements, even if you are not changing their values. Because optional elements revert to default values if they are omitted or empty, it is a best practice to include optional elements in updates. `Link` elements and `href` attributes from responses do not need to be included in modified sections. Some elements and attributes might be read-only. See the schema reference for details.

5 Update the settings with your modifications.

Find the Link element in the settings element where rel="edit". Make a PUT request to the URL in that link's href attribute value, and supply the modified section as the request body. See the request portion of "Example: Update Organization General Settings," on page 117.

Example: Update Organization General Settings

This example updates the GeneralOrgSettings of the organization created in "Example: Create an Organization," on page 113. The update changes the limits on deployed and stored virtual machines. The request includes all members of the GeneralOrgSettings element, even those that are not changing. It is a best practice to include all members of the GeneralOrgSettings element, because optional elements that are missing or empty in the request are reset to their default values.

Request:

```
PUT https://vcloud.example.com/api/admin/org/26/settings/general
Content-Type: application/vnd.vmware.admin.organizationGeneralSettings+xml
...
<?xml version="1.0" encoding="UTF-8"?>
<GeneralOrgSettings
   type="application/vnd.vmware.admin.organizationGeneralSettings+xml">
   <CanPublishCatalogs>false</CanPublishCatalogs>
   <DeployedVMQuota>10</DeployedVMQuota>
   <StoredVmQuota>100</StoredVmQuota>
   <UseServerBootSequence>false</UseServerBootSequence>
   <DelayAfterPowerOnSeconds>0</DelayAfterPowerOnSeconds>
</GeneralOrgSettings>
```

The response echoes the request, and includes a rel="edit" link and other attributes that the server creates.

Response:

```
200 OK
Content-Type: application/vnd.vmware.admin.organizationGeneralSettings+xml
...
<GeneralOrgSettings
   type="application/vnd.vmware.admin.organizationGeneralSettings+xml"
   href="https://vcloud.example.com/api/admin/org/26/settings/general">
   <Link
      rel="edit"
      type="application/vnd.vmware.admin.organizationGeneralSettings+xml"
      href="https://vcloud.example.com/api/admin/org/26/settings/general" />
   <CanPublishCatalogs>false</CanPublishCatalogs>
   <DeployedVMQuota>10</DeployedVMQuota>
   <StoredVmQuota>100</StoredVmQuota>
   <UseServerBootSequence>false</UseServerBootSequence>
   <DelayAfterPowerOnSeconds>0</DelayAfterPowerOnSeconds>
</GeneralOrgSettings>
```

Enable, Disable, or Remove an Organization

An AdminOrg element includes action links that a system administrator can use to enable, disable, or remove the organization.

Prerequisites

- Verify that you are logged in as a system administrator.

- Retrieve the XML representation of the organization. See "Retrieve a List of Organizations Accessible to You," on page 46.

Procedure

- To enable an organization, POST a request to its `action/enable` link.
- To disable an organization, POST a request to its `action/disable` link.
- To remove an organization:

  a POST a request to the `action/disable` link to disable the organization.

  After the organization is disabled, its representation includes a `rel="remove"` link.

  b Delete or change ownership of all objects that the organization's users own.

  c Make a DELETE request to the organization's `rel="remove"` link.

The server takes the requested action and returns an HTTP status of `204 No Content`.

Example: Enable an Organization

This example enables the organization created in "Example: Create an Organization," on page 113

Request:

```
POST https://vcloud.example.com/api/admin/org/26/action/enable
```

Response:

```
204 No Content
```

Network Administration

A newly created organization has no networks in it. An organization administrator must create organization networks from resources such as external networks and network pools that a system administrator provisions.

An organization can be provisioned with one or more networks. These organization networks can be configured to provide services such as DHCP, NAT, VPN, and firewalls.

About vCloud Director Networks

Simple vCloud API networks can rely on default values for many configuration parameters. vCloud API networks that have advanced features require a more detailed specification.

There are three categories of vCloud networks: external networks, organization networks, and vApp networks. There are also network pools which, while not networks themselves, provide network resources to certain types of organization and vApp networks. You can configure organization and vApp networks to allow direct or routed connections to other networks. You can also configure them to be isolated, with no connectivity to other networks.

vApp Networks

A vApp network is a logical network that controls how the virtual machines in a vApp connect to each other and to organization networks. Users specify vApp network details in an `instantiateVAppTemplate` or `composeVApp` request. The network is created when the vApp is deployed, and deleted when the vApp is undeployed. All nonisolated virtual machines in the vApp connect to a vApp network, as specified in their `NetworkConnectionSection` elements.

Every VApp element includes a link like this one, which you can use to retrieve details of its vApp networks.

```
<Link
      rel="down"
      type="application/vnd.vmware.vcloud.vAppNetwork+xml"
      name="isoNet1"
      href="https://vcloud.example.com/api/network/94 />
```

A GET request to this link returns a read-only VAppNetwork element. To modify an existing vApp network, you must find the NetworkConfigSection of the VApp element and use its edit link, as shown in "Update a vApp Network Configuration," on page 96

You specify the configuration of a vApp network in the NetworkConfig element of the InstantiateVAppTemplateParams request body, which includes the following information:

- A name for the network, specified in the networkName attribute of the NetworkConfig element. The instantiation parameters must create a vApp network whose name matches the value of the network attribute of the NetworkConnection of each Vm element in the template. If this attribute has the value none or is missing, the Vm can connect to any network. If the template contains Vm elements that specify different names for their network connections, you must create a vApp network for each.

  NOTE When you create a vApp network where the FenceMode is bridged, the networkName of the vApp network must match the name of the ParentNetwork. This requirement is enforced by the composeVapp operation. The instantiateVappTemplate operation automatically corrects a name mismatch by changing the value of the network attribute in the NetworkConnection element of the VApp.

- A Configuration element that specifies network configuration details.
  - For routed and directly connected networks, the ParentNetwork element contains a reference to the organization network that the vApp network connects to. The FenceMode element controls how the two networks connect. Specify a FenceMode of bridged for a direct connection to the parent network, or natRouted to specify a routed connection controlled by network Features such as a NatService or FirewallService. If you want the organization network to be isolated, with no external connection, omit the ParentNetwork element and specify the FenceMode as isolated.
  - The Features element defines features of the vApp network. See "Network Services," on page 121.
  - Additional modifiable elements like IpScope and RetainNetInfoAcrossDeployments, and read-only elements such as SyslogServerSettings and RouterInfo. See the schema reference for more information about the type and scope of these elements.
- Network pool resources required by an isolated or natRouted vApp network are allocated by the system from an unnamed pool owned by the vDC in which the vApp is deployed.

Organization Networks

System administrators create organization networks from resources that external networks and network pools provide. An organization network allows virtual machines in the organization to communicate with each other and to access external networks like the Internet.

Table 6-2. Types of Organization Networks and Their Requirements

Organization Network Type	Description	Requirements
External organization network - direct connection	Accessible by multiple organizations. Virtual machines belonging to different organizations can connect to and see traffic on this network. This network provides direct layer 2 connectivity to machines outside of the organization. Machines outside of this organization can connect to machines within the organization directly.	External network
External organization network - NAT-routed connection	Accessible only by this organization. Only virtual machines within this organization can connect to this network. This network also provides controlled access to an external network. System administrators and organization administrators can configure network address translation (NAT) and firewall settings to make specific virtual machines accessible from the external network.	External network and network pool
Internal organization network	Accessible only by this organization. Only virtual machines within this organization can connect to and see traffic on this network. This network provides an organization with an isolated, private network that multiple vApps can connect to. This network provides no connectivity to machines outside this organization. Machines outside of this organization have no connectivity to machines within the organization.	Network pool

Organization networks require the support of one or both of the following objects, which must be created by a system administrator:

- External networks, which are backed by vSphere portgroup, VLAN, or DVswitch objects.
- Network Pools, which are backed by vSphere DVswitch objects.

After creating the necessary external networks and network pools, a system administrator can create organization networks based on either or both types of objects. The configuration of an organization network is specified in an `OrgNetwork` element, and includes the following information:

- A name for the network, specified in the `name` attribute of the `OrgNetwork` element.
- A `Configuration` element that specifies network configuration details.
  - For routed and directly connected networks, the `ParentNetwork` element contains a reference to the external network that the organization network connects to. The `FenceMode` element controls how the two networks connect. Specify a `FenceMode` of `bridged` for a direct connection to the parent network, or `natRouted` to specify a routed connection controlled by network `Features` such as a `NatService` or `FirewallService`. If you want the organization network to be isolated, with no external connection, omit the `ParentNetwork` element and specify the `FenceMode` as `isolated`.
  - The `Features` element defines features of the network. See "Network Services," on page 121.
  - Additional modifiable elements like `IpScope` and `RetainNetInfoAcrossDeployments`, and read-only elements such as `SyslogServerSettings` and `RouterInfo`. See the schema reference for more information about the type and scope of these elements..
- Routed or isolated organization networks must include a `NetworkPool` element.

Network Services

vApp and organization networks can include any of the following services, defined in the Features element of the network's Configuration.

DhcpService Provides DHCP services to virtual machines on the network.

FirewallService Allows you to create firewall rules that, when matched, block or allow incoming or outgoing network traffic. See "vCloud Director Firewall Configurations," on page 122.

IpsecVpnService Allows you to define one or more virtual private networks that virtual machines can use to connect to other networks in or outside of the cloud. Requires a routed network. See "Add a Virtual Private Network to a Routed Organization Network," on page 131.

NatService Provides network address translation services to computers on the network.

StaticRoutingService Allows you to specify static routes to other networks. Requires a routed network. See "Static Routing Configurations for vCloud Director Networks," on page 123.

For detailed information about the elements and attributes that define these services, see the schema reference.

External Networks and Network Pools

External networks and network pools are vSphere resources that support organization and vApp networks. A system administrator must create them, as described in "Create an External Network," on page 180 and "Create a Network Pool," on page 183.

When you create a network for an organization, the creation request must specify an external network, a network pool, or both. The VMWExtension element contains references to the available external networks and network pools, as shown in this excerpt.

```
<vmext:VMWExtension ... >
   ...
   <vcloud:Link
      rel="down"
      type="application/vnd.vmware.admin.vmwExternalNetworkReferences+xml"
      href="https://10.115.123.105/api/admin/extension/externalNetworkReferences" />
   <vcloud:Link
      rel="down"
      type="application/vnd.vmware.admin.vmwNetworkPoolReferences+xml"
      href="https://10.115.123.105/api/admin/extension/networkPoolReferences" />
   ...
</vmext:VMWExtension>
```

For more information about the VMWExtension element, see "Retrieve a List of vSphere Platform Operations and Objects for a Cloud," on page 49.

Each reference to an external network or network pool includes its type, name, and href attributes, as shown in this excerpt from a VMWExternalNetworkReferences element.

```
<vmext:VMWExternalNetworkReferences ... >
   ...
   <vmext:ExternalNetworkReference
      type="application/vnd.vmware.admin.vmwexternalnet+xml"
      name="pnet_vc1"
      href="https://vcloud.example.com/api/admin/extension/externalnet/59" />
   <vmext:ExternalNetworkReference
      type="application/vnd.vmware.admin.vmwexternalnet+xml"
      name="pnet_vc2"
      href="https://vcloud.example.com/api/admin/extension/externalnet/13" />
</vmext:VMWExternalNetworkReferences>
```

The corresponding element for network pools, `VMWNetworkPoolReferences`, is similar. In most cases, you can supply just the `href` attribute value when you specify an external network or network pool in an organization network creation request. You can retrieve additional information about the external network or network pool by making a GET request to its `href` attribute value.

vCloud Director Firewall Configurations

When you include a `FirewallService` in a network configuration, it is enabled by default and blocks all incoming traffic to the network. You can add firewall rules to allow incoming traffic, block outgoing traffic, or both. Firewall rules can specify source and destination IP addresses and ports for TCP, UDP, or both.

You configure a firewall service in the `Features` element of a routed organization network or a routed or fenced vApp network. In addition to the service parameters, you can specify one or more firewall rules to apply to incoming or outgoing packets.

Firewall Rules

Each firewall rule specifies a packet direction (incoming or outgoing), protocol, IP address, and port. Packets that match the criteria in the rule are subject to an action that the rule's `Policy` element defines. The action can be to forward the packet to the destination IP address and port, or drop it and optionally log a message describing the packet that was dropped. Packets that do not match any rule are subject to the policy contained in the `DefaultAction` element of the `FirewallService`

Firewall Rule Logging

The `Configuration` element of a network can include `SyslogServerSettings` that specify IP addresses to which `syslog` messages are sent. When you specify a value of `true` in the `EnableLogging` element of a `FirewallRule`, all packets that trigger the rule are logged to the configured `syslog` server. Logging for all rules is controlled by the value of the `LogDefaultAction` element of the `FirewallService`.

Example: Firewall Service Definition with Two Rules

The following fragment of a network `Configuration` defines a firewall service with two rules. Each rule is defined in a `FirewallRule` element, and can include the following specifications.

Policy
The default policy value, `allow`, causes the firewall to forward packets that match the rules. Specify `drop` to drop packets that match the rules.

Protocols
By default, a rule applies to both UDP and TCP protocols. You can limit the rule to one protocol or the other by including `Tcp` and `Udp` elements in `Protocols` and specifying a value of `true` or `false` for each.

SourcePort
Specify a source IP port, or set to `-1` to match any port.

SourceIp
Specify a source IP address, or use `Any` to match any address.

DestinationIp
Specify a destination IP address, or use `Any` to match any address.

Direction
Use `in` to apply the rule to incoming packets, which are sent from an external network to an organization network, or from an organization network to a vApp network. Use `out` to apply the rule to outgoing packets, which are packets sent from an organization network or vApp network to any other network.

EnableLogging
Set to `true` to log all packets that trigger this rule. See "Firewall Rule Logging," on page 122.

Rules are applied to packets in the order in which the `FirewallRule` elements appear in the `FirewallService` definition.

```
<FirewallService>
   <IsEnabled>true</IsEnabled>
   <DefaultAction>allow</DefaultAction>
   <LogDefaultAction>false</LogDefaultAction>
   <FirewallRule>
      <IsEnabled>true</IsEnabled>
      <Description>allow ssh</Description>
      <Policy>allow</Policy>
      <Protocols>
         <Tcp>true</Tcp>
      </Protocols>
      <Port>22</Port>
      <DestinationIp>Any</DestinationIp>
      <SourcePort>-1</SourcePort>
      <SourceIp>Any</SourceIp>
      <Direction>in</Direction>
      <EnableLogging>false</EnableLogging>
   </FirewallRule>
   <FirewallRule>
      <IsEnabled>true</IsEnabled>
      <Description>deny outgoing rdp</Description>
      <Policy>drop</Policy>
      <Protocols>
         <Tcp>true</Tcp>
      </Protocols>
      <Port>3389</Port>
      <DestinationIp>Any</DestinationIp>
      <SourcePort>-1</SourcePort>
      <SourceIp>Any</SourceIp>
      <Direction>out</Direction>
      <EnableLogging>true</EnableLogging>
   </FirewallRule>
</FirewallService>
```

You can see this fragment in the context of an organization network in "Example: Create an Organization Network With a Routed Connection," on page 128.

Static Routing Configurations for vCloud Director Networks

A network `Configuration` element can define a static routing service and specify one or more static routes between routed vApp or organization networks.

You can create static routes between two routed networks that do not have overlapping IP address spaces. Static routing service details and routes are defined in a `StaticRoutingService` element contained by the `Features` element of a network's `Configuration`. A `StaticRoutingService` element can contain zero or more `StaticRoute` elements. Each `StaticRoute` specification requires the following elements:

Name A name for the route.

Network The network specification in CIDR notation.

NextHopIp The IP address of the next hop on the route. This address is typically the value in the `ExternalIp` element of the `RouterInfo` from the network to which this static route connects.

Interface Specify `internal` if `NextHopIp` contains an IP address in the same network. Specify `external` if `NextHopIp` contains an IP address in a different network.

Example: Static Routes Between vApp Networks

Assume two vApp networks that have the following properties:

- The `Configuration` of the vApp network in vApp1 has a `RouterInfo` element whose `ExternalIp` value is `192.168.0.100`.
- The `Configuration` of the vApp network in vApp2 has a `RouterInfo` element whose `ExternalIp` value is `192.168.0.101`.
- Both vApp networks have the same `ParentNetwork`, an organization network whose network specification in CIDR notation is `192.168.0.0/24`.

You can enable static routing between these two vApp networks by inserting a `StaticRoutingService` element in the `Features` of each vApp network `Configuration`. This excerpt from the `NetworkConfigSection` of vApp1 shows the network's `Configuration` and `Features` elements..

```
<!-- Static route from vApp network 1 to vApp network 2 -->
<Configuration>
   ...
   <Features>
      <StaticRoutingService>
         <IsEnabled>true</IsEnabled>
         <StaticRoute>
            <Name>TovAppNet2</Name>
            <Network>192.168.2.0/24</Network>
            <NextHopIp>192.168.0.101</NextHopIp>
            <Interface>External</Interface>
         </StaticRoute>
       </StaticRoutingService>
   </Features>
   ...
</Configuration>
```

This is a similar excerpt from the `NetworkConfigSection` of vApp2.

```
<!-- Static route from vApp network 2 to vApp network 1 -->
<Configuration>
   ...
   <Features>
      <StaticRoutingService>
         <IsEnabled>true</IsEnabled>
         <StaticRoute>
            <Name>TovAppNet1</Name>
            <Network>192.168.1.0/24</Network>
            <NextHopIp>192.168.0.100</NextHopIp>
            <Interface>External</Interface>
         </StaticRoute>
      </StaticRoutingService>
   </Features>
   ...
</Configuration>
```

Add a Network to an Organization

To add a network to an organization, a system administrator POSTs an `OrgNetwork` element to the organization's `add` URL for `networks`.

A newly created organization has no networks in it. A system administrator must create an organization network from resources that an external network or network pool provide. The contents of the `Configuration` element of the `OrgNetwork` define the properties of the network, including its connections to other networks. See "Create an Organization Network With a Direct Connection," on page 125, "Create an Organization Network With a Routed Connection," on page 128, and "Create an Isolated Organization Network," on page 136.

For more information about the types of networks you can create and the resources on which they depend, see "About vCloud Director Networks," on page 118.

Prerequisites

Verify that you are logged in as a system administrator.

Procedure

1 Retrieve the XML representation of the organization to which you want to add the network.

   This request retrieves an administrative view of an organization.

```
GET https://vcloud.example.com/api/admin/org/26
```

2 Examine the response to locate the `Link` element that contains the URL for adding networks to the organization.

   This element has a `rel` attribute value of `add` and a `type` attribute value of `application/vnd.vmware.admin.network+xml`, as shown here:

```
<Link
   href="https://vcloud.example.com/api/admin/org/26/networks"
   rel="add"
   type="application/vnd.vmware.admin.network+xml"/>
```

3 Create an `OrgNetwork` element.

4 POST the `OrgNetwork` element to the URL described in Step 2.

The server creates the new organization network, and returns an `OrgNetwork` element that includes the contents you POSTed, along with a set of `Link` elements that you can use to access, remove, or modify it. Users can reference this network using the URL specified in the `href` attribute of the `Link` where `rel="alternate"`. The new network is added to the list of `Networks` shown in the `AdminOrg` element that represents the organization.

Create an Organization Network With a Direct Connection

An organization network with a direct connection is accessible by multiple organizations. Virtual machines belonging to different organizations can connect to and see traffic on this network.

An organization network with a direct connection is configured as a child network of one of the external networks provisioned to the cloud by the system administrator.

Prerequisites

- Verify that you are logged in as a system administrator.
- Retrieve the list of external networks. For information about how to retrieve this list, see "External Networks and Network Pools," on page 121.

Procedure

1 Choose an external network on which to base the new network.

2 Create an `OrgNetwork` element.

  Specify the `href` of the external network you chose in Step 1 in the `ParentNetwork` element. The `type` and `name` attributes are optional here. See the request portion of "Example: Create an Organization Network With a Direct Connection," on page 126.

3 POST the `OrgNetwork` element you created in Step 2 to the URL for adding networks to the organization.

  See the request portion of "Example: Create an Organization Network With a Direct Connection," on page 126

The server creates the new organization network, and returns an `OrgNetwork` element that includes the contents you POSTed, along with a set of `Link` elements that you can use to access, remove, or modify it. Users can reference this network using the URL specified in the `href` attribute of the `Link` where `rel="alternate"`. The new network is added to the list of `Networks` shown in the `AdminOrg` element that represents the organization.

See the response portion of "Example: Create an Organization Network With a Direct Connection," on page 126.

Example: Create an Organization Network With a Direct Connection

This example adds a directly-connected network with a DHCP service to the organization created in "Example: Create an Organization," on page 113. Because the network has a `Configuration` whose `ParentNetwork` element specifies an `ExternalNetwork` to connect to and sets the `FenceMode` to `bridged`, it provides a direct connection to the parent network.

Request:

```
POST https://vcloud.example.com/api/admin/org/26/networks
Content-Type: application/vnd.vmware.admin.orgNetwork+xml
...
<?xml version="1.0" encoding="UTF-8"?>
<OrgNetwork
   name="Internet"
   xmlns="http://www.vmware.com/vcloud/v1.5">
   <Description>Bridged to the public Internet</Description>
   <Configuration>
      <ParentNetwork
         href="https://vcloud.example.com/api/admin/extension/externalnet/network/7" />
      <FenceMode>bridged</FenceMode>
         <Features>
            <DhcpService>
            <IsEnabled>false</IsEnabled>
            <DefaultLeaseTime>3600</DefaultLeaseTime>
            <MaxLeaseTime>7200</MaxLeaseTime>
            <IpRange>
               <StartAddress>192.168.0.2</StartAddress>
               <EndAddress>192.168.0.99</EndAddress>
            </IpRange>
         </DhcpService>
      </Features>
   </Configuration>
</OrgNetwork>
```

Response:

```
201 Created
Content-Type: application/vnd.vmware.admin.orgNetwork+xml
...
<OrgNetwork
   xmlns="http://www.vmware.com/vcloud/v1.5"
   name="Internet"
   type="application/vnd.vmware.admin.orgNetwork+xml"
   id="urn:vcloud:network:54"
   href="https://vcloud.example.com/api/admin/network/54">
   <Link
      rel="alternate"
      type="application/vnd.vmware.vcloud.network+xml"
      href="https://vcloud.example.com/api/network/54" />
   <Link
      rel="up"
      type="application/vnd.vmware.admin.organization+xml"
      href="https://vcloud.example.com/api/admin/org/26" />
   <Link
      rel="down"
      type="application/vnd.vmware.vcloud.metadata+xml"
      href="https://vcloud.example.com/api/admin/network/54/metadata" />
   <Description>Bridged to the public Internet</Description>
   <Tasks>
      <Task ...
         operation="Creating Network Internet (54)" ... >
      </Task>
   </Tasks>
   <Configuration>
      <IpScope>
         <IsInherited>true</IsInherited>
         <Gateway>10.147.122.190</Gateway>
         <Netmask>255.255.255.192</Netmask>
         <Dns1>10.115.120.71</Dns1>
         <DnsSuffix>example.com</DnsSuffix>
         <IpRanges>
            <IpRange>
               <StartAddress>10.147.122.100</StartAddress>
               <EndAddress>10.147.122.200</EndAddress>
            </IpRange>
         </IpRanges>
         <AllocatedIpAddresses />
      </IpScope>
      <ParentNetwork
         type="application/vnd.vmware.admin.orgNetwork+xml"
         name="ExternalNetwork-VC1"
         href="https://vcloud.example.com/api/admin/network/7" />
      <FenceMode>bridged</FenceMode>
   </Configuration>
</OrgNetwork>
```

Create an Organization Network With a Routed Connection

An organization network with a routed connection is accessible only by this organization, and provides controlled access to an external network. System administrators and organization administrators can configure settings to make specific virtual machines accessible from the external network.

When you create an organization network with a routed connection, you actually create two networks joined by a virtual router. The network outside the firewall is an existing external network. The one inside the firewall is created from network pool resources. Firewall and NAT parameters for the virtual router are configured in the `Features` element of the network's `Configuration`.

Prerequisites

- Verify that you are logged in as a system administrator.
- Retrieve the lists of external networks and network pools. An organization network with a routed connection requires you to specify an external network and a network pool. See "External Networks and Network Pools," on page 121 for information about how to retrieve these lists.

Procedure

1. Choose an external network on which to base the external (outside the firewall) part of the new network.
2. Choose a network pool on which to base the internal (inside the firewall) part of the new network.
3. Create an `OrgNetwork` element.

   The following requirements apply when you are constructing this element:

   - Set the `FenceMode` to `natRouted`.
   - Specify the `href` of the external network you chose in Step 1 in the `ParentNetwork` element. The `type` and `name` attributes are optional.
   - Specify the `href` of the network pool you chose in Step 2 in the `NetworkPool` element. The `type` and `name` attributes are optional.

   See the request portion of "Example: Create an Organization Network With a Routed Connection," on page 128.
4. POST the `OrgNetwork` element you created in Step 3 to the URL for adding networks to the organization.

   See the request portion of "Example: Create an Organization Network With a Routed Connection," on page 128.

The server creates the new organization network, and returns an `OrgNetwork` element that includes the contents you POSTed, along with a set of `Link` elements that you can use to access, remove, or modify it. Users can reference this network using the URL specified in the `href` attribute of the `Link` where `rel="alternate"`. The new network is added to the list of `Networks` shown in the `AdminOrg` element that represents the organization.

See the response portion of "Example: Create an Organization Network With a Routed Connection," on page 128.

Example: Create an Organization Network With a Routed Connection

This example adds a network to the organization created in "Example: Create an Organization," on page 113. The new network has a routed connection to an external network. The `Configuration` includes a `ParentNetwork` element that specifies an `ExternalNetwork` to which to connect. It also includes a `NetworkPool`, which provides resources needed to create the internal network inside the firewall.

Request:

```
POST https://vcloud.example.com/api/admin/org/26/networks
Content-Type: application/vnd.vmware.admin.orgNetwork+xml
...
<?xml version="1.0" encoding="UTF-8"?>
<OrgNetwork
   xmlns="http://www.vmware.com/vcloud/v1.5"
   name="RoutedOrgNetwork">
   <Description>NAT-routed organization network</Description>
   <Configuration>
      <IpScope>
         <IsInherited>false</IsInherited>
         <Gateway>192.168.0.1</Gateway>
         <Netmask>255.255.255.0</Netmask>
         <Dns1>10.147.115.1</Dns1>
         <DnsSuffix>example.com</DnsSuffix>
         <IpRanges>
            <IpRange>
               <StartAddress>192.168.0.100</StartAddress>
               <EndAddress>192.168.0.199</EndAddress>
            </IpRange>
         </IpRanges>
      </IpScope>
      <ParentNetwork
         href="https://vcloud.example.com/api/admin/extension/externalnet/119" />
      <FenceMode>natRouted</FenceMode>
      <RetainNetInfoAcrossDeployments>false</RetainNetInfoAcrossDeployments>
      <Features>
         <DhcpService>
            <IsEnabled>false</IsEnabled>
            <DefaultLeaseTime>3600</DefaultLeaseTime>
            <MaxLeaseTime>7200</MaxLeaseTime>
            <IpRange>
               <StartAddress>192.168.0.2</StartAddress>
               <EndAddress>192.168.0.99</EndAddress>
            </IpRange>
         </DhcpService>
         <FirewallService>
            <IsEnabled>true</IsEnabled>
            <DefaultAction>allow</DefaultAction>
            <LogDefaultAction>false</LogDefaultAction>
            <FirewallRule>
               <IsEnabled>true</IsEnabled>
               <Description>allow ssh</Description>
               <Policy>allow</Policy>
               <Protocols>
                  <Tcp>true</Tcp>
               </Protocols>
               <Port>22</Port>
               <DestinationIp>Any</DestinationIp>
               <SourcePort>-1</SourcePort>
               <SourceIp>Any</SourceIp>
               <Direction>in</Direction>
               <EnableLogging>false</EnableLogging>
            </FirewallRule>
            <FirewallRule>
               <IsEnabled>true</IsEnabled>
               <Description>deny outgoing rdp</Description>
               <Policy>drop</Policy>
               <Protocols>
                  <Tcp>true</Tcp>
               </Protocols>
               <Port>3389</Port>
               <DestinationIp>Any</DestinationIp>
               <SourcePort>-1</SourcePort>
               <SourceIp>Any</SourceIp>
               <Direction>out</Direction>
               <EnableLogging>true</EnableLogging>
            </FirewallRule>
         </FirewallService>
         <NatService>
            <IsEnabled>false</IsEnabled>
            <NatType>portForwarding</NatType>
            <Policy>allowTraffic</Policy>
         </NatService>
      </Features>
   </Configuration>
   <NetworkPool
      href="https://vcloud.example.com/api/admin/extension/networkPool/313"></NetworkPool>
   <AllowedExternalIpAddresses />
</OrgNetwork>
```

The response echoes the request, and includes a Task element that tracks creation of the object and links that an administrator can use to manage it after it has been created.

Response:

```
201 Created
Content-Type: application/vnd.vmware.admin.orgNetwork+xml
...
<OrgNetwork
   xmlns="http://www.vmware.com/vcloud/v1.5"
   name="RoutedOrgNetwork"
   id="urn:vcloud:network:59"
   type="application/vnd.vmware.vcloud.network+xml"
   href="https://vcloud.example.com/api/admin/network/59" ... >
   <Link
      rel="alternate"
      type="application/vnd.vmware.vcloud.network+xml"
      href="https://vcloud.example.com/api/network/59" />
   <Link
      rel="up"
      type="application/vnd.vmware.admin.organization+xml"
      href="https://vcloud.example.com/api/admin/org/26" />
   <Link
      rel="down"
      type="application/vnd.vmware.vcloud.metadata+xml"
      href="https://vcloud.example.com/api/admin/network/59/metadata" />
   <Description>NAT-routed organization network</Description>
   <Tasks>
      <Task
```

```
         status="running"
         startTime="2011-06-28T10:31:39.788-07:00"
         operationName="networkCreateOrgNetwork"
         operation="Creating Network RoutedOrgNetwork(54)"
         expiryTime="2011-10-06T10:31:39.788-07:00"
         ...
      </Task>
   </Tasks>
   <Configuration>
      ...
   </Configuration>
   <NetworkPool
      href="https://vcloud.example.com/api/admin/extension/networkPool/313" />
   <AllowedExternalIpAddresses />
</OrgNetwork>
```

Add a Virtual Private Network to a Routed Organization Network

An organization network with a routed connection can include a virtual private network feature. You can include the virtual private network when you create the organization network, or you can add it later.

The Features element of an organization network with a routed connection can define a virtual private network (VPN) with zero or more tunnels. Virtual machines can use these tunnels for secure communications between organization networks in a single cloud, between two clouds, or between a cloud and any external IP address.

You can include the definition of an IpsecVpnService when you create an organization network. This procedure shows how to add one later, and provides a generalized procedure for updating any kind of organization network.

Prerequisites

Verify that you are logged in as a system administrator.

Procedure

1 Retrieve a routed organization network.

   Use a request like this one:

   ```
   GET https://vcloud.example.com/api/admin/network/59
   ```

2 Insert an IpsecVpnService element into the Features of the retrieved OrgNetwork.

3 POST the modified OrgNetwork element to the network's edit URL.

Example: Add a Virtual Private Network to a Routed Organization Network

This example adds a VPN to the organization network created in "Example: Create an Organization Network With a Routed Connection," on page 128. Start with the OrgNetwork element from the response to the request made in Step 1

```
<OrgNetwork
   xmlns="http://www.vmware.com/vcloud/v1.5"
   name="RoutedOrgNetwork"
   id="urn:vcloud:network:59"
   type="application/vnd.vmware.vcloud.network+xml"
   href="https://vcloud.example.com/api/admin/network/59" ... >
  <VCloudExtension
      required="false">
      <vmext:VimObjectRef>
         <vmext:VimServerRef
```

```
            type="application/vnd.vmware.admin.vmwvirtualcenter+xml"
            href="https://vcloud.example.com/api/admin/extension/vimServer/82" />
         <vmext:MoRef>dvportgroup-908</vmext:MoRef>
         <vmext:VimObjectType>DV_PORTGROUP</vmext:VimObjectType>
      </vmext:VimObjectRef>
   </VCloudExtension>
   <Link
      rel="alternate"
      type="application/vnd.vmware.vcloud.network+xml"
      href="https://vcloud.example.com/api/network/59" />
   <Link
      rel="edit"
      type="application/vnd.vmware.admin.orgNetwork+xml"
      href="https://vcloud.example.com/api/admin/network/59" />
   <Link
      rel="remove"
      href="https://vcloud.example.com/api/admin/network/59" />
   <Link
      rel="up"
      type="application/vnd.vmware.admin.organization+xml"
      href="https://vcloud.example.com/api/admin/org/26" />
   <Link
      rel="repair"
      type="application/vnd.vmware.admin.orgNetwork+xml"
      href="https://vcloud.example.com/api/admin/network/59/action/reset" />
   <Link
      rel="down"
      type="application/vnd.vmware.vcloud.metadata+xml"
      href="https://vcloud.example.com/api/admin/network/59/metadata" />
   <Link
      rel="syncSyslogSettings"
      type="application/vnd.vmware.vcloud.task+xml"
      href="https://vcloud.example.com/api/admin/network/59/action/syncSyslogServerSettings" />
   <Description>NAT-routed organization network</Description>
   <Configuration>
      <IpScope>
         <IsInherited>false</IsInherited>
         <Gateway>192.168.0.1</Gateway>
         <Netmask>255.255.255.0</Netmask>
         <Dns1>10.147.115.1</Dns1>
         <DnsSuffix>example.com</DnsSuffix>
         <IpRanges>
            <IpRange>
               <StartAddress>192.168.0.100</StartAddress>
               <EndAddress>192.168.0.199</EndAddress>
            </IpRange>
         </IpRanges>
      </IpScope>
      <ParentNetwork
         href="https://vcloud.example.com/api/admin/extension/externalnet/119" />
      <FenceMode>natRouted</FenceMode>
      <RetainNetInfoAcrossDeployments>false</RetainNetInfoAcrossDeployments>
      <Features>
         <DhcpService>
            <IsEnabled>false</IsEnabled>
            <DefaultLeaseTime>3600</DefaultLeaseTime>
            <MaxLeaseTime>7200</MaxLeaseTime>
            <IpRange>
               <StartAddress>192.168.0.2</StartAddress>
               <EndAddress>192.168.0.99</EndAddress>
            </IpRange>
         </DhcpService>
         <FirewallService>
            <IsEnabled>true</IsEnabled>
            <DefaultAction>allow</DefaultAction>
            <LogDefaultAction>false</LogDefaultAction>
            <FirewallRule>
               <IsEnabled>true</IsEnabled>
               <Description>allow ssh</Description>
               <Policy>allow</Policy>
               <Protocols>
                  <Tcp>true</Tcp>
               </Protocols>
               <Port>22</Port>
               <DestinationIp>Any</DestinationIp>
               <SourcePort>-1</SourcePort>
               <SourceIp>Any</SourceIp>
               <Direction>in</Direction>
               <EnableLogging>false</EnableLogging>
            </FirewallRule>
            <FirewallRule>
               <IsEnabled>true</IsEnabled>
               <Description>deny outgoing rdp</Description>
               <Policy>drop</Policy>
               <Protocols>
                  <Tcp>true</Tcp>
               </Protocols>
               <Port>3389</Port>
               <DestinationIp>Any</DestinationIp>
               <SourcePort>-1</SourcePort>
               <SourceIp>Any</SourceIp>
               <Direction>out</Direction>
               <EnableLogging>true</EnableLogging>
            </FirewallRule>
         </FirewallService>
         <NatService>
            <IsEnabled>false</IsEnabled>
            <NatType>portForwarding</NatType>
            <Policy>allowTraffic</Policy>
         </NatService>
      </Features>
     <SyslogServerSettings />
      <RouterInfo>
         <ExternalIp>10.147.65.7</ExternalIp>
      </RouterInfo>
   </Configuration>
   <NetworkPool
      type="application/vnd.vmware.admin.networkPool+xml"
```

```
      name="netpool_vc1"
      href="https://vcloud.example.com/api/admin/extension/networkPool/313" />
   <AllowedExternalIpAddresses />
</OrgNetwork>
```

Modify the `OrgNetwork` response to create a new request body. Modifications shown here include the following:

- `Link` elements, href attributes, and other content that the server adds to a response but ignores in a request have been removed.
- The reference to the `NetworkPool` and `ParentNetwork` have been removed. These references cannot be changed after the network is created. If you include them in a request, they are ignored.
- Aan `IpsecVpnService` element has been inserted into the network's `Features`.
- The `Description` has been adjusted.

To update the network with these modifications, PUT the modified `OrgNetwork` to the network's `edit` URL

Request:

```
PUT https://vcloud.example.com/api/admin/network/59
Content-Type: application/vnd.vmware.admin.orgNetwork+xml
...
<?xml version="1.0" encoding="UTF-8"?>
<OrgNetwork
   xmlns="http://www.vmware.com/vcloud/v1.5"
   name="RoutedOrgNetwork" >
   <Description>NAT-routed organization network with VPN</Description>
   <Configuration>
      <IpScope>
         <IsInherited>false</IsInherited>
         <Gateway>192.168.0.1</Gateway>
         <Netmask>255.255.255.0</Netmask>
         <Dns1>10.147.115.1</Dns1>
         <DnsSuffix>example.com</DnsSuffix>
         <IpRanges>
            <IpRange>
               <StartAddress>192.168.0.100</StartAddress>
               <EndAddress>192.168.0.199</EndAddress>
            </IpRange>
         </IpRanges>
      </IpScope>
      <FenceMode>natRouted</FenceMode>
      <Features>
         <DhcpService>
            <IsEnabled>false</IsEnabled>
            <DefaultLeaseTime>3600</DefaultLeaseTime>
            <MaxLeaseTime>7200</MaxLeaseTime>
            <IpRange>
               <StartAddress>192.168.0.2</StartAddress>
               <EndAddress>192.168.0.99</EndAddress>
            </IpRange>
         </DhcpService>
         <FirewallService>
            <IsEnabled>true</IsEnabled>
            <DefaultAction>allow</DefaultAction>
            <LogDefaultAction>false</LogDefaultAction>
            <FirewallRule>
```

```
               <IsEnabled>true</IsEnabled>
               <Description>allow ssh</Description>
               <Policy>allow</Policy>
               <Protocols>
                  <Tcp>true</Tcp>
               </Protocols>
               <Port>22</Port>
               <DestinationIp>Any</DestinationIp>
               <SourcePort>-1</SourcePort>
               <SourceIp>Any</SourceIp>
               <Direction>in</Direction>
               <EnableLogging>false</EnableLogging>
            </FirewallRule>
            <FirewallRule>
               <IsEnabled>true</IsEnabled>
               <Description>deny outgoing rdp</Description>
               <Policy>drop</Policy>
               <Protocols>
                  <Tcp>true</Tcp>
               </Protocols>
               <Port>3389</Port>
               <DestinationIp>Any</DestinationIp>
               <SourcePort>-1</SourcePort>
               <SourceIp>Any</SourceIp>
               <Direction>out</Direction>
               <EnableLogging>true</EnableLogging>
            </FirewallRule>
         </FirewallService>
         <NatService>
            <IsEnabled>false</IsEnabled>
            <NatType>portForwarding</NatType>
            <Policy>allowTraffic</Policy>
         </NatService>
         <IpsecVpnService>
            <IsEnabled>true</IsEnabled>
            <ExternalIpAddress>165.128.100.0</ExternalIpAddress>
            <PublicIpAddress>165.128.100.1</PublicIpAddress>
            <IpsecVpnTunnel>
               <Name>Holland</Name>
               <Description>The Holland Tunnel</Description>
               <IpsecVpnThirdPartyPeer />
               <PeerIpAddress>60.100.10.100</PeerIpAddress>
               <PeerNetworkAddress>60.100.10.10</PeerNetworkAddress>
               <PeerNetworkMask>255.255.0.0</PeerNetworkMask>
               <SharedSecret>gP3c768qGthGVbo77HHUMbK4htc8Pm5w</SharedSecret>
               <EncryptionProtocol>AES</EncryptionProtocol>
               <Mtu>1400</Mtu>
               <IsEnabled>true</IsEnabled>
            </IpsecVpnTunnel>
         </IpsecVpnService>
      </Features>
    </Configuration>
</OrgNetwork>
```

The response is a Task element. When the task is complete, the network is updated.

Response:

```
202 Accepted
Content-Type: application/vnd.vmware.vcloud.task+xml
...
<Task ...
   operation="Updating Network RoutedOrgNetwork (59)" ... >
   ...
</Task>
```

Create an Isolated Organization Network

An isolated organization network is accessible only by the organization for which it is created. Only virtual machines in this organization can connect to and see traffic on the isolated network.

Prerequisites

- Verify that you are logged in as a system administrator.
- Retrieve the list of network pools. See "External Networks and Network Pools," on page 121 for information about how to retrieve these lists.

Procedure

1 Choose a network pool on which to base the internal (inside the firewall) part of the new network.

2 Create an OrgNetwork element.

Specify the name and href of the network pool you chose in Step 1 in the NetworkPool element of this OrgNetwork. See the request portion of "Example: Create an Isolated Organization Network," on page 136.

3 POST the OrgNetwork element you created in Step 2 to the URL for adding networks to the organization.

See the request portion of "Example: Create an Isolated Organization Network," on page 136.

The server creates the new organization network, and returns an OrgNetwork element that includes the contents you POSTed, along with a set of Link elements that you can use to access, remove, or modify it. Users can reference this network using the URL specified in the href attribute of the Link where rel="alternate". The new network is added to the list of Networks shown in the AdminOrg element that represents the organization.

See the response portion of "Example: Create an Isolated Organization Network," on page 136.

Example: Create an Isolated Organization Network

This example adds an isolated network with a DHCP service to the organization created in "Example: Create an Organization," on page 113. The Configuration does not need a ParentNetwork element. All the resources needed by the network come from the specified NetworkPool. Set the FenceMode to isolated.

Request:

```
POST https://vcloud.example.com/api/admin/org/26/networks
Content-Type: application/vnd.vmware.admin.orgNetwork+xml
...
<?xml version="1.0" encoding="UTF-8"?>
<OrgNetwork
   name="IsolatedOrgNet"
   xmlns="http://www.vmware.com/vcloud/v1.5">
   <Description>Isolated Organization Network</Description>
   <Configuration>
      <FenceMode>isolated</FenceMode>
      <Features>
         <DhcpService>
            <IsEnabled>false</IsEnabled>
            <DefaultLeaseTime>3600</DefaultLeaseTime>
            <MaxLeaseTime>7200</MaxLeaseTime>
            <IpRange>
               <StartAddress>192.168.0.2</StartAddress>
               <EndAddress>192.168.0.99</EndAddress>
            </IpRange>
         </DhcpService>
      </Features>
   </Configuration>
   <NetworkPool
      name="VC1-POOL"
      href="https://10.147.23.1/api/admin/extension/networkPool/203" />
</OrgNetwork>
```

The response echoes the request, and includes a `Task` element that tracks creation of the object and links that an administrator can use to manage it after it is created.

Response:

```
201 Created
Content-Type: application/vnd.vmware.admin.orgNetwork+xml
...
<OrgNetwork
   xmlns="http://www.vmware.com/vcloud/v1.5"
   name="IsolatedOrgNet"
   type="application/vnd.vmware.admin.orgNetwork+xml"
   id="urn:vcloud:network:60"
   href="https://vcloud.example.com/api/admin/network/60" ...>
   <Link
      rel="alternate"
      type="application/vnd.vmware.vcloud.network+xml"
      href="https://vcloud.example.com/api/network/60" />
   <Link
      rel="up"
      type="application/vnd.vmware.admin.organization+xml"
      href="https://vcloud.example.com/api/admin/org/26" />
   <Link
      rel="down"
      type="application/vnd.vmware.vcloud.metadata+xml"
      href="https://vcloud.example.com/api/admin/network/60/metadata" />
   <Description>Isolated Organization Network</Description>
   <Tasks>
      <Task ...
         operation="Creating Network IsolatedOrgNet (60)" ... >
      </Task>
   </Tasks>
   <Configuration>
      ...
   </Configuration>
</OrgNetwork>
```

Synchronize Syslog Server Settings for a Network

When you change the IP addresses of the primary or secondary syslog server for a cloud, you must also synchronize the syslog server settings for organization and vApp networks that include a firewall service that has logging enabled.

If a system administrator changes the `SyslogServerSettings` for a cloud, all organization and vApp networks that are configured with a firewall service whose `EnableLogging` element has a value of `true` must be synchronized with the new syslog server settings so that logging can continue without interruption.

Prerequisites

- To synchronize syslog server settings for an organization network, you must be an organization or system administrator.
- To synchronize syslog server settings for a vApp network, you must be the vApp owner.

Procedure

1 Retrieve the XML representation of the network.

Use a request like this one.

```
GET https://vcloud.example.com/api/admin/network/id
```

2 Examine the response to locate the `Link` element that contains the URL for the `syncSyslogServerSettings` action.

This element has a `rel` attribute value of `syncSyslogSettings` and a `type` attribute value of `application/vnd.vmware.vcloud.task+xml`, as shown in this excerpt:

```
<Link
      rel="syncSyslogSettings"
      type="application/vnd.vmware.vcloud.task+xml"

href="https://vcloud.example.com/api/admin/network/59/action/syncSyslogServerSettings" />
```

3 Make a POST request to the link described in Step 2.

The request does not have a request body. The response is a task.

Example: Synchronize Syslog Server Settings for a Network

This request synchronizes the syslog server settings for the `OrgNetwork` shown in "Example: Add a Virtual Private Network to a Routed Organization Network," on page 131.

```
POST https://vcloud.example.com/api/admin/network/59/action/syncSyslogServerSettings />
```

Response:

```
202 Accepted
Content-Type: application/vnd.vmware.vcloud.task+xml
...
<Task ...
   operation="Updating Network RoutedOrgNetwork (59)" ... >
   ...
</Task>
```

vDC Administration

A newly created organization has no vDCs in it. A system administrator must use system resources to create them.

A virtual datacenter (vDC) is a deployment environment for virtual systems and an allocation mechanism for resources such as networks, storage, CPU, and memory. In a vDC, computing resources are fully virtualized, and can be allocated based on demand, service level requirements, or a combination of the two.

vDC administration involves the following objects:

- A `ProviderVdc`, which a system administrator creates from vSphere platform resources. See "Create a Provider vDC," on page 173.
- An `AdminVdc`, which a system administrator creates to allocate a subset of `ProviderVdc` resources to a vDC in a specific organization. Organization members see an `AdminVdc` as a `Vdc`.

Add a vDC to an Organization

A system administrator can allocate resources from a provider vDC to a vDC in an organization by POSTing an `AdminVdc` element to an organization's `add` URL for `vdcs`.

Prerequisites

Verify that you are logged in as a system administrator.

Procedure

1 Retrieve the XML representation of the organization to which you want to add the vDC.

Use a request like this one:

```
GET https://vcloud.example.com/api/admin/org/26
```

2 Examine the response to locate the `Link` element that contains the URL for adding vDCs to the organization.

This element has a `rel` attribute value of `add` and a `type` attribute value of `application/vnd.vmware.admin.vdc+xml`, as shown here:

```
<Link
   href="https://vcloud.example.com/api/admin/org/5/vdcs"
   rel="add"
   type="application/vnd.vmware.admin.vdc+xml"/>
```

3 Choose a provider vDC to supply resources for the new organization vDC.

a Retrieve the XML representation of the VCloud object and examine the ProviderVdcReferences element it contains.

The following request retrieves the representation of the VCloud object:

```
GET https://vcloud.example.com/api/admin
```

The VCloud element contains a ProviderVdcReferences element . Each provider vDC available to the organization is represented in that element by a ProviderVdcReference element, as shown here:

```
<ProviderVdcReference
   type="application/vnd.vmware.admin.providervdc+xml"
   name="Main Provider"
   href="https://vcloud.example.com/api/admin/providervdc/2"/>
```

b (Optional) List the organization vDCs that each ProviderVdc supports.

The following request retrieves the list of organization vDCs that .../providervdc/2 supports:

```
GET https://vcloud.example.com/api/admin/providervdc/2/vdcReferences
```

Taking this optional step can help you allocate ProviderVdc resources equitably across the organization vDCs in a cloud.

4 Create an AdminVdc request body.

Include the ProviderVdcReference element you chose in Step 3. You must also include an AllocationModel element that specifies how provider vDC resources are allocated by this vDC. Choose one of the following values for AllocationModel:

AllocationVApp Pay as you go. Resources are committed to a vDC only when vApps are created in it. When you use this allocation model, any Limit values you specify for Memory and CPU are ignored when you create a vApp and returned as 0 when retrieve the request results. Resources available to this kind of vDC can grow or shrink as needed when its provider vDC has multiple resource pools.

AllocationPool Only a percentage of the resources you allocate are committed to the organization vDC.

ReservationPool All the resources you allocate are committed as a pool to the organization vDC. vApps in vDCs that support this allocation model can specify values for resources and limitations.

See the request portion of "Example: Allocate a vDC to an Organization," on page 140.

5 POST the AdminVdc request body to the organization's add link for vdcs.

See the request portion of "Example: Allocate a vDC to an Organization," on page 140.

The server creates the new vDC in the specified organization and returns an AdminVdc element that includes the contents you POSTed. The server also returns a set of Link elements that you can use to access, remove, or modify the new vDC. Users can reference this vDC using the URL specified in the href attribute in the Link where rel="alternate". See the response portion of "Example: Allocate a vDC to an Organization," on page 140.

Example: Allocate a vDC to an Organization

This example allocates an AllocationvApp vDC to the organization created in "Example: Create an Organization," on page 113.

Request:

```
POST https://vcloud.example.com/api/admin/org/26/vdcs
Content-Type: application/vnd.vmware.admin.vdc+xml
...
<?xml version="1.0" encoding="UTF-8"?>
<AdminVdc
   name="org26vdc1"
   xmlns="http://www.vmware.com/vcloud/v1.5">
   <Description>Example vDC</Description>
   <AllocationModel>AllocationVapp</AllocationModel>
   <StorageCapacity>
      <Units>MB</Units>
      <Allocated>3072</Allocated>
      <Limit>4096</Limit>
   </StorageCapacity>
   <ComputeCapacity>
      <Cpu>
         <Units>MHz</Units>
         <Allocated>2048</Allocated>
         <Limit>2048</Limit>
      </Cpu>
      <Memory>
         <Units>MB</Units>
         <Allocated>2048</Allocated>
         <Limit>2048</Limit>
      </Memory>
   </ComputeCapacity>
   <NicQuota>0</NicQuota>
   <NetworkQuota>0</NetworkQuota>
   <ProviderVdcReference
      type="application/vnd.vmware.admin.providervdc+xml"
      name="Main Provider"
      href="https://vcloud.example.com/api/admin/providervdc/2" />
</AdminVdc>
```

The response, a subset of which appears here, echoes the request, and includes a `Task` element that tracks creation of the vDC. The response also includes `Link` elements that enable administrative operations on the vDC, and a `Capabilities` element that lists the VMware hardware architectures that the vDC supports. These elements are retrieved from the provider vDC that you specified when you created the `AdminVdc`. While the vDC is under construction, its `status` remains `0`.

Response:

```
201 Created
Content-Type: application/vnd.vmware.admin.vdc+xml
...
<AdminVdc
   xmlns="http://www.vmware.com/vcloud/v1.5"
   status="0"
   name="org26vdc1"
   id="urn:vcloud:vdc:44"
   type="application/vnd.vmware.admin.vdc+xml"
   href="https://vcloud.example.com/api/admin/vdc/44" ... >
   <Link
      rel="up"
      type="application/vnd.vmware.admin.organization+xml"
      href="https://vcloud.example.com/api/admin/org/26" />
   <Link
      rel="down"
      type="application/vnd.vmware.vcloud.metadata+xml"
      href="https://vcloud.example.com/api/admin/vdc/44/metadata" />
   <Link
      rel="alternate"
      type="application/vnd.vmware.vcloud.vdc+xml"
      href="https://vcloud.example.com/api/vdc/44" />
   <Link
      rel="down"
      type="application/vnd.vmware.admin.OrganizationVdcResourcePoolSet+xml"
      href="https://vcloud.example.com/api/admin/extension/vdc/44/resourcePools" />
   <Description>Example vDC</Description>
   <Tasks>
      <Task
         status="running"
         startTime="2011-06-29T10:03:09.402-07:00"
         operationName="vdcCreateVdc"
         operation="Creating Virtual Datacenter org26vdc1(44)"
         ...
      </Task>
   </Tasks>
   <AllocationModel>AllocationVApp</AllocationModel>
   <StorageCapacity>
      ...
   </StorageCapacity>
   <ComputeCapacity>
      ...
   </ComputeCapacity>
   <ResourceEntities />
   <AvailableNetworks />
   <Capabilities>
      <SupportedHardwareVersions>
         <SupportedHardwareVersion>vmx-04</SupportedHardwareVersion>
         <SupportedHardwareVersion>vmx-07</SupportedHardwareVersion>
         <SupportedHardwareVersion>vmx-08</SupportedHardwareVersion>
      </SupportedHardwareVersions>
   </Capabilities>
   ...
</AdminVdc>
```

When construction is complete, the `status` changes to `1` and the `Task` is no longer included in representation. The following changes in the `AdminVdc` are also evident:

- A reference to the vCenter resource pool that supports the vDC appears in a `VCloudExtension` element.
- The network created in "Example: Create an Organization Network With a Direct Connection," on page 126 appears in the `AvailableNetworks` element. If you added more networks to the organization that contains this vDC, they also appear in that element.

- Additional `Link` elements are included for operations that are now valid, but that were not valid while the vDC was under construction.

```
<AdminVdc
   xmlns="http://www.vmware.com/vcloud/v1.5"
   xmlns:vmext="http://www.vmware.com/vcloud/extension/v1.5"
   status="1"
   name="org26vdc1"
   id="urn:vcloud:vdc:44"
   type="application/vnd.vmware.admin.vdc+xml"
   href="https://vcloud.example.com/api/admin/vdc/44" ... >
   <VCloudExtension
      required="false">
      <vmext:VimObjectRef>
         <vmext:VimServerRef
            type="application/vnd.vmware.admin.vmwvirtualcenter+xml"
            name="vc1"
            href="https://vcloud.example.com/api/admin/extension/vimServer/9" />
         <vmext:MoRef>resgroup-949</vmext:MoRef>
         <vmext:VimObjectType>RESOURCE_POOL</vmext:VimObjectType>
      </vmext:VimObjectRef>
   </VCloudExtension>
   <Link
      rel="up"
      type="application/vnd.vmware.admin.organization+xml"
      href="https://vcloud.example.com/api/admin/org/26" />
   <Link
      rel="edit"
      type="application/vnd.vmware.admin.vdc+xml"
      href="https://vcloud.example.com/api/admin/vdc/44" />
   <Link
      rel="disable"
      href="https://vcloud.example.com/api/admin/vdc/44/action/disable" />
   <Link
      rel="down"
      type="application/vnd.vmware.vcloud.metadata+xml"
      href="https://vcloud.example.com/api/admin/vdc/44/metadata" />
   <Link
      rel="alternate"
      type="application/vnd.vmware.vcloud.vdc+xml"
      href="https://vcloud.example.com/api/vdc/44" />
   <Link
      rel="down"
      type="application/vnd.vmware.admin.OrganizationVdcResourcePoolSet+xml"
      href="https://vcloud.example.com/api/admin/extension/vdc/44/resourcePools" />
   <Description>Example vDC</Description>
   <AllocationModel>AllocationVApp</AllocationModel>
   <StorageCapacity>
      <Units>MB</Units>
      <Allocated>4096</Allocated>
      <Limit>4096</Limit>
      <Used>0</Used>
      <Overhead>0</Overhead>
   </StorageCapacity>
   <ComputeCapacity>
```

```
      <Cpu>
         <Units>MHz</Units>
         <Allocated>0</Allocated>
         <Limit>2048</Limit>
         <Used>0</Used>
         <Overhead>0</Overhead>
      </Cpu>
      <Memory>
         <Units>MB</Units>
         <Allocated>0</Allocated>
         <Limit>2048</Limit>
         <Used>0</Used>
         <Overhead>0</Overhead>
      </Memory>
   </ComputeCapacity>
   <ResourceEntities />
   <AvailableNetworks>
      <Network
         type="application/vnd.vmware.admin.network+xml"
         name="ExampleRoutedOrgNetwork"
         href="https://vcloud.example.com/api/admin/network/59" />
   </AvailableNetworks>
   <Capabilities>
      <SupportedHardwareVersions>
         <SupportedHardwareVersion>vmx-04</SupportedHardwareVersion>
         <SupportedHardwareVersion>vmx-07</SupportedHardwareVersion>
         <SupportedHardwareVersion>vmx-08</SupportedHardwareVersion>
      </SupportedHardwareVersions>
   </Capabilities>
   <NicQuota>0</NicQuota>
   <NetworkQuota>0</NetworkQuota>
   <VmQuota>0</VmQuota>
   <IsEnabled>true</IsEnabled>
   <ResourceGuaranteedMemory>1.0</ResourceGuaranteedMemory>
   <ResourceGuaranteedCpu>0.01</ResourceGuaranteedCpu>
   <VCpuInMhz>256</VCpuInMhz>
   <IsThinProvision>false</IsThinProvision>
   <ProviderVdcReference
      type="application/vnd.vmware.admin.providervdc+xml"
      name="pvdc_vc1"
      href="https://vcloud.example.com/api/admin/providervdc/2" />
   <UsesFastProvisioning>false</UsesFastProvisioning>
</AdminVdc>
```

Enable, Disable, or Remove a vDC

A system administrator can use the `enable`, `disable`, and `remove` links in an `AdminVdc` body to enable, disable, or remove a vDC.

Prerequisites

Verify that you are logged in as a system administrator.

Procedure

- To enable a vDC, POST a request to its `action/enable` link.

- To disable a vDC, POST a request to its `action/disable` link.
- To remove a vDC:

  a POST a request to its `action/disable` link to disable the vDC.

  After the vDC is disabled, its representation includes a `rel="remove"` link.

  b Make a DELETE request to the vDC's `rel="remove"` link.

The server takes the requested action and returns an HTTP status of 204 No Content.

Catalog Administration

A newly created organization has no catalogs in it. After an organization administrator or catalog author creates a catalog, members of the organization can add items to it or remove items from it if they have the necessary rights.

Catalogs can contain references to vApp templates and media images. A system administrator or a privileged member of the organization that owns the catalog can create and remove these references. A catalog can be shared to make it visible to other members of an organization, and can be published to make it visible to administrators in other organizations. A system administrator specifies which organizations can publish catalogs, and an organization administrator controls access to catalogs by organization members.

Access to Catalogs

By default, a catalog is accessible only to its creator. An organization administrator or catalog author can grant catalog access to other members of the organization, individually, or collectively. See "Controlling Access to vApps and Catalogs," on page 157. An organization administrator or catalog author can also publish a catalog to make it available to administrators in other organizations. An organization administrator or system administrator can change the owner of a catalog. See "View or Change the Owner of an Object," on page 73.

Add a Catalog to an Organization

Every organization has an `add` URL for `catalogs`. An organization administrator or catalog author can create a catalog by POSTing an `AdminCatalog` element to this URL

A newly-created `Catalog` object is an empty container for references to vApp templates and media images.

Prerequisites

Verify that you are logged in as a system administrator, an organization administrator, or a user with the Catalog Author role.

Procedure

1 Retrieve the XML representation of the organization to which you want to add the catalog.

  Use a request like this one:

```
GET https://vcloud.example.com/api/admin/org/26
```

2 Examine the response to locate the `Link` element that contains the URL for adding catalogs to the organization.

  This element has a `rel` attribute value of `add` and a `type` attribute value of `application/vnd.vmware.admin.catalog+xml`, as shown here:

```
<Link
   href="https://vcloud.example.com/api/admin/org/5/catalogs"
   rel="add"
   type="application/vnd.vmware.admin.catalog+xml"/>
```

3. Create an AdminCatalog element.

   See the request portion of "Example: Create a Catalog," on page 146.

4. POST the AdminCatalog element to the organization's add URL for catalogs.

   See the request portion of "Example: Create a Catalog," on page 146.

The server creates an empty catalog. See the response portion of "Example: Create a Catalog," on page 146.

Example: Create a Catalog

This example adds a catalog to the organization created in "Example: Create an Organization," on page 113.

Request:

```
POST https://vcloud.example.com/api/admin/org/26/catalogs
Content-Type: application/vnd.vmware.admin.catalog+xml
...
<?xml version="1.0" encoding="UTF-8"?>
<AdminCatalog
   xmlns="http://www.vmware.com/vcloud/v1.5"
   name="Example Catalog">
   <Description>New Catalog for Example Org</Description>
</AdminCatalog>
```

The response echoes the request, and includes these additions that the server creates:

- A URL, in the value of the href attribute of the response body, that references the new catalog.
- Links that you can use to add an item to the catalog, edit catalog properties, or remove the catalog.
- A link to an alternate view of this catalog. All users can access the catalog at this URL.
- An empty CatalogItems element.
- A Task that tracks the creation of the catalog.
- An IsPublished element whose content is the string false, indicating that the catalog is not published.

Response:

```
201 Created
Content-Type: application/vnd.vmware.admin.catalog+xml
...
<AdminCatalog
   xmlns:vcloud="http://www.vmware.com/vcloud/v1.5"
   name="Example Catalog"
   id="urn:vcloud:catalog:32"
   type="application/vnd.vmware.admin.catalog+xml"
   href="https://vcloud.example.com/api/admin/catalog/32">
   <Link
      rel="up"
      type="application/vnd.vmware.admin.organization+xml"
      href="https://vcloud.example.com/api/admin/org/26" />
   <Link
      rel="alternate"
      type="application/vnd.vmware.vcloud.catalog+xml"
      href="https://vcloud.example.com/api/catalog/32" />
   <Link
      rel="down"
      type="application/vnd.vmware.vcloud.owner+xml"
      href="https://vcloud.example.com/api/admin/catalog/32/owner" />
   <Link
      rel="add"
      type="application/vnd.vmware.vcloud.catalogItem+xml"
      href="https://vcloud.example.com/api/catalog/32/catalogItems" />
   <Link
      rel="edit"
      type="application/vnd.vmware.admin.catalog+xml"
      href="https://vcloud.example.com/api/admin/catalog/32" />
   <Link
      rel="remove"
      href="https://vcloud.example.com/api/admin/catalog/32" />
   <Link
      rel="publish"
      type="application/vnd.vmware.admin.publishCatalogParams+xml"
      href="https://vcloud.example.com/api/admin/catalog/32/action/publish" />
   <Tasks>
      <Task ...>
         ...
      <Task>
   <Tasks>
   <Description>New Catalog for Example Org</Description>
   <CatalogItems/>
   <IsPublished>false</IsPublished>
</AdminCatalog>
```

What to do next

Add CatalogItem elements to the catalog. See "Add an Item to a Catalog," on page 70.

Publish a Catalog

Publishing a catalog makes it visible to the administrators of all organizations in a cloud.

An organization administrator can publish catalogs if the organization's CanPublishCatalogs element has a value of true. The value of this element is controlled by the system administrator. To publish a catalog, make a POST request to the catalog's action/publish URL and supply a PublishCatalogParams body that sets the value of the catalog's IsPublished element to true.

Prerequisites

Verify that you are logged in as an organization administrator or system administrator.

Procedure

1 Retrieve the XML representation of the catalog to publish.

Use a request like this one, where *id* is the identifier of the catalog:

```
GET https://vcloud.example.com/api/admin/catalog/id
```

2 Examine the response to locate the Link element that contains the URL for publishing the catalog.

This element has a rel attribute value of publish and a type attribute value of application/vnd.vmware.admin.publishCatalogParams+xml, as shown here:

```
<Link
      rel="publish"
      type="application/vnd.vmware.admin.publishCatalogParams+xml"
      href="https://vcloud.example.com/api/admin/catalog/32/action/publish" />
```

3 Create a PublishCatalogParams element that contains an IsPublished element with a value of true.

4 POST the PublishCatalogParams body to the catalog's rel="publish" URL.

The catalog is published and becomes available to administrators in other organizations in the cloud.

Example: Publish a Catalog

Request:

```
POST https://vcloud.example.com/api/admin/catalog/32/action/publish
Content-Type: application/vnd.vmware.admin.publishCatalogParams+xml
...
<?xml version="1.0" encoding="UTF-8"?>
<PublishCatalogParams
   xmlns="http://www.vmware.com/vcloud/v1.5">
   <IsPublished>true</IsPublished>
</PublishCatalogParams>
```

Response:

```
204 No Content
```

User and Group Administration

A newly created organization has no users or groups in it. An organization administrator must create or import them.

An organization can contain an arbitrary number of users and groups. Users can be created by the organization administrator or imported from an LDAP directory service. Groups must be imported from the directory service. Permissions within an organization are controlled through the assignment of rights and roles to users and groups.

Local Users and Imported Users

Users can be created locally or imported from the organization's LDAP service if it has defined one. Local user accounts are stored in the vCloud Director database and managed by the organization administrator. Imported user accounts are stored in the LDAP database and managed by the LDAP administrator. If the LDAP administrator changes the properties of an imported user account, or if an imported user changes his password, the changes are not be visible in vCloud Director until an organization administrator imports the user again.

Modifying User or Group Metadata

An organization administrator can modify metadata such as name and description for a user or group object by creating a modified version of the User or Group element that represents the object and updating the object by making a PUT request to the object's rel="edit" link, supplying the modified element in the request body.

Create a User

An organization administrator can create user accounts that are local to the organization. Local user accounts are stored in the vCloud Director database.

Every user exists within the context of an organization. An organization administrator can create a local user in an organization by POSTing a User element to the organization's add URL for users, as shown in "Example: Create a Local User," on page 149.

When you create a user, you must include the Role and Password elements in the request body. The role can be a predefined role or one created by the organization administrator. For more information about retrieving a list of predefined roles, see "Retrieve an Administrative View of a Cloud," on page 47. For more information about creating new roles, see "Create a Role," on page 156.

Prerequisites

Verify that you are logged in as an organization administrator or system administrator.

Procedure

1 Create a User element that defines the user account properties.

  See the request portion of "Example: Create a Local User," on page 149.

2 POST the User element to the organization's add URL for users.

The server creates a user account in the vCloud Director database and returns an updated User element to the client.

Example: Create a Local User

This example adds the user to the organization created in "Example: Create an Organization," on page 113. The request includes an optional IsEnabled element that enables the user. If not present in the request, IsEnabled defaults to false.

The response is a User element, most of which does not appear in the example. The response includes a link that an administrator can use to edit user properties, and additional elements, such as IsDefaultCached and StoredVmQuota, inherited from the organization. The Password element, which must not be empty when you create a User, is never returned.

Request:

```
POST https://vcloud.example.com/api/admin/org/26/users
Content-Type: application/vnd.vmware.admin.user+xml
...
<?xml version="1.0" encoding="UTF-8"?>
<User
   xmlns="http://www.vmware.com/vcloud/v1.5"
   name="ExampleUser" >
   <FullName>Example User Full Name</FullName>
   <EmailAddress>user@example.com</EmailAddress>
   <IsEnabled>true</IsEnabled>
   <Role
      href="https://vcloud.example.com/api/admin/role/105" />
   <Password>Pa55w0rd</Password>
   <GroupReferences />
</User>
```

Response:

```
201 Created
Content-Type: application/vnd.vmware.admin.user+xml
...
<User
   xmlns="http://www.vmware.com/vcloud/v1.5"
   name="ExampleUser"
   id="urn:vcloud:user:85"
   type="application/vnd.vmware.admin.user+xml"
   href="https://vcloud.example.com/api/admin/user/85" ... >
   <Link
```

```
      rel="edit"
      type="application/vnd.vmware.admin.user+xml"
      href="https://vcloud.example.com/api/admin/user/85" />
   <FullName>Example User Full Name</FullName>
   <EmailAddress>user@example.com</EmailAddress>
   <IsEnabled>true</IsEnabled>
   <IsAlertEnabled>false</IsAlertEnabled>
   <IsDefaultCached>false</IsDefaultCached>
   <StoredVmQuota>0</StoredVmQuota>
   <DeployedVmQuota>0</DeployedVmQuota>
   <Role
      type="application/vnd.vmware.admin.role+xml"
      name="vApp User"
      href="https://vcloud.example.com/api/admin/role/105" />
   <GroupReferences />
</User>
```

Import a User from an LDAP Database

If an organization defines an LDAP service to use, an organization or system administrator can import user accounts from that service.

Importing a group from LDAP imports all the users in the group. See "Import a Group from an LDAP Database," on page 151. You can also import users individually.

Prerequisites

- Verify that you are logged in as an organization administrator or system administrator.
- Verify that your organization has defined an LDAP service to use.

Procedure

1. Create a User element that identifies the LDAP user account to import.

   The name attribute of the User element must match the LDAP user name, as specified in the organization's LDAP properties. You must include the Role element in the request body.

2. POST the User element to the organization's users URL.

The server matches the value of the name attribute in the request body with the value of the LDAP attribute that the organization specified in the value of the UserName element in the UserAttributes of its OrgLdapSettings. LDAP attributes such as userPrincipalName or samAccountName are common choices here. The server imports the user from the organization's LDAP service, and returns an updated User element to the client.

Example: Import a User from an LDAP Database

This example imports a user to the organization created in "Example: Create an Organization," on page 113. The request includes an optional IsEnabled element, so the user is enabled as soon as the import is complete.

The response is a User element, most of which is not shown in the example. The response includes a link that an administrator can use to edit user metadata, and additional elements, such as IsDefaultCached and StoredVmQuota, inherited from organization defaults. It also includes a NameInSource element, which contains the user's name as stored by the LDAP server, using the server's native encoding.

Request:

```
POST https://vcloud.example.com/api/admin/org/26/users
Content-Type: application/vnd.vmware.admin.user+xml
...
<?xml version="1.0" encoding="UTF-8"?>
<User
   xmlns="http://www.vmware.com/vcloud/v1.5"
   name="user@example.com"
   type="application/vnd.vmware.admin.user+xml">
   <IsEnabled>true</IsEnabled>
   <IsExternal>true</IsExternal>
    <Role
      type="application/vnd.vmware.admin.role+xml"
      href="https://vcloud.example.com/api/admin/role/105" />
</User>
```

Response:

```
201 Created
Content-Type: application/vnd.vmware.admin.user+xml
...
<User
   xmlns="http://www.vmware.com/vcloud/v1.5"
   name="user@example.com"
   id="urn:vcloud:user:85"
   type="application/vnd.vmware.admin.user+xml"
   href="https://vcloud.example.com/api/admin/user/85">
   <Link
      rel="edit"
      type="application/vnd.vmware.admin.user+xml"
      href="https://vcloud.example.com/api/admin/user/85" />
   <FullName>Imported User Full Name</FullName>
   <EmailAddress>user@example.com</EmailAddress>
   <IsEnabled>true</IsEnabled>
   <NameInSource>\F4\D3\42\8E\6A\BC\D3</NameInSource>
   <IsAlertEnabled>false</IsAlertEnabled>
   <IsDefaultCached>false</IsDefaultCached>
   <StoredVmQuota>0</StoredVmQuota>
   <DeployedVmQuota>0</DeployedVmQuota>
   <Role
      type="application/vnd.vmware.admin.role+xml"
      href="https://vcloud.example.com/api/admin/role/105" />
   <GroupReferences />
</User>
```

Import a Group from an LDAP Database

If an organization defines an LDAP service to use, an organization or system administrator can import groups from that service. You cannot create a group. You must import it.

Importing a group from LDAP imports all the users in the group. You can also import users individually. See "Import a User from an LDAP Database," on page 150.

Prerequisites

- Verify that you are logged in as an organization administrator or system administrator.

- Verify that your organization has defined an LDAP service to use.

Procedure

1 Create a Group element that identifies the LDAP group to import.

The name attribute of the Group element must match the LDAP group name, as specified in the organization's LDAP properties. You must include a Role element in the request body. The role specified in this element is assigned to all group members during the import.

2 POST the Group element to the organization's groups URL.

The server matches the value of the name attribute in the request body with the value of the LDAP attribute that the organization specified in the value of the GroupName element in the GroupAttributes of its OrgLdapSettings. The LDAP cn attribute is a common choice here. The server imports that group and all of its users from organization's LDAP service, and returns an updated Group element to the client.

Example: Import a Group from an LDAP Database

This example imports a group to the organization created in "Example: Create an Organization," on page 113. The response is a Group element, most of which does not appear in the example. The response includes a link that an administrator can use to edit group metadata such as name and description, and a UsersList element that includes a UserReference element for each user in the group. The response also includes a NameInSource element, which contains the group's name as stored by the LDAP server, using the server's native encoding.

Request:

```
POST https://vcloud.example.com/api/admin/org/26/groups
Content-Type: application/vnd.vmware.admin.group+xml
...
<?xml version="1.0" encoding="UTF-8"?>
<Group
   name="Engineering"
   xmlns="http://www.vmware.com/vcloud/v1.5">
   <Role
        href="https://vcloud.example.com/api/admin/role/102"/>
</Group>
```

Response:

```
201 Created
Content-Type: application/vnd.vmware.admin.group+xml
...
<Group
   xmlns="http://www.vmware.com/vcloud/v1.5"
   name="Engineering"
   id="urn:vcloud:group:44"
   type="application/vnd.vmware.admin.group+xml"
   href="https://vcloud.example.com/api/admin/group/44" ...>
   <Role
      type="application/vnd.vmware.admin.role+xml"
      name="vApp Wrangler"
      href="https://vcloud.example.com/api/admin/role/102" />
</Group>
```

Until the import is complete, the Group element contains only partial information. After the import is complete, the element includes a list of users and other information.

```
<Group
   xmlns="http://www.vmware.com/vcloud/v1.5"
   name="Engineering"
   id="urn:vcloud:group:44"
   type="application/vnd.vmware.admin.group+xml"
   href="https://vcloud.example.com/api/admin/group/44" ...>
   <Link
      rel="edit"
      type="application/vnd.vmware.admin.group+xml"
      href="https://vcloud.example.com/api/admin/group/44" />
   <Link
      rel="remove"
      href="https://vcloud.example.com/api/admin/group/44" />
   <Description>Research and development</Description>
   <NameInSource>\C5\AF\B9\D4\9E\B5\32\40\AD\C5\E3\8E\17\4C\0D\28</NameInSource>
   <UsersList>
      <UserReference
         type="application/vnd.vmware.admin.user+xml"
         name="User-1"
         href="https://vcloud.example.com/api/admin/user/18" />
      <UserReference
         type="application/vnd.vmware.admin.user+xml"
         name="User-3"
         href="https://vcloud.example.com/api/admin/user/19" />
   </UsersList>
   <Role
      type="application/vnd.vmware.admin.role+xml"
      name="vApp Wrangler"
      href="https://vcloud.example.com/api/admin/role/102" />
</Group>
```

Working With Roles and Rights

A role associates a role name with a set of rights. A newly created organization includes a set of predefined roles and rights inherited from the containing cloud. An organization administrator can add new roles or modify predefined roles. You cannot create or modify rights.

vCloud Director uses roles, and their associated rights, to determine whether a user or group is authorized to perform an operation. When you create or import a user or group, you must assign it a role. You can use one of the predefined roles, or you can create a role from existing rights.

Predefined roles and rights are properties of a cloud. Roles that an organization administrator creates are properties of the organization.

Predefined Roles and Their Rights

vCloud Director includes predefined roles. Each of these roles includes a set of default rights.

Table 6-3 lists the predefined vCloud Director roles and the default rights assigned to each role. A system administrator can create new roles and modify existing roles, except the System Administrator role.

Table 6-3. Default Rights for the Predefined Roles

	System Administrator	Organization Administrator	Catalog Author	vApp Author	vApp User	Console Access Only
vApp: Create/Reconfigure a vApp	X	X	X	X		
vApp: Delete a vApp	X	X	X	X	X	
vApp: Edit vApp Properties	X	X	X	X	X	
vApp: Start/Stop/Suspend/Reset a vApp	X	X	X	X	X	
vApp: Share a vApp	X	X	X	X	X	
vApp: Copy a vApp	X	X	X	X	X	
vApp: Access to VM Console	X	X	X	X	X	X
vApp: Change Owner	X	X				
vApp: Edit VM Properties	X	X	X	X	X	
vApp: Edit VM Memory	X	X	X	X		
vApp: Edit VM CPU	X	X	X	X		
vApp: Edit VM Network	X	X	X	X	X	
vApp: Edit VM Hard Disk	X	X	X	X		
vApp: Manage VM Password Settings	X	X	X	X	X	X
Catalog: Create/Delete a new Catalog	X	X	X			
Catalog: Edit Catalog Properties	X	X	X			
Catalog: Add a vApp from My Cloud	X	X	X	X		
Catalog: Publish a Catalog	X	X	X			
Catalog: Share a Catalog	X	X	X			
Catalog: View Private and Shared Catalogs	X	X	X	X		

Table 6-3. Default Rights for the Predefined Roles (Continued)

	System Administrator	Organization Administrator	Catalog Author	vApp Author	vApp User	Console Access Only
Catalog: View Published Catalogs	X	X				
Catalog: Change Owner	X	X				
Catalog Item: Edit vApp Template/Media Properties	X	X	X			
Catalog Item: Create/Upload a vApp Template or Media	X	X	X			
Catalog Item: Download a vApp Template	X	X	X			
Catalog Item: Copy/Move a vApp Template or Media	X	X	X	X		
Catalog Item: View vApp Templates and Media	X	X	X	X	X	
Catalog Item: Add to My Cloud	X	X	X	X	X	
Organization: Edit Organization Properties	X	X				
Organization: Edit SMTP Settings	X	X				
Organization: Edit Quotas Policy	X	X				
Organization: View Organizations	X	X				
Organization: Edit Organization Network Properties	X	X				
Organization: View Organization Networks	X	X				
Organization: Edit Leases Policy	X	X				

Table 6-3. Default Rights for the Predefined Roles (Continued)

	System Administrator	Organization Administrator	Catalog Author	vApp Author	vApp User	Console Access Only
Organization: Edit Password Policy	X	X				
Organization vDC: View Organization vDCs	X	X				
User: View Group/User	X	X				
General: Send Notification	X	X				
General: Administrator Control	X	X				
General: Administrator View	X	X				

Create a Role

An organization administrator can create a role by aggregating a set of rights in a `Role` element and POSTing it to the organization's `add` URL for roles.

Prerequisites

Verify that you are logged in as an organization administrator or system administrator.

Procedure

1 Create a `Role` element that defines the role with a name and a set of rights.

To get the `RightReference` objects you will need to populate the `Role`, retrieve the administrative view of the cloud, using a request like this one:

```
GET https://vcloud.example.com/api/admin
```

2 POST the `Role` element to the organization's `add` URL for `roles`.

See the request portion of "Example: Create a Role," on page 156.

The server creates a `Role` element and returns its representation to the client.

Example: Create a Role

This example adds a role named `vAppWrangler` to the organization created in "Example: Create an Organization," on page 113. The response is a `Role` element, most of which does not appear in the example. The response includes links that an administrator can use to edit or remove the role.

Request:

```
POST https://vcloud.example.com/api/admin/roles
Content-Type: application/vnd.vmware.admin.role+xml
...
<?xml version="1.0" encoding="UTF-8"?>
<Role name="vAppWrangler" xmlns="http://www.vmware.com/vcloud/v1.5">
   <Description>Create and manage vApps</Description>
   <RightReferences>
      <RightReference
          type="application/vnd.vmware.admin.right+xml"
          name="vApp: View"
          href="https://vcloud.example.com/api/admin/right/16"/>
      <RightReference
         type="application/vnd.vmware.admin.right+xml"
         name="vApp: Power Operations"
         href="https://vcloud.example.com/api/admin/right/9"/>
      <RightReference
         type="application/vnd.vmware.admin.right+xml"
         name="vApp: Download"
         href="https://vcloud.example.com/api/admin/right/11"/>
    </RightReferences>
</Role>
```

Response:

```
201 Created
Content-Type: application/vnd.vmware.admin.role+xml
...
<Role
   name="vAppWrangler"
   id="urn:vcloud:role:102"
   xmlns="http://www.vmware.com/vcloud/v1.5"
   href="https://vcloud.example.com/api/admin/role/102" ...>
   <Link
      rel="edit"
      type="application/vnd.vmware.admin.role+xml"
      href="https://vcloud.example.com/api/admin/role/102"/>
   <Link
      rel="remove"
      href="https://vcloud.example.com/api/admin/role/102"/>
   <Description>Create and manage vApps</Description>
   <RightReferences>
      ...
   </RightReferences>
</Role>
```

Controlling Access to vApps and Catalogs

An organization administrator can use `controlAccess` links to control access to vApps and catalogs.

Catalogs and vApps include two types of access control links:

- Links where `rel="down"`.

  ```
  <Link
     rel="down"
     type="application/vnd.vmware.vcloud.controlAccess+xml"
     href="https://vcloud.example.com/api/object-type/id/controlAccess/"/>
  ```

  Use this kind of link to retrieve the access control settings for the object identified in the `href` value.

- Links where `rel="controlAccess"`.

  ```
  <Link
     rel="controlAccess"
     type="application/vnd.vmware.vcloud.controlAccess+xml"
     href="https://vcloud.example.com/api/object-type/id/controlAccess/"/>
  ```

  Use this kind of link to specify new access control settings for the object identified in the `href` value. You specify the new access control settings in a `ControlAccessParams` element that you post to the URL that the `href` value of this link specifies.

Granting Access to All Members of an Organization

To specify access controls that apply to all members of an organization, an organization administrator can set `IsSharedToEveryone` to `true` and specify an access level in the `EveryoneAccessLevel` element. The following `ControlAccessParams` element grants read access to all members of the organization.

```
<ControlAccessParams
   xmlns="http://www.vmware.com/vcloud/v1.5">
   <IsSharedToEveryone>true</IsSharedToEveryone>
   <EveryoneAccessLevel>ReadOnly</EveryoneAccessLevel>
</ControlAccessParams>
```

Granting Access to Individual Members of an Organization

To specify access controls that apply to individuals, an organization administrator can set `IsSharedToEveryone` to `false` and specify an access level in an `AccessSettings` element that the `ControlAccessParams` request contains. An `AccessSettings` element is populated with one or more `AccessSetting` elements, each of which assigns an access level to the user identified in the `Subject` element. The following `ControlAccessParams` element grants full control to one user and read-only access to another user.

```
<ControlAccessParams
   xmlns="http://www.vmware.com/vcloud/v1.5">
   <IsSharedToEveryone>true</IsSharedToEveryone>
   <AccessSettings>
      <AccessSetting>
         <Subject
            type="application/vnd.vmware.admin.user+xml"
            href="https://vcloud.example.com/api/admin/user/40"/>
         <AccessLevel>FullControl</AccessLevel>
      </AccessSetting>
      <AccessSetting>
         <Subject
            type="application/vnd.vmware.admin.user+xml"
            href="https://vcloud.example.com/api/admin/user/45"/>
         <AccessLevel>ReadOnly</AccessLevel>
      </AccessSetting>
   </AccessSettings>
</ControlAccessParams>
```

The schema reference includes detailed information and examples for `controlAccess` operations and the `ControlAccessParams` element. See "About the Schema Reference Downloadable Archive," on page 19.

Viewing or Changing the Owner of a vApp or Catalog

Ownership of a `VApp` or `Catalog` object is expressed in an `Owner` element that you can retrieve from the object. This element contains a `User` element that identifies the owner with a reference to a specific user. The initial owner of an object is the user who created it.

A system administrator can view or change the owner of a `VApp` or `Catalog` object using the procedure documented in "View or Change the Owner of an Object," on page 73.

Using vCloud API Extensions to Provision and Manage a Cloud

7

The VMware vCloud API includes extensions that support operations on the vSphere platform, which provides resources to vCloud Director. A system administrator can use these extensions to retrieve or update the configuration of a cloud, add or remove resources such as vCenter servers, network pools and external networks, and import vApps and media from vCenter.

Only the system administrator can perform vSphere platform operations. Before you attempt these operations, log in to the System organization with the user name and password of the system administrator account that was created when vCloud Director was installed. See "Administrator Credentials and Privileges," on page 111.

This chapter includes the following topics:

Summary of vSphere Platform Extension Requests

vSphere platform extension requests allow a system administrator to operate on vSphere platform objects registered to vCloud Director.

- *API-URL* is a URL of the form `https://vcloud.example.com/api`.
- *id* is a unique identifier in the form of a UUID, as defined by RFC 4122.

Request URLs are always available when you GET the representation of the object on which they operate. URL forms are for reference purposes only. For more information about the requests and responses for each operation, see "About the Schema Reference Downloadable Archive," on page 19.

Table 7-1. Summary of vSphere Platform Extension Requests

Operation	Request	Request Body	Response
Retrieve a list of provider vDCs in a cloud.	GET *API-URL*/admin/extension/providerVdcReferences	None	`VMWProviderVdcReferences`
Retrieve a list of external networks in a cloud.	GET *API-URL*/admin/extension/externalNetworkReferences	None	`VMWExternalNetworkReferences`
Retrieve a list of network pools in a cloud.	GET *API-URL*/admin/extension/networkPoolReferences	None	`VMWNetworkPoolReferences`
Retrieve a list of vCenter servers registered to a cloud.	GET *API-URL*/admin/extension/vimServerReferences	None	`VMWVimServerReferences`
Retrieve information about a vCenter server.	GET *API-URL*/admin/extension/vimServer/*id*	None	`VimServer`
Retrieve a list of available resource pools on a vCenter server.	GET *API-URL*/admin/extension/vimServer/*id*/resourcePoolList	None	`ResourcePoolList`
Update vCenter server settings.	PUT *API-URL*/admin/extension/vimServer/*id*	`VimServer`	`Task`
Register a vCenter server and vShield manager.	POST *API-URL*/admin/extension/action/registervimserver	`RegisterVimServerParams`	`RegisterVimServerParams`
Unregister a vCenter server and vShield manager.	POST *API-URL*/admin/extension/action/unregister	None	`Task`
Force reconnection to a vCenter server.	POST *API-URL*/admin/extension/vimServer/*id*/action/forcevimserverreconnect	None	`Task`
Retrieve a list of ESX/ESXi hosts in a cloud.	GET *API-URL*/admin/extension/hostReferences	None	`VMWHostReferences`
Retrieve information about a host.	GET *API-URL*/admin/extension/host/*id*	None	`Host`
Prepare a host.	POST *API-URL*/admin/extension/host/*id*/action/prepare	`PrepareHostParams`	`Task`
Unprepare a host.	POST *API-URL*/admin/extension/host/*id*/action/unprepare	None	`Task`
Enable a host.	POST *API-URL*/admin/extension/host/*id*/action/enable	None	`Task`
Disable a host.	POST *API-URL*/admin/extension/host/*id*/action/disable	None	`Task`

Table 7-1. Summary of vSphere Platform Extension Requests (Continued)

Operation	Request	Request Body	Response
Repair a host.	POST *API-URL*/admin/extension/host/*id*/action/repair	None	204 No Content
Upgrade a host.	POST *API-URL*/admin/extension/host/*id*/action/upgrade	None	`Task`
Create a provider vDC.	POST *API-URL*/admin/extension/providervdcs	`VMWProviderVdc`	`VMWProviderVdc`
Retrieve a provider vDC.	GET *API-URL*/admin/extension/providervdc/*id*	None	`VMWProviderVdc`
Update a provider vDC.	PUT *API-URL*/admin/extension/providervdc/*id*	`VMWProviderVdc`	`VMWProviderVdc`
Enable a provider vDC.	POST *API-URL*/admin/extension/providervdc/*id*/action/enable	None	204 No Content
Disable a provider vDC.	POST *API-URL*/admin/extension/providervdc/*id*/action/disable	None	204 No Content
Delete a provider vDC.	DELETE *API-URL*/admin/extension/providervdc/*id*	None	`Task`
Create an external network.	POST *API-URL*/admin/extension/externalnets	`VMWExternalNetwork`	`VMWExternalNetwork`
Retrieve an external network.	GET *API-URL*/admin/extension/externalnet/*id*	None	`VMWExternalNetwork`
Update an external network.	PUT *API-URL*/admin/extension/externalnet/*id*	`VMWExternalNetwork`	`VMWExternalNetwork`
Delete an external network.	DELETE *API-URL*/admin/extension/externalnet/*id*	None	`Task`
Create a network pool.	POST *API-URL*/admin/extension/networkPools	`VMWNetworkPool`	`VMWNetworkPool`
Retrieve a network pool.	GET *API-URL*/admin/extension/networkPool/*id*	None	`VMWNetworkPool`
Update a network pool.	PUT *API-URL*/admin/extension/networkPool/*id*	`VMWNetworkPool`	`VMWNetworkPool`
Delete a network pool.	DELETE *API-URL*/admin/extension/networkPool/*id*	None	`Task`
Import a virtual machine from vCenter as a vApp.	POST *API-URL*/admin/extension/vimServer/*id*/importVmAsVapp	`ImportVmAsVAppParams`	`VApp`

Table 7-1. Summary of vSphere Platform Extension Requests (Continued)

Operation	Request	Request Body	Response
Import a virtual machine from vCenter as a vApp template.	POST *API-URL*/admin/extension/vimServer/*id*/importVmAsVappTemplate	`ImportVmAsVAppTemplateParams`	`VAppTemplate`
Retrieve the representation of a datastore. [NEW]	GET *API-URL*/admin/extension/datastore/*id*	None	`Datastore`
Enable a datastore. [NEW]	POST *API-URL*/admin/extension/datastore/*id*/action/enable	None	`Datastore`
Disable a datastore. [NEW]	POST *API-URL*/admin/extension/datastore/*id*/action/disable	None	`Datastore`
Update a datastore. [NEW]	PUT *API-URL*/admin/extension/datastore/*id*	`Datastore`	`Datastore`
Delete a datastore. [NEW]	DELETE *API-URL*/admin/extension/datastore/*id*	None	204 No Content
Import a virtual machine from vCenter into an existing vApp. [NEW]	POST *API-URL*/admin/extension/vimServer/*id*/importVmIntoExistingVApp	`ImportVmIntoExistingVAppParams`	`Task`
Import a media image from vCenter. [NEW]	POST *API-URL*/admin/extension/vimServer/*id*/importMedia	`ImportMediaParams`	`Media`
Enable a resource pool. [NEW]	POST *API-URL*/admin/extension/resourcePool/*id*/action/enable	None	204 No Content
Disable a resource pool. [NEW]	POST *API-URL*/admin/extension/resourcePool/*id*/action/disable	None	204 No Content
Place a vApp in maintenance mode to prevent users from changing vApp metadata. [NEW]	POST *API-URL*/vApp/*id*/action/enterMaintenanceMode	None	204 No Content
Remove a vApp from maintenance mode. [NEW]	POST *API-URL*/vApp/*id*/action/exitMaintenanceMode	None	204 No Content
Relocate a virtual machine in a vApp template to a different datastore. [NEW]	POST *API-URL*/vAppTemplate/vm-*id*/action/relocate	`RelocateParams`	`Task`
Relocate a virtual machine to a different datastore. [NEW]	POST *API-URL*/vApp/vm-*id*/action/relocate	`RelocateParams`	`Task`

Retrieve or Update System Settings

A system administrator can retrieve or update a cloud-wide set of system properties. These properties specify default values and behaviors for the cloud and all of the organizations in it.

The `SystemSettings` element includes all system settings for this cloud. The element also includes links that allow you to retrieve these subsidiary elements, which define specific categories of settings.

GeneralSettings Control the configuration and behavior of the entire cloud.

NotificationsSettings Control the vCloud Director AMQP notifications service.

LdapSettings Specify details of the system LDAP directory service.

AmqpSettings Specify credentials and connection information for the AMQP broker that handles notifications and blocking task messages.

EmailSettings Define configuration and connection parameters for the system default email service, and specifies properties of email alerts that the system sends.

License System license serial number and related settings.

BrandingSettings Allow you to customize the branding of the vCloud Director client UI and some of the links that appear on the vCloud Director Home login screen.

BlockingTaskSettings Control the behavior of blocking tasks and enable blocking extensions for specific kinds of tasks.

PasswordPolicySettings Specify default policies to be followed when a user in any organization enters an incorrect password. Organization administrators can override this default for their organization.

You can retrieve the entire `SystemSettings` element to view all of these settings. To update an individual subsection, retrieve it with a GET request, modify it, and update it with a PUT request.

Prerequisites

Verify that you are logged in as a system administrator.

Procedure

1. Retrieve the `SystemSettings` element.

   Use a request like this one.

   ```
   GET https://vcloud.example.com/api/admin/extension/settings
   ```

2. Examine the response to locate the `Link` elements that you can use to retrieve an individual subsection.

   These links have a `rel` attribute value of `down`.

3. Use the link to retrieve the subsection.

   Make a GET request to the `href` value of the link.

4. (Optional) Modify the retrieved subsection.

   Subsections that you can modify include a link where `rel="edit"`.

5. (Optional) To update the subsection, PUT the modified subsection to the `href` of the link described in Step 4.

Attach a vCenter Server

A system administrator can register a vCenter server and a companion vShield Manager server for use in a cloud by making a POST request to the cloud's `action/registervimserver` URL and supplying a `RegisterVimServerParams` request body

Prerequisites

- Verify that you are logged in as a system administrator.
- Verify that you know the name, IP address, and administrator password of the vCenter server and vShield Manager server.

Procedure

1 Retrieve the XML representation of the vSphere platform extensions.

Use a request like this one.

```
GET https://vcloud.example.com/api/admin/extension
```

2 Examine the response to locate the `Link` element that contains the URL for adding vCenter servers to the cloud.

This element has a `rel` attribute value of `add` and a `type` attribute value of `application/vnd.vmware.admin.registerVimServerParams+xml`, as shown here:

```
<Link
   type="application/vnd.vmware.admin.registerVimServerParams+xml"
   rel="add"
   href="https://vcloud.example.com/api/admin/extension/action/registervimserver"/>
```

3 Create a `RegisterVimServerParams` element that includes the information required to register the vCenter server and vShield manager.

4 POST the `RegisterVimServerParams` element you created in Step 3 to the URL described in Step 2.

See the request portion of "Example: Register a vCenter Server and vShield Manager," on page 166.

Example: Register a vCenter Server and vShield Manager

You must supply the user name and password of the vCenter administrator in the request. The response includes vCloud URLs for the newly registered vCenter and vShield Manager servers, and omits the password.

Request:

```
POST https://vcloud.example.com/api/admin/extension/action/registervimserver
Content-Type: application/vnd.vmware.admin.registerVimServerParams+xml
...
<?xml version="1.0" encoding="UTF-8"?>
<vmext:RegisterVimServerParams
   xmlns:vmext="http://www.vmware.com/vcloud/extension/v1.5"
   xmlns:vcloud="http://www.vmware.com/vcloud/v1.5">
   <vmext:VimServer
      name="VC-22">
      <vmext:Username>Administrator</vmext:Username>
      <vmext:Password>Pa55w0rd</vmext:Password>
      <vmext:Url>https://10.100.121.123:443</vmext:Url>
      <vmext:IsEnabled>false</vmext:IsEnabled>
   </vmext:VimServer>
   <vmext:ShieldManager
```

```
      name="VSM-VC-22">
      <vmext:Username>Administrator</vmext:Username>
      <vmext:Password>Pa55w0rd</vmext:Password>
      <vmext:Url>https://10.100.121.66</vmext:Url>
   </vmext:ShieldManager>
</vmext:RegisterVimServerParams>
```

Response:

```
200 OK
Content-Type: application/vnd.vmware.admin.registerVimServerParams+xml
...
<vmext:RegisterVimServerParams
   xmlns:vmext="http://www.vmware.com/vcloud/extension/v1.5"
   xmlns:vcloud="http://www.vmware.com/vcloud/v1.5">
   <vmext:VimServer
      type="application/vnd.vmware.admin.vmwvirtualcenter+xml"
      name="VC-22"
      href="https://vcloud.example.com/api/admin/extension/vimServer/100">
      ...
   </vmext:VimServer>
   <vmext:ShieldManager
      ...
   </vmext:ShieldManager>
</vmext:RegisterVimServerParams>
```

Finding Available vCenter Resources

Many of the operations required to import virtual machines or create Provider vDCs, external networks, and network pools require you to identify vCenter resources and obtain references to them. You use these references to make the vCenter resources available in the cloud.

Every vCenter server registered to your cloud is represented as a `VimServerReference` element in the cloud's `vimServerReferences` list. You can retrieve one of these references to get a detailed representation of the server object, including links to the server's resource pools, networks, ESX/ESXi hosts, and virtual machines.

Prerequisites

Verify that you are logged in as a system administrator.

Procedure

1 Retrieve the list of vCenter servers registered to this cloud.

   Use a request like this one.

   ```
   GET https://vcloud.example.com/api/admin/extension/vimServerReferences
   ```

2 Retrieve the representation of a vCenter server.

   The response to the request you made in Step 1 contains a list of `VimServerReference` elements. You can make a GET request to any of these references to retrieve the XML representation of a vCenter server registered to this cloud.

The `VimServer` element returned in response to the request you made in Step 2 includes several `Link` elements where `rel="down"`. These links contain URLs that you can use to retrieve lists of references to vCenter resources on this server.

- To see a list of references to the available resource pools, retrieve the `resourcePoolList` link.
- To see a list of references to the available networks, retrieve the `networks` link.

- To see a list of references to ESX/ESXi hosts managed by this vCenter server, retrieve the `hostReferences` link.
- To see a list of references to virtual machines on ESX/ESXi hosts managed by this vCenter server, retrieve the `vmsList` link.

Example: Resources on a vCenter Server

Request:

```
GET https://vcloud.example.com/api/admin/extension/vimServer/9
```

Response:

```
200 OK
...
<vmext:VimServer
   xmlns:vmext="http://www.vmware.com/vcloud/extension/v1.5"
   xmlns:vcloud="http://www.vmware.com/vcloud/v1.5"
   name="VC22"
   id="urn:vcloud:vimserver:9"
   type="application/vnd.vmware.admin.vmwvirtualcenter+xml"
   href="https://vcloud.example.com/api/admin/extension/vimServer/9" ...>
   ...
   <vcloud:Link
      rel="down"
      type="application/vnd.vmware.admin.vmsObjectRefsList+xml"
      href="https://vcloud.example.com/api/admin/extension/vimServer/9/vmsList" />
   <vcloud:Link
      rel="down"
      type="application/vnd.vmware.admin.vmwHostReferences+xml"
      href="https://vcloud.example.com/api/admin/extension/vimServer/9/hostReferences" />
   <vcloud:Link
      rel="down"
      type="application/vnd.vmware.admin.resourcePoolList+xml"
      href="https://vcloud.example.com/api/admin/extension/vimServer/9/resourcePoolList" />
    <vcloud:Link
      rel="down"
      type="application/vnd.vmware.admin.vimServerNetworks+xml"
      href="https://vcloud.example.com/api/admin/extension/vimServer/9/networks" />
   ...
   <vmext:Username>administrator</vmext:Username>
   <vmext:Url>https://10.115.124.37:443</vmext:Url>
   <vmext:IsEnabled>true</vmext:IsEnabled>
   <vmext:IsConnected>true</vmext:IsConnected>
   <vmext:ShieldManagerHost>cloud-vsla-e-lm-124-82.example.com</vmext:ShieldManagerHost>
   <vmext:ShieldManagerUserName>admin</vmext:ShieldManagerUserName>
</vmext:VimServer>
```

Retrieve a List of Resource Pools from a vCenter Server

You can retrieve the list of resource pools available on a vCenter server registered to a cloud. To retrieve the list, you make a GET request to the server's `resourcePoolList` link.

The `ResourcePoolList` of a `VimServer` element contains an entry for every available resource pool on the server. Resource pools that a provider vDC is already using are not listed, because they are considered unavailable. See “Finding Available vCenter Resources,” on page 167.

Prerequisites

- Verify that you are logged in as a system administrator.
- Retrieve the XML representation of a vCenter server registered to your cloud. See "Finding Available vCenter Resources," on page 167.

Procedure

1 Examine the `VimServer` element to locate its `resourcePoolList` link.

The link has the following form:

```
<vcloud:Link
      rel="down"
      type="application/vnd.vmware.admin.resourcePoolList+xml"
      href="https://vcloud.example.com/api/admin/extension/vimServer/9/resourcePoolList" />
```

2 Retrieve the list of resource pools.

See "Example: Retrieve a List of Resource Pools from a vCenter Server," on page 169. If the list is empty, all resource pools on the server are already in use.

Example: Retrieve a List of Resource Pools from a vCenter Server

Request:

```
GET https://vcloud.example.com/api/admin/extension/vimServer/9/resourcePoolList
```

Response:

```
200 OK
Content-Type: application/vnd.vmware.admin.resourcepoollist+xml
...
<vmext:ResourcePoolList
   xmlns:vmext="http://www.vmware.com/vcloud/extension/v1.5"
   xmlns:vcloud="http://www.vmware.com/vcloud/v1.5"
   type="application/vnd.vmware.admin.resourcePoolList+xml" ... >
   <vcloud:Link
      rel="up"
      type="application/vnd.vmware.admin.vmwvirtualcenter+xml"
      href="https://vcloud.example.com/api/admin/extension/vimServer/9" />
   <vmext:ResourcePool
      name="cluster2">
      <vmext:MoRef>resgroup-195</vmext:MoRef>
      <vmext:VimObjectType>RESOURCE_POOL</vmext:VimObjectType>
      <vmext:DataStoreRefs>
         <vmext:VimObjectRef>
            <vmext:VimServerRef
               type="application/vnd.vmware.admin.vmwvirtualcenter+xml"
               name="vc9-ds1"
               href="https://vcloud.example.com/api/admin/extension/vimServer/9" />
            <vmext:MoRef>datastore-172</vmext:MoRef>
            <vmext:VimObjectType>DATASTORE</vmext:VimObjectType>
         </vmext:VimObjectRef>
         <vmext:VimObjectRef>
            <vmext:VimServerRef
               type="application/vnd.vmware.admin.vmwvirtualcenter+xml"
               name="vc9-ds2"
               href="https://vcloud.example.com/api/admin/extension/vimServer/9" />
            <vmext:MoRef>datastore-173</vmext:MoRef>
            <vmext:VimObjectType>DATASTORE</vmext:VimObjectType>
         </vmext:VimObjectRef>
       </vmext:DataStoreRefs>
   </vmext:ResourcePool>
   <vmext:ResourcePool
      name="cluster3">
       <vmext:MoRef>resgroup-230</vmext:MoRef>
      <vmext:VimObjectType>RESOURCE_POOL</vmext:VimObjectType>
      <vmext:DataStoreRefs>
         <vmext:VimObjectRef>
            <vmext:VimServerRef
               type="application/vnd.vmware.admin.vmwvirtualcenter+xml"
               name="vc9-ds10"
               href="https://vcloud.example.com/api/admin/extension/vimServer/9" />
            <vmext:MoRef>datastore-174</vmext:MoRef>
            <vmext:VimObjectType>DATASTORE</vmext:VimObjectType>
         </vmext:VimObjectRef>
      </vmext:DataStoreRefs>
   </vmext:ResourcePool>
   ...
</vmext:ResourcePoolList>
```

Retrieve a List of Network Resources from a vCenter Server

You can retrieve the list of network resources such as portgroups and switches that are available on a vCenter server registered to a cloud. To retrieve the list, make a GET request to the server's `networks` link.

Retrieving the `networks` link from a `VimServer` element returns a `VimObjectRefList` element that contains references to available `DV_SWITCH` and `DV_PORTGOUP` objects on the server. `DV_SWITCH` and `DV_PORTGOUP` objects that a provider network is using are not listed, because they are considered unavailable. See "Finding Available vCenter Resources," on page 167.

Prerequisites

Verify that you are logged in as a system administrator.

Procedure

1. Retrieve the XML representation of a vCenter server registered to your cloud.

2. Examine the response, a `VimServer` element, to locate the `networks` link.

   This link has the following form:

   ```
   <vcloud:Link
         rel="down"
         type="application/vnd.vmware.admin.vimServerNetworks+xml"
         href="https://vcloud.example.com/api/admin/extension/vimServer/9/networks" />
   ```

3. Retrieve the list of network resources.

   See "Example: Retrieve a List of Network Resources from a vCenter Server," on page 170. If the list is empty, all network resources on the server are already in use.

Example: Retrieve a List of Network Resources from a vCenter Server

Request:

```
GET https://vcloud.example.com/api/admin/extension/vimServer/9/networks
```

Response:

```
200 OK
Content-Type: application/vnd.vmware.admin.vimservernetworks+xml
...
<vmext:VimObjectRefList
   xmlns:vmext="http://www.vmware.com/vcloud/extension/v1.5"
   xmlns:vcloud="http://www.vmware.com/vcloud/v1.5"
   xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" ... >
   <vcloud:Link
      rel="up"
      type="application/vnd.vmware.admin.vmwvirtualcenter+xml"
      href="https://vcloud.example.com/api/admin/extension/vimServer/9" />
   <vmext:VimObjectRefs>
      <vmext:VimObjectRef>
         <vmext:VimServerRef
            type="application/vnd.vmware.admin.vmwvirtualcenter+xml"
            name="vc9"
            href="https://vcloud.example.com/api/admin/extension/vimServer/9" />
         <vmext:MoRef>dvportgroup-32</vmext:MoRef>
         <vmext:VimObjectType>DV_PORTGROUP</vmext:VimObjectType>
      </vmext:VimObjectRef>
      <vmext:VimObjectRef>
         <vmext:VimServerRef
            type="application/vnd.vmware.admin.vmwvirtualcenter+xml"
            name=""
            href="https://vcloud.example.com/api/admin/extension/vimServer/9" />
         <vmext:MoRef>network-175</vmext:MoRef>
         <vmext:VimObjectType>NETWORK</vmext:VimObjectType>
      </vmext:VimObjectRef>
   </vmext:VimObjectRefs>
</vmext:VimObjectRefList>
```

Retrieve a List of Virtual Machines from a vCenter Server

You can retrieve the list of virtual machines in the inventory of a vCenter server that is registered to a cloud. To retrieve the list, make a GET request to the server's `vmslist` link.

When you import a virtual machine from vCenter, your request must supply a reference to the vCenter server and a VIM object reference to the virtual machine. See "Finding Available vCenter Resources," on page 167.

Prerequisites

Verify that you are logged in as a system administrator.

Procedure

1 Retrieve the XML representation of a vCenter server registered to your cloud.

2 Examine the response, a `VimServer` element, to locate the `vmsList` link.

This link has the following form:

```
<vcloud:Link
      rel="down"
      type="application/vnd.vmware.admin.vmsObjectRefsList+xml"
      href="https://vcloud.example.com/api/admin/extension/vimServer/9/vmsList" />
```

3 Retrieve the list of virtual machines.

See the request portion of "Example: Retrieve a List of Virtual Machines from a vCenter Server," on page 172. If the list is empty, no virtual machines are in the server's inventory.

Example: Retrieve a List of Virtual Machines from a vCenter Server

Request:

```
GET https://vcloud.example.com/api/admin/extension/vimServer/9/vmsList
```

Response:

```
200 OK
Content-Type: application/vnd.vmware.admin.vmsobjectrefslist+xml
...
<vmext:VmObjectRefsList
   xmlns:vmext="http://www.vmware.com/vcloud/extension/v1.5"
   page="1"
   numberOfPages="1"
   xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" ... >
   <vmext:VmObjectRef
      name="RH5u3_32bit">
      <vmext:VimServerRef
         type="application/vnd.vmware.admin.vmwvirtualcenter+xml"
         name="vc2-v41u1"
         href="https://vcloud.example.com/api/admin/extension/vimServer/9" />
      <vmext:MoRef>vm-41</vmext:MoRef>
      <vmext:VimObjectType>VIRTUAL_MACHINE</vmext:VimObjectType>
   </vmext:VmObjectRef>
   <vmext:VmObjectRef
      name="W2K3 64 R2">
      <vmext:VimServerRef
         type="application/vnd.vmware.admin.vmwvirtualcenter+xml"
         name="vc2-v41u1"
         href="https://vcloud.example.com/api/admin/extension/vimServer/9" />
      <vmext:MoRef>vm-43</vmext:MoRef>
      <vmext:VimObjectType>VIRTUAL_MACHINE</vmext:VimObjectType>
   </vmext:VmObjectRef>
   <vmext:VmObjectRef
      name="Ubuntu91_32_vt4">
      <vmext:VimServerRef
         type="application/vnd.vmware.admin.vmwvirtualcenter+xml"
         name="vc2-v41u1"
         href="https://vcloud.example.com/api/admin/extension/vimServer/9" />
      <vmext:MoRef>vm-44</vmext:MoRef>
      <vmext:VimObjectType>VIRTUAL_MACHINE</vmext:VimObjectType>
   </vmext:VmObjectRef>
</vmext:VmObjectRefsList>
```

Create a Provider vDC

A Provider vDC is a collection of compute, memory, and storage resources from one vCenter. A system administrator can make resources from a provider vDC available through organization vDCs in the cloud.

A Provider vDC is represented as a `VMWProviderVdc` element in the extension view and a `ProviderVdc` element in the admin view. A system administrator can create a `VMWProviderVdc` or modify it to add or remove datastores and resource pools, or change other properties such as its description. A system administrator cannot change the primary resource pool or vCenter server that were specified at creation time. An organization administrator can retrieve a read-only representation of a provider vDC in a `ProviderVdc` element.

For information about choosing a resource pool, see "Retrieve a List of Resource Pools from a vCenter Server," on page 168. For information about adding or removing a resource pool, see "Add Resource Pools to a Provider vDC," on page 178.

Prerequisites

Verify that you are logged in as a system administrator.

Procedure

1 Retrieve the XML representation of the vSphere platform extensions.

Use a request like this one.

```
GET https://vcloud.example.com/api/admin/extension
```

2 Examine the response to locate the `Link` element that contains the URL for adding provider vDCs to the cloud.

This element has a `rel` attribute value of `add` and a `type` attribute value of `application/vnd.vmware.admin.vmwprovidervdc+xml`, as shown here:

```
<Link
   type="application/vnd.vmware.admin.vmwprovidervdc+xml"
   rel="add"
   href="https://vcloud.example.com/api/admin/extension/providervdcs"/>
```

3 Choose a vCenter server to provide a resource pool and datastores.

4 Create a `VMWProviderVdc` element that specifies the properties of the provider vDC.

In the `DataStoreRefs` and `ResourcePoolRef` elements of the `VMWProviderVdc`, use the `href` attribute value of the `VimServer` you selected in Step 3. Use the `MoRef` and `VimObjectType` values of the resource pool and datastores as they appear in the `ResourcePool` element from the resource pool list. See the request portion of "Example: Create a Provider vDC," on page 174.

NOTE You must specify exactly one resource pool when you create the provider vDC. You can add more resource pools after the provider vDC is created.

5 POST the `VMWProviderVdc` element you created in Step 4 to the URL described in Step 2.

See the request portion of "Example: Create a Provider vDC," on page 174.

The server creates and enables the provider vDC and returns a `VMWProviderVdc` element that includes the contents you POSTed, along with a set of `Link` elements that you can use to access, remove, disable, or modify the provider vDC. The new provider vDC becomes a member of the `ProviderVdcReferences` element of the `VCloud`. The resource pool you selected is removed from the resource pool list of the vCenter server.

Example: Create a Provider vDC

This request creates a small Provider vDC using a resource pool and datastore extracted from the response portion of "Example: Retrieve a List of Resource Pools from a vCenter Server," on page 169. In practice, the DataStoreRefs element typically includes additional VimObjectRef elements that reference other datastores available on the specified VimServer. The vCenter server that provides the resources is referenced in the VimServerRef and VimServer elements.

Request:

```
POST https://vcloud.example.com/api/admin/extension/providervdcs
Content-Type: application/vnd.vmware.admin.vmwprovidervdc+xml
...
<?xml version="1.0" encoding="UTF-8"?>
<vmext:VMWProviderVdc
   xmlns:vmext="http://www.vmware.com/vcloud/extension/v1.5"
   xmlns:vcloud="http://www.vmware.com/vcloud/v1.5"
   name="PvDC-Example"
   type="application/vnd.vmware.admin.vmwprovidervdc+xml">
   <vcloud:Description>Example Provider vDC</vcloud:Description>
   <vmext:DataStoreRefs>
      <vmext:VimObjectRef>
         <vmext:VimServerRef
            type="application/vnd.vmware.admin.vmwvirtualcenter+xml"
            href="https://vcloud.example.com/api/admin/extension/vimServer/9" />
         <vmext:MoRef>datastore-173</vmext:MoRef>
         <vmext:VimObjectType>DATASTORE</vmext:VimObjectType>
      </vmext:VimObjectRef>
   </vmext:DataStoreRefs>
   <vmext:ResourcePoolRefs>
      <vmext:VimObjectRef>
         <vmext:VimServerRef
            type="application/vnd.vmware.admin.vmwvirtualcenter+xml"
            href="https://vcloud.example.com/api/admin/extension/vimServer/9" />
         <vmext:MoRef>resgroup-195</vmext:MoRef>
         <vmext:VimObjectType>RESOURCE_POOL</vmext:VimObjectType>
      </vmext:VimObjectRef>
   </vmext:ResourcePoolRefs>
   <vmext:VimServer
      type="application/vnd.vmware.admin.vmwvirtualcenter+xml"
      href="https://vcloud.example.com/api/admin/extension/vimServer/9" />
</vmext:VMWProviderVdc>
```

The response includes a Task that tracks the creation of the Provider vDC, and a set of Link elements that you can use to operate on or modify the Provider vDC. It also includes read-only values for ComputeCapacity, StorageCapacity, and HighestSupportedHardwareVersion that were derived from the vCenter, datastores, and resource pool specified in the request, and a list of HostReferences identifying the ESX hosts that provide the resources.

Response:

```
201 Created
Content-Type: application/vnd.vmware.admin.vmwprovidervdc+xml
...
<vmext:VMWProviderVdc
   xmlns:vmext="http://www.vmware.com/vcloud/extension/v1.5"
   xmlns:vcloud="http://www.vmware.com/vcloud/v1.5"
   status="0"
   name="PvDC-Example"
   id="urn:vcloud:providervdc:35"
   type="application/vnd.vmware.admin.vmwprovidervdc+xml"
   href="https://vcloud.example.com/api/admin/extension/providervdc/35"
   ... >
   <vcloud:Link
      rel="up"
      type="application/vnd.vmware.admin.vmwProviderVdcReferences+xml"
      href="https://vcloud.example.com/api/admin/extension/providerVdcReferences" />
   <vcloud:Link
      rel="enable"
      href="https://vcloud.example.com/api/admin/extension/providervdc/35/action/enable" />
   <vcloud:Link
      rel="edit"
      type="application/vnd.vmware.admin.vmwprovidervdc+xml"
      href="https://vcloud.example.com/api/admin/extension/providervdc/35" />
   <vcloud:Link
      rel="remove"
      href="https://vcloud.example.com/api/admin/extension/providervdc/35" />
   <vcloud:Link
      rel="alternate"
      type="application/vnd.vmware.admin.providervdc+xml"
      href="https://vcloud.example.com/api/admin/providervdc/35" />
   <vcloud:Link
     rel="down"
     type="application/vnd.vmware.admin.vmwProviderVdcResourcePoolSet+xml"
     href="https://vcloud.example.com/api/admin/extension/providervdc/35/resourcePools"/>
   <vcloud:Link
      rel="down"
      type="application/vnd.vmware.admin.vdcReferences+xml"
      href="https://vcloud.example.com/api/admin/providervdc/35/vdcReferences" />
   <vcloud:Link
     rel="update:resourcePools"
     type="application/vnd.vmware.admin.vmwProviderVdcResourcePoolSet+xml"

href="https://vcloud.example.com/api/admin/extension/providervdc/35/action/updateResourcePools"/>
   <vcloud:Description>Example Provider vDC</vcloud:Description>
   <vcloud:Tasks>
      <vcloud:Task
         status="running"
         ...
         operation="Creating Provider Virtual Datacenter PvDC-Example(35)"
         ...
      </vcloud:Task>
   </vcloud:Tasks>
   <vcloud:ComputeCapacity>
      ...
   </vcloud:ComputeCapacity>
   <vcloud:StorageCapacity>
      ...
   </vcloud:StorageCapacity>
   <vcloud:AvailableNetworks />
   <vcloud:IsEnabled>false</vcloud:IsEnabled>
   <vcloud:NetworkPoolReferences />
```

```
      <vmext:DataStoreRefs>
         <vmext:VimObjectRef>
            <vmext:VimServerRef
               type="application/vnd.vmware.admin.vmwvirtualcenter+xml"
               name="ds1"
               href="https://vcloud.example.com/api/admin/extension/vimServer/9" />
            <vmext:MoRef>datastore-173</vmext:MoRef>
            <vmext:VimObjectType>DATASTORE</vmext:VimObjectType>
         </vmext:VimObjectRef>
      </vmext:DataStoreRefs>
      <vmext:ResourcePoolRef>
         <vmext:VimServerRef
            type="application/vnd.vmware.admin.vmwvirtualcenter+xml"
            name="vc-a"
            href="https://vcloud.example.com/api/admin/extension/vimServer/9" />
         <vmext:MoRef>resgroup-195</vmext:MoRef>
         <vmext:VimObjectType>RESOURCE_POOL</vmext:VimObjectType>
      </vmext:ResourcePoolRef>
      <vmext:VimServer
         type="application/vnd.vmware.admin.vmwvirtualcenter+xml"
         name="vc-a"
         href="https://vcloud.example.com/api/admin/extension/vimServer/9" />
      <vmext:HostReferences>
         <vmext:HostReference
            type="application/vnd.vmware.admin.host+xml"
            name="ESX01.vcloud.example.com"
            href="https://vcloud.example.com/api/admin/extension/host/83" />
         <vmext:HostReference
            type="application/vnd.vmware.admin.host+xml"
            name="ESX02.vcloud.example.com"
            href="https://vcloud.example.com/api/admin/extension/host/261" />
      </vmext:HostReferences>
      <vcloud:HighestSupportedHardwareVersion>vmx-08</vcloud:HighestSupportedHardwareVersion>
   </vmext:VMWProviderVdc>
```

Retrieve a Provider vDC Resource Pool Set

The VMWProviderVdcResourcePoolSet of a provider vDC contains information about all of the provider vDC's resource pools. Getting this information is usually a prerequisite to adding or removing a resource pool.

Each reference in a VMWProviderVdcResourcePoolSet lists the vCenter server that provides the resource pool and indicates whether the resource pool is primary. All resource pools in a VMWProviderVdcResourcePoolSet must come from the same vCenter server.

Prerequisites

Verify that you are logged in as a system administrator.

Procedure

1 Retrieve the XML representation of the provider vDC.

Use a request like this one:

```
GET https://vcloud.example.com/api/admin/extension/providervdc/35
```

2 Locate the resourcePools link in the VMWProviderVdc.

Every VMWProviderVdc element includes a link like this one to the provider vDC's resource pools.

```
<Link
     rel="down"
     type="application/vnd.vmware.admin.vmwProviderVdcResourcePoolSet+xml"
     href="https://vcloud.example.com/api/admin/extension/providervdc/35/resourcePools"/>
```

3 Retrieve the VMWProviderVdcResourcePoolSet for the provider vDC.

See "Example: Retrieve a Resource Pool Set," on page 177.

Example: Retrieve a Resource Pool Set

This example lists the resource pools for the provider vDC created in "Example: Create a Provider vDC," on page 174. The response is a VMWProviderVdcResourcePoolSet that contains two resource pools, one of which is designated primary. Both reference the same vCenter server at https://vcloud.example.com/api/admin/extension/vimServer/9.

Request:

```
GET https://vcloud.example.com/api/admin/extension/providervdc/35/resourcePools
```

Response:

```
200 OK
Content-Type: application/vnd.vmware.admin.vmwprovidervdcresourcepoolset+xml
...
<vmext:VMWProviderVdcResourcePoolSet
   xmlns:vmext="http://www.vmware.com/vcloud/extension/v1.5"
   xmlns:vcloud="http://www.vmware.com/vcloud/v1.5" ... >
   <vcloud:Link
      rel="up"
      type="application/vnd.vmware.admin.vmwprovidervdc+xml"
      href="https://vcloud.example.com/api/admin/extension/providervdc/35" />
   <vmext:VMWProviderVdcResourcePool>
      <vcloud:Link
         rel="disable"
         href="https://vcloud.example.com/api/admin/extension/resourcePool/64/action/disable" />
      <vmext:ResourcePoolVimObjectRef>
         <vmext:VimServerRef
            type="application/vnd.vmware.admin.vmwvirtualcenter+xml"
            href="https://vcloud.example.com/api/admin/extension/vimServer/9" />
         <vmext:MoRef>resgroup-195</vmext:MoRef>
         <vmext:VimObjectType>RESOURCE_POOL</vmext:VimObjectType>
      </vmext:ResourcePoolVimObjectRef>
     <vmext:ResourcePoolRef
         type="application/vnd.vmware.admin.vmwProviderVdcResourcePool+xml"
         href="https://vcloud.example.com/api/admin/extension/resourcePool/64 />
      <vmext:Enabled>true</vmext:Enabled>
      <vmext:Primary>true</vmext:Primary>
   </vmext:VMWProviderVdcResourcePool>
   <vmext:VMWProviderVdcResourcePool>
      <vcloud:Link
         rel="disable"
         href="https://vcloud.example.com/api/admin/extension/resourcePool/66/action/disable" />
      <vmext:ResourcePoolVimObjectRef>
         <vmext:VimServerRef
```

```
            type="application/vnd.vmware.admin.vmwvirtualcenter+xml"
            href="https://vcloud.example.com/api/admin/extension/vimServer/9" />
         <vmext:MoRef>resgroup-235</vmext:MoRef>
         <vmext:VimObjectType>RESOURCE_POOL</vmext:VimObjectType>
      </vmext:ResourcePoolVimObjectRef>
     <vmext:ResourcePoolRef
         type="application/vnd.vmware.admin.vmwProviderVdcResourcePool+xml"
         href="https://vcloud.example.com/api/admin/extension/resourcePool/66 />
      <vmext:Enabled>true</vmext:Enabled>
      <vmext:Primary>false</vmext:Primary>
   </vmext:VMWProviderVdcResourcePool>
</vmext:VMWProviderVdcResourcePoolSet>
```

Add Resource Pools to a Provider vDC

A system administrator can add secondary resource pools to an existing Provider vDC. Adding resource pools allows organization vDCs that reference the Provider vDC to provide additional resources.

When you create a Provider vDC, it initially contains one resource pool, called the primary resource pool. Adding secondary resource pools allows a Provider vDC to support additional resource elasticity in all `AllocationVApp` (pay as you go) organization vDCs that it supports. Resource elasticity in an organization vDC means that the vDC's capacity to support compute resources can grow or shrink on demand.

Secondary resource pools must come from the vCenter server that provides the primary resource pool. See "Retrieve a List of Resource Pools from a vCenter Server," on page 168.

Prerequisites

Verify that you are logged in as a system administrator.

Procedure

1 Retrieve the XML representation of the Provider vDC.

Use a request like this one:

```
GET https://vcloud.example.com/api/admin/extension/providervdc/35
```

2 Locate the `updateResourcePools` link in the `VMWProviderVdc`.

Every `VMWProviderVdc` element includes an action link like this one to the Provider vDC's `updateResourcePools` action.

```
<Link
     rel="update:resourcePools"
     type="application/vnd.vmware.admin.vmwProviderVdcResourcePoolSet+xml"

href="https://vcloud.example.com/api/admin/extension/providervdc/35/action/updateResourcePool
s"/>
```

3 Retrieve the resource pool list from the Provider vDC.

The `VMWProviderVdcResourcePoolSet` contains references to the Provider vDC's existing resource pools and the vCenter server that hosts them.

4 Choose another resource pool from the same vCenter server.

5 Create an `UpdateResourcePoolSetParams` element that contains an `AddItem` element for each resource pool to add.

6 POST the `UpdateResourcePoolSetParams` element you created in Step 5 to the Provider vDC's `resourcePools` link shown in Step 2.

Example: Add Resource Pools to a Provider vDC

This example adds a resource pool to the Provider vDC created in "Example: Create a Provider vDC," on page 174. The additional resource pool is hosted on the same vCenter server that hosts the existing resource pool. See "Retrieve a List of Resource Pools from a vCenter Server," on page 168 for an example that lists the resource pools available on that server.

Request:

```
POST https://vcloud.example.com/api/admin/extension/providervdc/35/action/updateResourcePools
Content-Type: application/vnd.vmware.admin.resourcePoolSetUpdateParams+xml
...
<?xml version="1.0" encoding="UTF-8"?>
<vmext:UpdateResourcePoolSetParams
   xmlns:vmext="http://www.vmware.com/vcloud/extension/v1.5"
   xmlns:vcloud="http://www.vmware.com/vcloud/v1.5">
   <vmext:AddItem>
      <vmext:VimServerRef
         type="application/vnd.vmware.admin.vmwvirtualcenter+xml"
         href="https://vcloud.example.com/api/admin/extension/vimServer/9" />
      <vmext:MoRef>resgroup-230</vmext:MoRef>
      <vmext:VimObjectType>RESOURCE_POOL</vmext:VimObjectType>
   </vmext:AddItem>
</vmext:UpdateResourcePoolSetParams>
```

Response:

```
202 Accepted
Content-Type: application/vnd.vmware.vcloud.task+xml
...
<Task ... >
   ...
</Task>
```

Remove Resource Pools From a Provider vDC

A system administrator can remove secondary resource pools from a provider vDC. The primary resource pool cannot be changed or removed.

Prerequisites

- Verify that you are logged in as a system administrator.
- Retrieve the resource pool list from the provider vDC. See "Retrieve a Provider vDC Resource Pool Set," on page 176. The `VMWProviderVdcResourcePoolSet` lists the provider vDC's existing resource pools.

Procedure

1. Examine the resource pool list and find the pool to remove.
2. Verify that no virtual machines are using the resource pool.
3. Create an `UpdateResourcePoolSetParams` element that contains a `DeleteItem` element for each resource pool to remove.
4. POST the `UpdateResourcePoolSetParams` element you created in Step 3 to the provider vDC's `resourcePools` link.

Example: Remove a Resource Pool From a Provider vDC

This request removes one of the two resource pools shown in "Retrieve a Provider vDC Resource Pool Set," on page 176. The resource pool removed is the one that contains `<vmext:Primary>false</vmext:Primary>`. The response is a task.

Request:

```
POST https://vcloud.example.com/api/admin/extension/providervdc/35/action/updateResourcePools
Content-Type: application/vnd.vmware.admin.resourcePoolSetUpdateParams+xml
...
<?xml version="1.0" encoding="UTF-8"?>
<vmext:UpdateResourcePoolSetParams
   xmlns:vmext="http://www.vmware.com/vcloud/extension/v1.5"
   xmlns:vcloud="http://www.vmware.com/vcloud/v1.5">
   <vmext:DeleteItem
      href="https://vcloud.example.com/api/admin/extension/resourcePool/66" />
</vmext:UpdateResourcePoolSetParams>
```

Response:

```
202 Accepted
Content-Type: application/vnd.vmware.vcloud.task+xml
...
<Task ... >
...
</Task>
```

Create an External Network

An external network is a reference to a portgroup on a vCenter server attached to vCloud Director. To create an external network, a system administrator must specify the vCenter server and a portgroup associated with it. External networks provide support for bridged organization networks.

Only a system administrator can create an external network. A system administrator can modify an external network to change properties such as its description, but cannot change the portgroup that backs the network. An organization administrator can retrieve the a read-only representation of an external network to examine its properties.

Prerequisites

- Verify that you are logged in as a system administrator.
- Retrieve the list of available portgroups. See "Retrieve a List of Network Resources from a vCenter Server," on page 170.

Procedure

1 Retrieve the XML representation of the vSphere platform extensions.

Use a request like this one.

```
GET https://vcloud.example.com/api/admin/extension
```

2 Examine the response to locate the `Link` element that contains the URL for adding external networks to the cloud.

   This element has a `rel` attribute value of `add` and a `type` attribute value of `application/vnd.vmware.admin.vmwexternalnet+xml`, as shown here:

```
<Link
   type="application/vnd.vmware.admin.vmwexternalnet+xml"
   rel="add"
   href="https://vcloud.example.com/api/admin/extension/externalnets"/>
```

3 Choose a vCenter server to provide a portgroup for the network.

4 Create a `VMWExternalNetwork` element that specifies the properties of the external network.

   These properties include the portgroup you specified in Step 3.

5 POST the `VMWExternalNetwork` element you created in Step 4 to the URL described in Step 2.

   See the request portion of "Example: Create an External Network," on page 181.

The server creates the external network and returns a `VMWExternalNetwork` element that includes the contents you POSTed, along with a set of `Link` elements that you can use to access, remove, disable, or modify it. A reference to the new external network is added to the `VMWExternalNetworkReferences` element of the `VCloud`. The portgroup you specified is removed from the `VimObjectRefList` of the vCenter server.

Example: Create an External Network

This request creates an external network backed by a portgroup listed in the response portion of "Retrieve a List of Network Resources from a vCenter Server," on page 170.

Request:

```
POST https://vcloud.example.com/api/admin/extension/externalnets
Content-Type: application/vnd.vmware.admin.vmwexternalnet+xml
...
<?xml version="1.0" encoding="UTF-8"?>
<vmext:VMWExternalNetwork
   xmlns:vmext="http://www.vmware.com/vcloud/extension/v1.5"
   xmlns:vcloud="http://www.vmware.com/vcloud/v1.5"
   name="example-extnet"
   type="application/vnd.vmware.admin.vmwexternalnet+xml"
   xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
   <vcloud:Description>ExternalNet</vcloud:Description>
   <vcloud:Configuration>
      <vcloud:IpScope>
         <vcloud:IsInherited>false</vcloud:IsInherited>
         <vcloud:Gateway>10.24.64.126</vcloud:Gateway>
         <vcloud:Netmask>255.255.255.192</vcloud:Netmask>
         <vcloud:Dns1>10.115.120.71</vcloud:Dns1>
         <vcloud:Dns2>10.6.64.29</vcloud:Dns2>
         <vcloud:DnsSuffix>eng.example.com</vcloud:DnsSuffix>
      </vcloud:IpScope>
      <vcloud:FenceMode>isolated</vcloud:FenceMode>
   </vcloud:Configuration>
   <vmext:VimPortGroupRef>
      <vmext:VimServerRef
          href="https://vcloud.example.com/api/admin/extension/vimServer/9" />
      <vmext:MoRef>network-175</vmext:MoRef>
      <vmext:VimObjectType>NETWORK</vmext:VimObjectType>
   </vmext:VimPortGroupRef>
</vmext:VMWExternalNetwork>
```

The response includes a `Task` that tracks the creation of the network, and a set of `Link` elements that you can use to operate on or modify it.

Response:

```
201 Created
Content-Type: application/vnd.vmware.admin/vmwexternalnet+xml
...
<vmext:VMWExternalNetwork
   xmlns:vmext="http://www.vmware.com/vcloud/extension/v1.5"
   xmlns:vcloud="http://www.vmware.com/vcloud/v1.5"
   name="example-extnet"
   id="urn:vcloud:network:85"
   type="application/vnd.vmware.admin.vmwexternalnet+xml"
   href="https://vcloud.example.com/api/admin/extension/externalnet/85"... >
   <vcloud:Link
      rel="alternate"
      type="application/vnd.vmware.admin.network+xml"
      href="https://vcloud.example.com/api/admin/network/85" />
   <vcloud:Link
      rel="edit"
      type="application/vnd.vmware.admin.vmwexternalnet+xml"
      href="https://vcloud.example.com/api/admin/extension/externalnet/85" />
   <vcloud:Link
      rel="remove"
      href="https://vcloud.example.com/api/admin/extension/externalnet/85" />
   <vcloud:Link
      rel="up"
      type="application/vnd.vmware.admin.vmwExternalNetworkReferences+xml"
      href="https://vcloud.example.com/api/admin/extension/externalNetworkReferences" />
   <vcloud:Link
      rel="repair"
      type="application/vnd.vmware.admin.network+xml"
      href="https://vcloud.example.com/api/admin/network/85/action/reset" />
   <vcloud:Description>ExternalNet</vcloud:Description>
   <vcloud:Tasks>
      <vcloud:Task
         status="running"
         startTime="2011-03-10T06:08:31.506-08:00"
         operationName="_network_create_provider_network"
         operation="Busy Network example-extnet(85)" ... >
         ...
       </vcloud:Task>
   </vcloud:Tasks>
   <vcloud:Configuration>
      <vcloud:IpScope>
         <vcloud:IsInherited>false</vcloud:IsInherited>
         <vcloud:Gateway>10.24.64.126</vcloud:Gateway>
         <vcloud:Netmask>255.255.255.192</vcloud:Netmask>
         <vcloud:Dns1>10.115.120.71</vcloud:Dns1>
         <vcloud:Dns2>10.6.64.29</vcloud:Dns2>
```

```
            <vcloud:DnsSuffix>eng.vmware.com</vcloud:DnsSuffix>
         </vcloud:IpScope>
         <vcloud:FenceMode>isolated</vcloud:FenceMode>
      </vcloud:Configuration>
      <vmext:VimPortGroupRef>
         <vmext:VimServerRef
            type="application/vnd.vmware.admin.vmwvirtualcenter+xml"
            href="https://vcloud.example.com/api/admin/extension/vimServer/9" />
         <vmext:MoRef>network-175</vmext:MoRef>
         <vmext:VimObjectType>NETWORK</vmext:VimObjectType>
      </vmext:VimPortGroupRef>
   </vmext:VMWExternalNetwork>
```

Create a Network Pool

Network pools provide support for isolated and NAT-routed networks in a cloud. To create a network pool, a system administrator must specify backing network resources on a vCenter server attached to the cloud.

A network pool is a collection of vSphere network resources that are available to be consumed as needed to create isolated or NAT-routed vApp networks and organization networks. Traffic on each network in a pool is isolated at layer 2 from all other networks.

A network pool can be backed by a vCenter port group or dvSwitch. Pools backed by a dvSwitch can define one or more VLANs or fences. All network pools are defined by a VMWNetworkPool element. This element can have one of three types, specified by its xsi:type attribute. The contents of the element depend on its type. See "Create a VLAN-Backed Network Pool," on page 184, "Create an Isolation-Backed Network Pool," on page 185, and "Create a Portgroup-Backed Network Pool," on page 187.

Only a system administrator can create a network pool. A system administrator can modify a network pool to change properties such as its description, but cannot change the network resources, such as virtual switches or portgroups, that provide backing for it.

Prerequisites

Verify that you are logged in as a system administrator.

Procedure

1 Retrieve the XML representation of the vSphere platform extensions.

Use a request like this one.

```
GET https://vcloud.example.com/api/admin/extension
```

2 Examine the response to locate the Link element that contains the URL for adding network pools to your cloud.

This element has a rel attribute value of add and a type attribute value of application/vnd.vmware.admin.networkPool+xml, as shown here:

```
<Link
   type="application/vnd.vmware.admin.networkPool+xml"
   rel="add"
   href="https://vcloud.example.com/api/admin/extension/networkPools"/>
```

3 Create a VMWNetworkPool element that specifies the pool type and backing vCenter resources.

Details of this element's contents depend on the type of pool you are creating.

4 POST the VMWNetworkPool element you created in Step 3 to the URL described in Step 2.

The server creates the network pool and returns a VMWNetworkPool element that includes the contents you POSTed, along with a set of Link elements that you can use to access, remove, disable, or modify it. A reference to the new network pool is added to the VMWNetworkPoolReferences element of the VCloud. Network resources you specified in the VMWNetworkPool element are removed from the VimObjectRefList of the vCenter server.

Create a VLAN-Backed Network Pool

To create a VLAN-backed network pool, create a VMWNetworkPool element whose type attribute has the value VlanPoolType, and POST the element to your cloud's add link for networkPools.

A VLAN-backed network pool is backed by a range of VLAN IDs.

Prerequisites

- Verify that you are logged in as a system administrator.
- Verify that you know your cloud's add URL for networkPools. See "Create a Network Pool," on page 183.
- Verify that at least one vCenter server attached to your cloud has network resources available. See "Retrieve a List of Network Resources from a vCenter Server," on page 170

Procedure

1. Choose a vCenter server to provide a switch for the network pool.
2. Create a VMWNetworkPool element that specifies the properties of the network pool.

   See the request potion of "Example: Create a VLAN-Backed Network Pool," on page 184.
3. POST the VMWNetworkPool element you created in Step 2 to your cloud's add URL for networkPools.

   See the request portion of "Example: Create a VLAN-Backed Network Pool," on page 184.

Example: Create a VLAN-Backed Network Pool

Request:

```
POST https://vcloud.example.com/api/admin/extension/networkPools
Content-Type: application/vnd.vmware.admin.networkPool+xml
...
<?xml version="1.0" encoding="UTF-8"?>
<vmext:VMWNetworkPool
   xmlns:vmext="http://www.vmware.com/vcloud/extension/v1.5"
   xmlns:vcloud="http://www.vmware.com/vcloud/v1.5"
   xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
   xsi:type="vmext:VlanPoolType"
   name="example-Vlan-pool">
   <vcloud:Description>Example VLAN-backed network pool</vcloud:Description>
   <vmext:VlanRange>
      <vmext:Start>1</vmext:Start>
      <vmext:End>4</vmext:End>
   </vmext:VlanRange>
      <vmext:VimSwitchRef>
         <vmext:VimServerRef
            href="https://vcloud.example.com/api/admin/extension/vimServer/9" />
         <vmext:MoRef>dvs-33</vmext:MoRef>
         <vmext:VimObjectType>DV_SWITCH</vmext:VimObjectType>
      </vmext:VimSwitchRef>
 </vmext:VMWNetworkPool>
```

The response includes a `Task` that tracks the creation of the network pool, and a set of `Link` elements that you can use to operate on or modify it.

Response:

```
201 Created
Content-Type: application/vnd.vmware.admin.networkPool+xml
...
<vmext:VMWNetworkPool
   xmlns:vmext="http://www.vmware.com/vcloud/extension/v1.5"
   xmlns:vcloud="http://www.vmware.com/vcloud/v1.5"
   xsi:type="vmext:VlanPoolType"
   name="example-Vlan-pool"
   id="urn:vcloud:networkpool:67"
   type="application/vnd.vmware.admin.networkPool+xml"
   href="https://vcloud.example.com/api/admin/extension/networkPool/67" ... >
   <vcloud:Link
      rel="up"
      type="application/vnd.vmware.admin.vmwNetworkPoolReferences+xml"
      href="https://vcloud.example.com/api/admin/extension/networkPoolReferences" />
   <vcloud:Link
      rel="edit"
      type="application/vnd.vmware.admin.networkPool+xml"
      href="https://vcloud.example.com/api/admin/extension/networkPool/67" />
   <vcloud:Link
      rel="remove"
      href="https://vcloud.example.com/api/admin/extension/networkPool/67" />
   <vcloud:Description>Example VLAN-backed network pool</vcloud:Description>
   <vcloud:Tasks>
      <vcloud:Task
         status="running"
         ...
         operation="Creating Network Pool 67"
         ...
      </vcloud:Task>
   </vcloud:Tasks>
   ...
</vmext:VMWNetworkPool>
```

Create an Isolation-Backed Network Pool

To create an isolation-backed network pool, create a `VMWNetworkPool` element whose `type` attribute has the value `FencePoolType`, and POST the element to your cloud's `add` link for `networkPools`.

An isolation-backed network pool is backed by one or more vCenter isolated networks, and provides traffic isolation from other hosts. The system provisions isolated networks automatically.

Prerequisites

- Verify that you are logged in as a system administrator.
- Verify that you know your cloud's `add` URL for `networkPools`. See "Create a Network Pool," on page 183.
- Verify that at least one vCenter server attached to your cloud has network resources available. See "Retrieve a List of Network Resources from a vCenter Server," on page 170

Procedure

1. Choose a vCenter server to provide a switch for the network pool.
2. Create a `VMWNetworkPool` element that specifies the properties of the network pool.
3. POST the `VMWNetworkPool` element you created in Step 2 to your cloud's `add` URL for `networkPools`.

   See the request portion of "Example: Create an Isolation-Backed Network Pool," on page 186.

Example: Create an Isolation-Backed Network Pool

Request:

```
POST https://vcloud.example.com/api/admin/extension/networkPools
Content-Type: application/vnd.vmware.admin.networkPool+xml
...
<?xml version="1.0" encoding="UTF-8"?>
<vmext:VMWNetworkPool
   xmlns:vmext="http://www.vmware.com/vcloud/extension/v1.5"
   xmlns:vcloud="http://www.vmware.com/vcloud/v1.5"
   xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
   xsi:type="vmext:FencePoolType"
   name="example-isolation-pool"
   type="application/vnd.vmware.admin.networkPool+xml">
   <vcloud:Description>Example Isolation-backed network pool</vcloud:Description>
   <vmext:FenceIdCount>5</vmext:FenceIdCount>
   <vmext:VlanId>0</vmext:VlanId>
   <vmext:VimSwitchRef>
      <vmext:VimServerRef
         type="application/vnd.vmware.admin.vmwvirtualcenter+xml"
         href="https://vcloud.example.com/api/admin/extension/vimServer/9" />
      <vmext:MoRef>dvs-39</vmext:MoRef>
      <vmext:VimObjectType>DV_SWITCH</vmext:VimObjectType>
   </vmext:VimSwitchRef>
</vmext:VMWNetworkPool>
```

The response includes a `Task` that tracks the creation of the network pool, and a set of `Link` elements that you can use to operate on or modify it.

Response:

```
201 Created
Content-Type: application/vnd.vmware.admin.networkPool+xml
...
<vmext:VMWNetworkPool
   xmlns:vmext="http://www.vmware.com/vcloud/extension/v1.5"
   xmlns:vcloud="http://www.vmware.com/vcloud/v1.5"
   xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
   xsi:type="vmext:FencePoolType"
   name="example-isolation-pool"
   id="urn:vcloud:networkpool:70"
   type="application/vnd.vmware.admin.networkPool+xml"
   href="https://vcloud.example.com/api/admin/extension/networkPool/70" ... >
   <vcloud:Link
      rel="up"
      type="application/vnd.vmware.admin.vmwNetworkPoolReferences+xml"
      href="https://vcloud.example.com/api/admin/extension/networkPoolReferences" />
   <vcloud:Link
```

```
      rel="edit"
      type="application/vnd.vmware.admin.networkPool+xml"
      href="https://vcloud.example.com/api/admin/extension/networkPool/70" />
   <vcloud:Link
      rel="remove"
      href="https://vcloud.example.com/api/admin/extension/networkPool/70" />
   <vcloud:Description>Example isolation-backed network pool</vcloud:Description>
   <vcloud:Tasks>
      <vcloud:Task
         status="running"
         ...
         operation="Creating Network Pool 70 ..."
         ...
      </vcloud:Task>
   </vcloud:Tasks>
   ...
</vmext:VMWNetworkPool>
```

Create a Portgroup-Backed Network Pool

To create a portgroup-backed network pool, you create a VMWNetworkPool element whose type attribute has the value PortGroupPoolType, and POST the element to your cloud's add link for networkPools.

Prerequisites

- Verify that you are logged in as a system administrator.
- Verify that you know your cloud's add URL for networkPools. See "Create a Network Pool," on page 183.
- Verify that at least one vCenter server attached to your cloud has network resources available. See "Retrieve a List of Network Resources from a vCenter Server," on page 170

Procedure

1 Choose a vCenter server to provide a portgroup for the network pool.

2 Create a VMWNetworkPool element that specifies the properties of the network pool.

  See the request portion of "Example: Create a Portgroup-Backed Network Pool," on page 187.

3 POST the VMWNetworkPool element you created in Step 2 to your cloud's add URL for networkPools. See "Create a Network Pool," on page 183.

  See the request portion of "Example: Create a Portgroup-Backed Network Pool," on page 187.

Example: Create a Portgroup-Backed Network Pool

Request:

```
POST https://vcloud.example.com/api/admin/extension/networkPools
Content-Type: application/vnd.vmware.admin.networkPool+xml
...
<?xml version="1.0" encoding="UTF-8"?>
<vmext:VMWNetworkPool
   xmlns:vmext="http://www.vmware.com/vcloud/extension/v1.5"
   xmlns:vcloud="http://www.vmware.com/vcloud/v1.5"
   xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
   xsi:type="vmext:PortGroupPoolType"
   name="example-portgroup-pool"
   type="application/vnd.vmware.admin.networkPool+xml">
   <vcloud:Description>Example portgroup-backed network pool</vcloud:Description>
   <vmext:PortGroupRefs>
      <vmext:VimObjectRef>
         <vmext:VimServerRef
            href="https://vcloud.example.com/api/admin/extension/vimServer/9" />
         <vmext:MoRef>dvportgroup-32</vmext:MoRef>
         <vmext:VimObjectType>DV_PORTGROUP</vmext:VimObjectType>
      </vmext:VimObjectRef>
   </vmext:PortGroupRefs>
   <vmext:VimServer
      href="https://vcloud.example.com/api/admin/extension/vimServer/9"></vmext:VimServer>
</vmext:VMWNetworkPool>
```

The response includes a `Task` that tracks the creation of the network pool, and a set of `Link` elements that you can use to operate on or modify it.

Response:

```
201 Created
Content-Type: application/vnd.vmware.admin.networkPool+xml
...
<vmext:VMWNetworkPool
   xmlns:vmext="http://www.vmware.com/vcloud/extension/v1.5"
   xmlns:vcloud="http://www.vmware.com/vcloud/v1.5"
   xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
   xsi:type="vmext:PortGroupPoolType"
   name="example-portgroup-pool"
   id="urn:vcloud:networkpool:66"
   type="application/vnd.vmware.admin.networkPool+xml"
   href="https://vcloud.example.com/api/admin/extension/networkPool/66" ... >
   <vcloud:Link
      rel="up"
      type="application/vnd.vmware.admin.vmwNetworkPoolReferences+xml"
      href="https://vcloud.example.com/api/admin/extension/networkPoolReferences" />
   <vcloud:Link
      rel="edit"
      type="application/vnd.vmware.admin.networkPool+xml"
      href="https://vcloud.example.com/api/admin/extension/networkPool/66" />
   <vcloud:Link
      rel="remove"
      href="https://vcloud.example.com/api/admin/extension/networkPool/66" />
   <vcloud:Description>Example portgroup-backed network pool</vcloud:Description>
   <vcloud:Tasks>
      <vcloud:Task
         status="running"
         ...
         operation="Creating Network Pool 66"
         ...
      </vcloud:Task>
   </vcloud:Tasks>
   ...
</vmext:VMWNetworkPool>
```

Import a Virtual Machine from vCenter

A system administrator can import virtual machines from the inventory of any vCenter server registered to vCloud Director. You can import the virtual machines to any vDC in a cloud, and you can import them in vApp or vApp template form.

When you import a virtual machine from vCenter, you must specify the following items:

- A target vDC to receive the import
- A form for the imported virtual machine to take. Choose vApp or vApp template.
- Whether to remove the source virtual machine from vCenter inventory after the import is complete

Prerequisites

- Verify that you are logged in as a system administrator.
- Identify the virtual machine to import. See "Retrieve a List of Virtual Machines from a vCenter Server," on page 171.

Procedure

1 Choose whether to import the virtual machine as a vApp or vApp template.

  The `VimServer` element that represents the vCenter server from which you import the virtual machine contains two links that import virtual machines. One has the following form, and imports the virtual machine as a vApp.

```
<vcloud:Link
      rel="add"
      type="application/vnd.vmware.admin.importVmAsVAppParams+xml"
      href="https://vcloud.example.com/api/admin/extension/vimServer/9/importVmAsVApp" />
```

  The other has the following form, and imports the virtual machine as a vApp template.

```
<vcloud:Link
      rel="add"
      type="application/vnd.vmware.admin.importVmAsVAppTemplateParams+xml"

href="https://vcloud.example.com/api/admin/extension/vimServer/9/importVmAsVAppTemplate" />
```

2 (Optional) If you plan to import the virtual machine as a vApp template, identify a catalog where you want to place a reference to the template.

Import a Virtual Machine as a vApp

To import a virtual machine as a vApp, a system administrator can make a request to the `importVmAsVApp` link of the `VimServer` that manages the virtual machine.

Prerequisites

- Verify that you are logged in as a system administrator.
- Identify the virtual machine to import. See "Retrieve a List of Virtual Machines from a vCenter Server," on page 171.

Procedure

1 Create an `ImportVmAsVAppParams` element that specifies the `VmMoRef` of the source virtual machine and a target vDC to hold the imported vApp.

2 POST the `ImportVmAsVAppParams` element to the `importVmAsVApp` link of the source vCenter server.

Example: Import a Virtual Machine as a vApp

This example imports one of the virtual machines shown in the response portion of "Example: Retrieve a List of Virtual Machines from a vCenter Server," on page 172. The request body is an ImportVmAsVAppParams element whose sourceMove attribute specifies that the source virtual machine should remain in vCenter inventory after the import is complete. The request body includes the href of the vDC that receives the import and a VmMoRef element that contains the managed object reference of the virtual machine to import. The response is an unresolved vApp body that contains a Task that tracks the import.

Request:

```
POST https://vcloud.example.com/api/admin/extension/vimServer/9/importVmAsVapp
Content-type: application/vnd.vmware.admin.importVmAsVAppParams+xml
...
<?xml version="1.0" encoding="UTF-8"?>
<ImportVmAsVAppParams
   xmlns="http://www.vmware.com/vcloud/extension/v1.5"
   name="ImportedWin2K8"
   sourceMove="false">
   <VmMoRef>vm-43</VmMoRef>
   <Vdc
      href="http://vcloud.example.com/api/vdc/2" />
</ImportVmAsVAppParams>
```

Response:

```
201 Created
Content-Type: application/vnd.vmware.vcloud.vApp+xml
...
<VApp ...
   status="0"
   name="ImportedWin2K8"
   type="application/vnd.vmware.vcloud.vApp+xml"
   href="https://vcloud.example.com/api/vApp/vapp-102" ... >
   ...
   <Description />
   <Tasks>
      <Task
         operation=”Busy Virtual Application Win2K8 ”>
         ...
      </Task>
   </Tasks>
</VApp>
```

Import a Virtual Machine as a vApp Template

To import a virtual machine as a vApp template, a system administrator can make a request to the importVmAsVAppTemplate link of the VimServer that manages the virtual machine.

Prerequisites

- Verify that you are logged in as a system administrator.
- Identify the virtual machine to import. See "Retrieve a List of Virtual Machines from a vCenter Server," on page 171.

Procedure

1 Create an `ImportVmAsVAppTemplateParams` element that specifies the `VmMoRef` of the source virtual machine, a target vDC to hold the imported vApp template, and an optional catalog where you want to place a reference to the template.

2 POST the `ImportVmAsVAppTemplateParams` element to the `importVmAsVAppTemplate` link of the source vCenter server.

Example: Import a Virtual Machine as a vApp Template

This example imports one of the virtual machines shown in the response portion of "Example: Retrieve a List of Virtual Machines from a vCenter Server," on page 172 as a vApp template. The request body is an `ImportVmAsVAppTemplateParams` element whose `sourceMove` attribute specifies that the source virtual machine should remain in vCenter inventory after the import is complete. The request body includes the `href` of the vDC that receives the import, a `VmMoRef` element that contains the managed object reference of the virtual machine to import, and a `Catalog` element that references the catalog to which the imported template should be added. The response is an unresolved `VAppTemplate` body that contains a `Task` that tracks the import.

Request:

```
POST https://vcloud.example.com/api/admin/extension/vimServer/9/importVmAsVappTemplate
Content-type: application/vnd.vmware.admin.importVmAsVAppTemplateParams+xml
...
<?xml version="1.0" encoding="UTF-8"?>
<ImportVmAsVAppTemplateParams
   xmlns="http://www.vmware.com/vcloud/extension/v1.5"
   name="ImportedWin2K8"
   sourceMove="false">
   <VmMoRef>vm-43</VmMoRef>
   <Vdc
      href="http://vcloud.example.com/api/vdc/2" />
   <Catalog
      href="http://vcloud.example.com/api/catalog/32" />
</ImportVmAsVAppTemplateParams>
```

Response:

```
201 Created
Content-Type: application/vnd.vmware.vcloud.vAppTemplate+xml
...
<VAppTemplate ...
   status="0"
   name="ImportedWin2K8"
   type="application/vnd.vmware.vcloud.vAppTemplate+xml"
   href="https://vcloud.example.com/api/vAppTemplate/vappTemplate-64" ... >
   ...
   <Description />
   <Tasks>
      <Task
         operation=”Busy Virtual Application Template Win2K8 ”>
         ...
      </Task>
   </Tasks>
</VAppTemplate>
```

Relocate a Virtual Machine to a Different Datastore

A Vm element, whether part of a Vapp or VappTemplate element, includes an action/relocate link that you can use to move the virtual machine to a different datastore

In both vApps and vApp templates, virtual machines are defined by a combination of metadata in the vCloud Director database and virtual disk files stored on a datastore in the cloud. vCloud Director determines the initial location of this datastore based on how the virtual machine was created. You can move the disk files to another datastore in the cloud by POSTing a RelocateParams element to the virtual machine's action/relocate link.

When a virtual machine is deployed in a vDC that supports fast provisioning, its disks can become part of a tree structure that includes parent and child disks. Disk tree structure and total storage consumption remain the same regardless of the order in which virtual machines are relocated The folder structure in the target datastore might change depending on the order in which virtual machines are relocated.

Prerequisites

Verify that you are logged in as a system administrator.

Procedure

1 List the datastores available in the cloud.

Use a request like this one.

```
GET https://vcloud.example.com/api/admin/extension/datastores
```

The response is a DatastoreReferences element, as shown here:.

```
<vmext:DatastoreReferences
   xmlns:vmext="http://www.vmware.com/vcloud/extension/v1.5"
   xmlns:vcloud="http://www.vmware.com/vcloud/v1.5"
   type="application/vnd.vmware.admin.datastoreList+xml"
   xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
   xsi:schemaLocation="http://vcloud.example.com/api/v1.5/schema/master.xsd">
   <vcloud:Link
      rel="up"
      type="application/vnd.vmware.admin.vmwExtension+xml"
      href="https://vcloud.example.com/api/admin/extension" />
   <vcloud:Reference
      type="application/vnd.vmware.admin.datastore+xml"
      name="example-lun2"
      href="https://vcloud.example.com/api/admin/extension/datastore/607" />
   <vcloud:Reference
      type="application/vnd.vmware.admin.datastore+xml"
      name="example-lun4"
      href="https://vcloud.example.com/api/admin/extension/datastore/191" />
 </vmext:DatastoreReferences>
```

2 Retrieve the representation of the virtual machine.

The following request retrieves a Vm from a Vapp:

```
GET https://vcloud.example.com/api/vApp/vm-257
```

3 Find the `action/relocate` link for the `Vm`, and note the datastore that the `Vm` currently occupies.

  This fragment of the response to the request made in Step 2 includes the `action/relocate` link and the reference to the datastore that the `Vm` currently occupies.

```
<Vm
   xmlns="http://www.vmware.com/vcloud/v1.5"
   ... >
   <VCloudExtension
      required="false">
   ...
       <vmext:VimObjectRef>
         <vmext:VimServerRef
            type="application/vnd.vmware.admin.vmwvirtualcenter+xml"
            name="VC1"
            href="https://vcloud.example.com/api/admin/extension/vimServer/35" />
         <vmext:MoRef>datastore-191</vmext:MoRef>
         <vmext:VimObjectType>DATASTORE</vmext:VimObjectType>
      </vmext:VimObjectRef>
   </VCloudExtension>
   ...
   <Link
      rel="relocate"
      type="application/vnd.vmware.vcloud.relocateVmParams+xml"
      href="https://vcloud.example.com/api/vApp/vm-257/action/relocate" />
   ...
</Vm>
```

4 Construct a `RelocateParams` element that specifies the destination datastore for the `Vm`.

  See "Example: Relocate a Virtual Machine," on page 193.

5 POST the `RelocateParams` element to the `action/relocate` link described in Step 3.

  You do not need to power off the virtual machine before you make the request. See "Example: Relocate a Virtual Machine," on page 193.

The system creates a task to manage the relocation and returns a `Task` element in the response. When the task is complete, the virtual machine is relocated.

Example: Relocate a Virtual Machine

This request relocates the virtual machine retrieved in Step 2 to one of the datastores listed in Step 1. The response is a `Task`.

Request:

```
POST https://vcloud.example.com/api/vApp/vm-257/action/relocate
Content-Type: application/vnd.vmware.vcloud.relocateVmParams+xml
...
<?xml version="1.0" encoding="UTF-8"?>
<RelocateParams
   xmlns="http://www.vmware.com/vcloud/v1.5">
   <Datastore
      href="https://vcloud.example.com/api/admin/extension/datastore/607" />
</RelocateParams>
```

Working With Object Metadata 8

The vCloud API provides a general-purpose facility for associating user-defined metadata with an object. A system administrator or the owner of an object can use the `metadata` link in the object's representation to access object metadata.

The representations of most first-class objects in the vCloud API include a link to a `Metadata` element, which contains zero or more *`name=value`* pairs that the owner of an object can create, retrieve, update, and delete. Object metadata gives cloud operators and cloud tenants a flexible way to apply properties to objects and use property values to help integrate the use and management of those objects with a variety of applications. Object metadata is preserved when objects are copied.

vCloud API Object Metadata Links

The representation of any object that has metadata includes a link that you can use to retrieve the object's `Metadata` element. This example shows the `metadata` link from an `Org` element.

```
<Link
   rel="down"
   type="application/vnd.vmware.vcloud.metadata+xml"
   href="https://vcloud.example.com/api/org/26/metadata"/>
```

vCloud API Object Metadata Contents

An object's `Metadata` element can contain up to 1024 `MetadataEntry` elements. Each `MetadataEntry` includes a single *`name=value`* pair, represented in its `Key` and `Value` elements. A system administrator or object owner can retrieve or update a `Value` by making a request to a URL that includes the name of the `Key`.

Key names are defined by `Key` element contents. These names are Unicode strings and must be encoded as described in RFC3986 (`pct-encoded`). A key name must be unique within the scope of an object's metadata. To modify an individual key name, you must modify the entire `Metadata` element.

The following rules apply when you update a `Metadata` element.

- When the content of a `Key` element in the update does not match the content of an existing `Key`, the `MetadataEntry` containing that `Key` is added to the `Metadata` element.
- When the content of `Key` element in the update matches the content of an existing `Key`, the `MetadataEntry` containing that `Key` is replaced.

vCloud API Object Metadata Limits

The following limits apply to vCloud API object metadata:

Metadata key size The contents of a Key element in a MetadataEntry cannot exceed 256 Unicode characters.

Metadata size The size of all Metadata for an object, computed as the sum of all Key and Value UTF-8 strings, cannot exceed 128 KB.

MetadataEntry limit The total metadata associated with an object cannot exceed 1024 pairs, where a pair consists of a Key and corresponding Value.

This chapter includes the following topics:

- "Retrieve or Update a Metadata Element," on page 196
- "Retrieve or Update a Metadata Value," on page 199

Retrieve or Update a Metadata Element

A system administrator or the owner of an object can create, retrieve, or update the object's Metadata element. This element contains all object metadata, and operations that modify it merge the modifications with existing contents.

When you create an object, its representation is populated with an empty Metadata element. A system administrator or the object owner can add metadata by updating the Metadata element with new MetadataEntry elements. Each of these elements contains a Key and a Value. The contents of the Key element define the key name, which must be unique within the scope of the object's metadata. You can modify the value associated with an existing key. See "Retrieve or Update a Metadata Value," on page 199.

NOTE The Key element cannot contain a semicolon character (;). In addition, several other character sequences are not allowed, or not allowed in certain positions.

Table 8-1. Content Restrictions for Key

Cannot Contain	Cannot Start with	Cannot End with
/../	./	/.
/./	../	/..

Prerequisites

Verify that you are logged in as a system administrator or the object owner.

Procedure

1 Retrieve the representation of the object.

Examine the response to find its metadata link. This example shows the metadata link from an Org.

```
<Link
   rel="down"
   type="application/vnd.vmware.vcloud.metadata+xml"
   href="https://vcloud.example.com/api/org/26/metadata"/>
```

2 Retrieve the Metadata element.

If the object has no metadata, the element contains only a rel="edit" link that you can use when modifying the element and a rel="up" link that references the containing object, as shown in this example.

```
<Metadata
   xmlns="http://www.vmware.com/vcloud/v1.5"
   type="application/vnd.vmware.vcloud.metadata+xml"
   href="https://vcloud.example.com/api/admin/org/26/metadata"
   ... >
   <Link
      rel="add"
      type="application/vnd.vmware.vcloud.metadata+xml"
      href="https://vcloud.example.com/api/admin/org/26/metadata" />
   <Link
      rel="up"
      type="application/vnd.vmware.vcloud.org+xml"
      href="https://vcloud.example.com/api/admin/org/37" />
</Metadata>
```

3 Modify the retrieved Metadata element.

You can add new MetadataEntry elements or modify existing ones. If you modify existing ones, your modifications are merged into the object's Metadata following the rules listed in "vCloud API Object Metadata Contents," on page 195.

4 POST the Metadata element to the rel="add" link described in Step 2.

See "Example: Update a Metadata Element," on page 197.

Example: Update a Metadata Element

This example updates the empty Metadata element shown in Step 2 to create two MetadataEntry elements specific to this organization. The response is a Task.

Request:

```
POST https://vcloud.example.com/api/admin/org/26/metadata
Content-Type: application/vnd.vmware.vcloud.metadata+xml
...
<?xml version="1.0" encoding="UTF-8"?>
<Metadata
   type="application/vnd.vmware.vcloud.metadata+xml"
   xmlns="http://www.vmware.com/vcloud/v1.5">
   <MetadataEntry>
      <Key>Organization Web Page</Key>
      <Value>http://internal.example.com/orgs/Finance</Value>
   </MetadataEntry>
   <MetadataEntry>
      <Key>LOS</Key>
      <Value>bronze</Value>
   </MetadataEntry>
</Metadata>
```

Response:

```
<Task
   xmlns="http://www.vmware.com/vcloud/v1.5"
   status="running"
   ...
   operationName="metadataUpdate"
   operation="Updating metadata for Organization (26)"
   ... >
   ...
</Task>
```

After the task is complete, the `Metadata` element is updated to contain the entries specified in the request, along with links that you can use to retrieve or update individual values.

```
<Metadata
   xmlns="http://www.vmware.com/vcloud/v1.5"
   type="application/vnd.vmware.vcloud.metadata+xml"
   href="https://vcloud.example.com/api/org/26/metadata"
   ... >
   <Link
      rel="add"
      type="application/vnd.vmware.vcloud.metadata+xml"
      href="https://vcloud.example.com/api/org/26/metadata" />
   <Link
      rel="up"
      type="application/vnd.vmware.vcloud.org+xml"
      href="https://vcloud.example.com/api/org/26" />
   <MetadataEntry>
      <Link
         rel="edit"
         type="application/vnd.vmware.vcloud.metadata.value+xml"
         href="https://vcloud.example.com/api/org/26/metadata/Organization+Web+Page" />
      <Link
         rel="remove"
         type="application/vnd.vmware.vcloud.metadata.value+xml"
         href="https://vcloud.example.com/api/org/26/metadata/Organization+Web+Page" />
      <Link
         rel="up"
         type="application/vnd.vmware.vcloud.metadata+xml"
         href="https://vcloud.example.com/api/org/26/metadata" />
      <Key>Organization Web Page</Key>
      <Value>http://internal.example.com/orgs/Finance</Value>
   </MetadataEntry>
   <MetadataEntry>
      <Link
         rel="edit"
         type="application/vnd.vmware.vcloud.metadata.value+xml"
         href="https://vcloud.example.com/api/org/26/metadata/LOS" />
      <Link
         rel="remove"
         type="application/vnd.vmware.vcloud.metadata.value+xml"
         href="https://vcloud.example.com/api/org/26/metadata/LOS" />
      <Link
         rel="up"
         type="application/vnd.vmware.vcloud.metadata+xml"
         href="https://vcloud.example.com/api/org/26/metadata" />
      <Key>LOS</Key>
      <Value>bronze</Value>
   </MetadataEntry>
</Metadata>
```

Retrieve or Update a Metadata Value

Each *name*=*value* pair in an object's metadata is represented as a MetadataEntry element, which includes links that a system administrator or object owner can use to retrieve or update the metadata value, or delete the MetadataEntry.

Prerequisites

- Verify that you are logged in as a system administrator or the object owner.
- Retrieve the object's Metadata element. See "Retrieve or Update a Metadata Element," on page 196

Procedure

1 Examine the retrieved Metadata element to find the Key you want.

  Each key name is defined by the content of a Key element.

2 Retrieve or update the value.

  Make a request to the URL in the value of the href attribute of the MetadataEntry that contains the Key. (To modify the Key contents, you must update the entire Metadata element. See "Retrieve or Update a Metadata Element," on page 196.)

  - To retrieve the value, make a GET request to the URL. The response is a MetadataValue element.
  - To update the value, make a PUT request to the URL and supply a MetadataValue element as the request body. See ."Example: Update a Metadata Value," on page 199

NOTE If you know the name of a key (the content of a Key element), you can construct a URL to retrieve its value by appending the key name to the object's metadata URL. The key name is case-sensitive. Any special characters it contains must be appropriately escaped in the URL. See "vCloud API Object Metadata Contents," on page 195.

Example: Update a Metadata Value

This request updates the value of the metadata Key named LOS from the original value of bronze (shown in "Example: Update a Metadata Element," on page 197) to a new value of silver. The response is a task.

Request:

```
PUT https://vcloud.example.com/api/admin/org/26/metadata/LOS
Content-Type: application/vnd.vmware.vcloud.metadata.value+xml
...
<?xml version="1.0" encoding="UTF-8"?>
<MetadataValue
   xmlns="http://www.vmware.com/vcloud/v1.5">
   <Value>silver</Value>
</MetadataValue>
```

Response:

```
<Task
   xmlns="http://www.vmware.com/vcloud/v1.5"
   status="running"
   ...
   operationName="metadataUpdate"
```

```
      operation="Updating metadata for Organization (26)"
      ... >
      ...
   </Task>
```

Using the Query Service 9

You can use the vCloud API query service to query the vCloud Director database for information about objects in the cloud.

The query service provides the following kinds of queries:

- Typed queries, which require you to construct a query URL that specifies a query type and optional parameters.
- Packaged queries, which have well-known URLs and can accept many of the same parameters used with typed queries.

Both typed and packaged queries allow you to specify one of the following formats in which to display the result set:

- A `records` format that returns *name*=*value* pairs for all properties of each matching object. This is the default.
- An `idrecords` format that is identical to the `records` format, except that object references are returned in `id` format rather than `href` format. See "Objects, References, and Representations," on page 11.
- A `references` format that returns a reference in `href` format to each object that matches the query criteria.

Query results are paginated, and include links to previous and next pages where needed. Page size can be specified in the query request. Default and maximum page sizes are specified in the vCloud Director configuration. You can also apply filter criteria to the list of items returned.

This chapter includes the following topics:

Typed Queries

Typed queries require you to construct a query URL that specifies an object type and optional parameters.

Query Syntax

Typed queries have the following syntax:

```
API-URL/query?type=name[&param][&param ... ][&filter]
```

- *API-URL* is a URL of the form `https://vcloud.example.com/api`.
- *name* is the name of a query type. Type names are case-sensitive.

- *param* is an optional query parameter. Zero or more parameters are allowed. See "Query Parameters," on page 211.
- *filter* is an optional filter expression. At most one filter expression is allowed. See "Filter Expressions," on page 212.

Query Types

Each query type returns its result set as an XML document in which objects are represented as elements and object properties are represented as attributes, pairing the name of the property with its value at the time the request was run. By default, result sets are returned in the `records` format, which shows all database records for each object. Most queries also support the `references` format, which returns a set of object references, including `name`, `type`, and `href` attributes. All queries that support the `records` format also support the `idrecords` format. For more information about the `format` parameter, see "Query Parameters," on page 211.

You can retrieve a summary list of all query types accessible to the currently authenticated user by making a request list this one:

```
GET https://vcloud.example.com/api/query
```

If you make a query whose result set you do not have rights to view, a response code of ACCESS_TO_RESOURCE_IS_FORBIDDEN (403) is returned.

Table 9-1. Query Types

Type Name	Result Set	User	Admin	System	Format
adminAllocatedExternalAddress	All allocated external IP addresses in the system and the networks that use them	No	No	YES	records, idrecords
adminCatalog	All catalogs in the system	No	No	YES	All
adminCatalogItem	Catalog items in all catalogs in the system	No	No	YES	All
adminGroup	Groups in all organizations in the system	No	No	YES	All
adminMedia	All media in the system	No	No	YES	All
adminOrgNetwork	All organization networks in the system	No	No	YES	All
adminOrgVdc	All vDCs in the system	No	No	YES	All
adminShadowVM	All shadow virtual machines in the system	No	No	YES	All
adminTask	All active or recently completed tasks in the system	No	No	YES	All
adminUser	Users in all organizations in the system	No	No	YES	All
adminVApp	All vApps in the system	No	No	YES	All
adminVAppNetwork	All vApp networks in the system	No	No	YES	All
adminVAppTemplate	All vApp templates in the system	No	No	YES	All
adminVM	All virtual machines in vApps or vApp templates in the system	No	No	YES	All
allocatedExternalAddress	All allocated external IP addresses in your organization and the networks that use them	No	YES	No	records, idrecords
blockingTask	All active blocking tasks in the system	No	No	YES	All
catalog	All catalogs in your organization that you have rights to view or modify	YES	YES	No	All
catalogItem	All catalog items in your organization you have rights to view or modify	YES	YES	No	All
cell	All vCloud Director cells supporting this cloud	No	No	YES	records, idrecords

Table 9-1. Query Types (Continued)

Type Name	Result Set	User	Admin	System	Format
datastore	All datastores in the system	No	No	YES	All
datastoreProviderVdcRelation	All datastore-to-provider-vDC relationships in the system	No	No	YES	records, idrecords
dvSwitch	All vCenter distributed virtual switches available for use by the system	No	No	YES	records, idrecords
event	All stored event records	YES	YES	YES	records, idrecords
externalNetwork	External networks in the system	No	No	YES	All
group	All groups in your organization	YES	YES	No	All
host	ESX/ESXi hosts registered to the system	No	No	YES	All
media	All media that you can view or modify	YES	YES	No	All
networkPool	All network pools in the system	No	No	YES	All
organization	All organizations visible to you	No	YES	YES	All
orgNetwork	Networks in your organization	YES	YES	No	All
orgVdc	All vDcs in your organization.	YES	YES	No	All
orgVdcResourcePoolRelation	vDC-to-resource-pool relationships in your organization	No	No	YES	records, idrecords
portgroup	All vCenter portgroups in the system	No	No	YES	records, idrecords
providerVdc	All provider vDCs in the system	No	No	YES	records, idrecords
providerVdcResourcePool Relation	Provider-vDC-to-resource-pool relationships in the system	No	No	YES	records, idrecords
resourcePool	All resource pools in the system	No	No	YES	records, idrecords
right	All rights defined in the system	No	YES	YES	All
role	All roles defined in the system	No	YES	YES	All
strandedUser	All stranded users. When you delete an LDAP group, users who were imported from that group become stranded and cannot log in.	No	YES	No	All
task	Active and recently completed tasks in your organization	YES	YES	No	All
user	Users in your organization	YES	YES	No	All
vApp	All vApps that you can view or modify	YES	YES	No	All
vAppNetwork	All vApp networks in your organization (or in the system, if the query is run by a system administrator)	YES	YES	No	All
vAppOrgNetworkRelation	All vApps that are connected to organization networks in your organization (or in the system, if the query is run by a system administrator)	No	YES	YES	All
vAppTemplate	All vApp templates in your organization	YES	YES	No	All

Table 9-1. Query Types (Continued)

Type Name	Result Set	User	Admin	System	Format
virtualCenter	All vCenter servers registered to the system	No	No	YES	All
vm	All virtual machines in your organization (or in the system, if the query is run by a system administrator)	YES	YES	No	All

Examples

Simple typed query using default (records) format.

```
GET https://vcloud.example.com/api/query?type=user
```

Response:

```
<QueryResultRecords
   xmlns="http://www.vmware.com/vcloud/v1.5"
   total="3"
   pageSize="25"
   page="1"
   name="user"
   type="application/vnd.vmware.vcloud.query.records+xml"
   href="https://vcloud.example.com/api/query?
type=user&amp;page=1&amp;pageSize=25&amp;format=records"
   xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
   xsi:schemaLocation="http://www.vmware.com/vcloud/v1.5
http://vcloud.example.com/api/v1.5/schema/master.xsd">
   <Link
      rel="alternate"
      type="application/vnd.vmware.vcloud.query.references+xml"
      href="https://vcloud.example.com/api/query?
type=user&amp;page=1&amp;pageSize=25&amp;format=references" />
   <UserRecord
      storedVMQuotaRank="-1"
      storedVMQuota="0"
      numStoredVMs="0"
      numDeployedVMs="0"
      name="bob"
      ldapUser="false"
      fullName=" "
      enabled="true"
      deployedVMQuotaRank="-1"
      deployedVMQuota="0"
      href="https://vcloud.example.com/api/admin/user/39" />
   <UserRecord
      storedVMQuotaRank="-1"
      storedVMQuota="0"
      numStoredVMs="0"
      numDeployedVMs="0"
      name="administrator"
      ldapUser="false"
      fullName="The Admin"
      enabled="true"
      deployedVMQuotaRank="-1"
      deployedVMQuota="0"
      href="https://vcloud.example.com/api/admin/user/24" />
   <UserRecord
      storedVMQuotaRank="-1"
      storedVMQuota="0"
      numStoredVMs="0"
      numDeployedVMs="0"
      name="zorro"
      ldapUser="true"
      fullName="Viewer"
      enabled="true"
      deployedVMQuotaRank="-1"
      deployedVMQuota="0"
      href="https://vcloud.example.com/api/admin/user/58" />
</QueryResultRecords>
```

Simple typed query using references format.

```
GET https://vcloud.example.com/api/query?type=user&format=references
```

Response:

```
<UserReferences
   xmlns="http://www.vmware.com/vcloud/v1.5"
   total="3"
   pageSize="25"
   page="1"
   name="user"
   type="application/vnd.vmware.vcloud.query.references+xml"
   href="https://vcloud.example.com/api/query?
type=user&amp;page=1&amp;pageSize=25&amp;format=references"
   xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" ...
   <Link
      rel="alternate"
      type="application/vnd.vmware.vcloud.query.records+xml"
      href="https://vcloud.example.com/api/query?
type=user&amp;page=1&amp;pageSize=25&amp;format=records" />
   <UserReference
      type="application/vnd.vmware.admin.user+xml"
      name="bob"
      href="https://vcloud.example.com/api/admin/user/39" />
   <UserReference
      type="application/vnd.vmware.admin.user+xml"
      name="administrator"
      href="https://vcloud.example.com/api/admin/user/24" />
   <UserReference
      type="application/vnd.vmware.admin.user+xml"
      name="zorro"
      href="https://vcloud.example.com/api/admin/user/58" />
</UserReferences>
```

Typed query, reference format, result set sorted by object name.

```
GET https://vcloud.example.com/api/query?type=user&format=references&sortAsc=name
```

Response:

```
<UserReferences
   xmlns="http://www.vmware.com/vcloud/v1.5"
   total="3"
   pageSize="25"
   page="1"
   name="user"
   type="application/vnd.vmware.vcloud.query.references+xml"
   href="https://vcloud.example.com/api/query?
type=user&amp;page=1&amp;pageSize=25&amp;format=references"
   xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" ... >
   <Link
      rel="alternate"
      type="application/vnd.vmware.vcloud.query.records+xml"
      href="https://vcloud.example.com/api/query?
type=user&amp;page=1&amp;pageSize=25&amp;format=records" />
   <UserReference
      type="application/vnd.vmware.admin.user+xml"
      name="administrator"
      href="https://vcloud.example.com/api/admin/user/24" />
   <UserReference
      type="application/vnd.vmware.admin.user+xml"
      name="bob"
      href="https://vcloud.example.com/api/admin/user/39" />
   <UserReference
      type="application/vnd.vmware.admin.user+xml"
      name="zorro"
      href="https://vcloud.example.com/api/admin/user/58" />
</UserReferences>
```

Typed query, reference format, result set filtered by attribute name and value. The result set lists the subset of users who are imported from LDAP. You can return a list of references even though the filter specifies an attribute value.

```
GET https://vcloud.example.com/api/query?type=adminUser&format=references&filter=ldapUser==true
```

Response:

```
<?xml version="1.0" encoding="UTF-8"?>
<UserReferences
   xmlns="http://www.vmware.com/vcloud/v1.5"
   total="2"
   pageSize="25"
   page="1"
   name="user"
   type="application/vnd.vmware.vcloud.query.references+xml"
   href="https://vcloud.example.com/api/query?
type=adminUser&amp;page=1&amp;pageSize=25&amp;format=references&amp;filter=ldapUser==true"
   xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" ... >
   <Link
      rel="alternate"
      type="application/vnd.vmware.vcloud.query.records+xml"

href="https://vcloud.example.com/api/querytype=adminUser&amp;page=1&amp;pageSize=25&amp;format=re
cords&amp;filter=ldapUser==true" />
   <UserReference
      type="application/vnd.vmware.admin.user+xml"
      name="bob"
      href="https://vcloud.example.com/api/admin/user/39" />
   <UserReference
      type="application/vnd.vmware.admin.user+xml"
      name="zorro"
      href="https://vcloud.example.com/api/admin/user/58" />
</UserReferences>
```

Packaged Queries

Packaged queries have well-known URLs and can accept most of the parameters used with typed queries.

Query Syntax

Packaged queries have the following syntax:

`API-URL/query-url[?param][&param ... ][&filter]`

- *API-URL* is a URL of the form `https://vcloud.example.com/api`.
- *query-url* is the packaged query URL.
- *param* is an optional query parameter. Zero or more parameters are allowed. See "Query Parameters," on page 211.
- *filter* is an optional filter expression. At most one filter expression is allowed. See "Filter Expressions," on page 212.

Query Categories

Packaged queries are divided into the following categories:

User queries
The queries have the form *API-URL*/*object-type*/query. Any user can run these queries.

Administrator queries
The queries have the form *API-URL*/admin/*object-type*/query. An organization administrator can run these queries.

Extension queries
The queries have the form *API-URL*/admin/extension/*object-type*/query. A system administrator can run these queries.

If you make a query whose result set you do not have rights to view, a response code of ACCESS_TO_RESOURCE_IS_FORBIDDEN (403) is returned.

Each query type returns its result set as an XML document in which objects are represented as elements and object properties are represented as attributes, pairing the name of the property with its value at the time the request was run. By default, result sets are returned in the `records` format, which shows all database records for each object. Most queries also support the `references` format, which returns a set of object references, including `name`, `type`, and `href` attributes. All queries that support the `records` format also support the `idrecords` format. For more information about the `format` parameter, see "Query Parameters," on page 211.

Table 9-2. Packaged Queries

Query URL	Result Set
API-URL/catalogs/query	All catalogs in your organization that you have rights to view or modify
API-URL/mediaList/query	All media that you can view or modify
API-URL/vAppTemplates/query	All vApp templates that you can view or modify
API-URL/vApps/query	All vApps that you can view or modify

Table 9-2. Packaged Queries (Continued)

Query URL	Result Set
API-URL/vms/query	All virtual machines that you can view or modify
API-URL/admin/groups/query	Groups in all organizations in the system
API-URL/admin/users/query	Users in all organizations in the system
API-URL/admin/strandedUsers/query	Stranded users in the organization. When you delete an LDAP group, users who were imported from that group become stranded and cannot log in.
API-URL/admin/roles/query	All roles defined in the system
API-URL/admin/rights/query	All rights defined in the system
API-URL/admin/orgs/query	All organizations visible to you
API-URL/admin/vdcs/query	All vDCs in the system
API-URL/admin/extension/hostReferences/query	ESX/ESXi hosts registered to the system
API-URL/admin/extension/datastores/query	All datastores in the system
API-URL/admin/extension/externalNetworkReferences/query	External networks in the system
API-URL/admin/extension/networkPoolReferences/query	All network pools in the system
API-URL/admin/extension/providerVdcReferences/query	All provider vDCs in the system
API-URL/admin/extension/vimServerReferences/query	All vCenter servers registered to the system
API-URL/admin/extension/orgNetworks/query	All organization networks in the system
API-URL/admin/extension/vapps/query	All vApps in the system
API-URL/admin/extension/orgVdcs/query	All vDCs in the system

Examples

Simple packaged query using the records format, which is the default.

```
GET https://vcloud.example.com/api/catalogs/query
```

Response:

```
<QueryResultRecords
   xmlns="http://www.vmware.com/vcloud/v1.5"
   total="15"
   pageSize="25"
   page="1"
   name="catalog"
   type="application/vnd.vmware.vcloud.query.records+xml"
   href="https://vcloud.example.com/api/catalogs/query?page=1&amp;pageSize=25&amp;format=records"
   xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" ... >
   <Link
      rel="alternate"
      type="application/vnd.vmware.vcloud.query.references+xml"
      href="https://vcloud.example.com/api/catalogs/query?
type=catalog&amp;page=1&amp;pageSize=25&amp;format=references" />
   <CatalogRecord
      ownerName="system"
      organizationName="VMware"
      numberOfTemplates="30"
      numberOfMedia="3"
      name="VAM"
      isShared="true"
      isPublished="true"
      description=""
      createdOn="2011-03-21T14:28:09.273-07:00"
      href="https://vcloud.example.com/api/catalog/150" />
   <CatalogRecord
      ownerName="system"
      organizationName="QA"
      numberOfTemplates="0"
      numberOfMedia="1"
      name="QA-Cat"
      isShared="false"
      isPublished="true"
      description=""
      createdOn="2011-03-24T16:37:11.130-07:00"
      href="https://vcloud.example.com/api/catalog/78" />
   <CatalogRecord
      ownerName="system"
      organizationName="Org-d5443f6b-85e"
      numberOfTemplates="0"
      numberOfMedia="1"
      name="Catalog-3f79780c-6b0"
      isShared="true"
      isPublished="true"
      description=""
      createdOn="2011-03-25T11:27:56.063-07:00"
      href="https://vcloud.example.com/api/catalog/99" />
   <CatalogRecord
      ownerName="system"
      organizationName="Engineering"
      numberOfTemplates="2"
      numberOfMedia="4"
      name="TestCat"
      isShared="true"
      isPublished="true"
      description="New Catalog"
      createdOn="2011-03-22T17:10:10.067-07:00"
      href="https://vcloud.example.com/api/catalog/43" />
   <CatalogRecord
      ownerName="system"
      organizationName="Engineering"
      numberOfTemplates="8"
      numberOfMedia="1"
      name="catalog1"
      isShared="true"
      isPublished="true"
      description=""
      createdOn="2011-03-22T16:14:06.360-07:00"
      href="https://vcloud.example.com/api/catalog/91" />
</QueryResultRecords>
```

Packaged query using references format.

```
GET https://vcloud.example.com/api/catalogs/query?format=references
```

Response:

```
<CatalogReferences
   xmlns="http://www.vmware.com/vcloud/v1.5"
   total="15"
   pageSize="25"
   page="1"
   name="catalog"
   type="application/vnd.vmware.vcloud.query.references+xml"
   href="https://vcloud.example.com/api/catalogs/query?
page=1&amp;pageSize=25&amp;format=references"
   xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" ... >
   <Link
      rel="alternate"
      type="application/vnd.vmware.vcloud.query.records+xml"
      href="https://vcloud.example.com/api/catalogs/query?
type=catalog&amp;page=1&amp;pageSize=25&amp;format=records" />
    <CatalogReference
      type="application/vnd.vmware.vcloud.catalog+xml"
      name="VAM"
      href="https://vcloud.example.com/api/catalog/150" />
   <CatalogReference
      type="application/vnd.vmware.vcloud.catalog+xml"
      name="QA-Cat"
      href="https://vcloud.example.com/api/catalog/78" />
   <CatalogReference
      type="application/vnd.vmware.vcloud.catalog+xml"
      name="Catalog-3f79780c-6b0"
      href="https://vcloud.example.com/api/catalog/99" />
   <CatalogReference
      type="application/vnd.vmware.vcloud.catalog+xml"
      name="TestCat"
      href="https://vcloud.example.com/api/catalog/43" />
   <CatalogReference
      type="application/vnd.vmware.vcloud.catalog+xml"
      name="catalog1"
      href="https://vcloud.example.com/api/catalog/91" />
</CatalogReferences>
```

Packaged query with sorting and filtering. This query adds parameters and a filter expression to produce a list of catalogs that contain one or more vApp templates. The query sorts the result set in ascending order by number of vApp templates:

```
GET https://vcloud.example.com/api/catalogs/query?
format=records&sortAsc=numberOfTemplates&filter=numberOfTemplates!=0
```

Response:

```
<QueryResultRecords
   xmlns="http://www.vmware.com/vcloud/v1.5"
   total="3"
   pageSize="25"
   page="1"
   name="catalog"
   type="application/vnd.vmware.vcloud.query.records+xml"
   href="https://vcloud.example.com/api/catalogs/query?
page=1&amp;pageSize=25&amp;format=records&amp;filter=numberOfTemplates!
=0&amp;sortAsc=numberOfTemplates"
```

```
   xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" ... >
   <Link
      rel="alternate"
      type="application/vnd.vmware.vcloud.query.references+xml"
      href="https://vcloud.example.com/api/catalogs/query?
type=catalog&amp;page=1&amp;pageSize=25&amp;format=references&amp;filter=numberOfTemplates!
=0&amp;sortAsc=numberOfTemplates" />
   <CatalogRecord
      ownerName="system"
      organizationName="Engineering"
      numberOfTemplates="2"
      numberOfMedia="4"
      name="TestCatalog"
      isShared="true"
      isPublished="true"
      description="New Catalog"
      createdOn="2011-03-22T17:10:10.067-07:00"
      href="https://vcloud.example.com/api/catalog/43" />
   <CatalogRecord
      ownerName="system"
      organizationName="Engineering"
      numberOfTemplates="8"
      numberOfMedia="1"
      name="catalog1"
      isShared="true"
      isPublished="true"
      description=""
      createdOn="2011-03-22T16:14:06.360-07:00"
      href="https://vcloud.example.com/api/catalog/91" />
   <CatalogRecord
      ownerName="system"
      organizationName="VMware"
      numberOfTemplates="30"
      numberOfMedia="3"
      name="VAM"
      isShared="true"
      isPublished="true"
      description=""
      createdOn="2011-03-21T14:28:09.273-07:00"
      href="https://vcloud.example.com/api/catalog/150" />
</QueryResultRecords>
```

Query Parameters

Query parameters specify result set properties such as pagination, sort order, and filter criteria.

Query Parameters

Typed queries must include a type parameter, which specifies the type of query to run. Packaged queries cannot specify a type parameter. All other parameters are optional. If a parameter is omitted, a default value is assumed.

Table 9-3. Query Parameters

Parameter Name	Parameter Description	Default
`type`	The type of the query. Type names are case-sensitive. See "Query Types," on page 202.	None. This parameter is required for all typed queries, and is not allowed for any packaged query.
`sortAsc=`*attribute-name*	Sort results by *attribute-name* in ascending order.	Sorted by database ID
`sortDesc=`*attribute-name*	Sort results by *attribute-name* in descending order.	Sorted by database ID
`page`	If the query result spans multiple pages, return this page.	`page=1`
`pageSize`	Number of records per page. If this number exceeds configured `maxPageSize`, then only `maxPageSize` records are returned.	`defaultPageSize`
`format`	One of the following types: **`references`** Returns a reference to each object, including its `name`, `type`, and `href` attributes. **`records`** Returns all database records for each object, with each record as an attribute. **`idrecords`** Identical to the `records` format, except that object references are returned in `id` format rather than `href` format.	`format=records`
`fields`	Comma-separated list of attribute names to return	Returns all attributes
`offset`	Integer value specifying the first record to return. Record numbers < `offset` are not returned.	`offset=0`
`filter`	Filter expression. See Table 9-4	None

Filter Expressions

You can filter query results by appending an optional filter to your query request. A filter comprises one or more subexpressions drawn from the following set of operators.

Table 9-4. Query Filter Expressions

Operator	Example	Operation
`==`	*attribute*`==`*value*	Matches. The example evaluates to true if *attribute* matches *value* in a case-insensitive comparison. Asterisk (*) characters that appear anywhere in *value* are treated as wildcards that match any character string.
`!=`	*attribute*`!=`*value*	Does not match. The example evaluates to true if *attribute* does not match *value* in a case-insensitive comparison. Wildcard characters not allowed.
`;`	*attribute1*`==`*value1*`;`*attribute2*`!=`*value2*	Logical AND. The example evaluates to true only if *attribute1* matches *value1* and *attribute2* does not match *value2*
`,`	*attribute1*`==`*value1*`,`*attribute2*`==`*value2*	Logical OR. The example evaluates to true if either *attribute1* matches *value1* or *attribute2* matches *value2*

Group filter subexpressions with parentheses. Separate grouped subexpressions with a semicolon (no spaces).

```
(attribute1==value1;attribute2!=value2);(attribute3==value3;attribute4!=value4)...
```

For example:

```
https://vcloud.example.com/api/query?
type=providerVdcResourcePoolRelation&format=records&filter=(numberOfVMs!=0;isPrimary==true)
```

Attribute Names and Values

Several parameters and all filter expressions require you to specify an attribute name. To find the list of attribute names for a particular result set use the schema reference. You can search for result set elements by entering the term reference, for result sets where `format=references`, or record, for result sets where `format=records`, in the **Quick Index** window.

Configuring and Using Blocking Tasks and Notifications 10

vCloud Director allows a system administrator to configure many operations as blocking tasks, which are suspended until a system administrator acts on them or a preconfigured timer expires. Blocking tasks also generate AMQP messages that you can use to automate the handling of the underlying user request. A system administrator can also enable nonblocking AMQP notifications of all system events.

When a user requests an operation that has been configured as a blocking task, the system sends a message about the task to the configured AMQP broker. The system also creates a reference to the task in the cloud's `BlockingTaskReferences` container. A system administrator can retrieve the list of `BlockingTask` elements by making a GET request to the system's `blockingTasks` link, or to a URL included in the AMQP message.

About AMQP

AMQP, the Advanced Message Queuing Protocol, is an open standard for message queuing that supports flexible messaging for enterprise systems. vCloud Director includes an AMQP service and defines a set of events that, when notifications are enabled, trigger publication of messages by this service. A cloud operator can configure the service to work with RabbitMQ and other AMQP brokers to provide a stream of notifications about events in the cloud. By configuring specific tasks as blocking and writing AMQP clients that process the messages generated when these tasks are launched, cloud operators can create a programmatic facility for reviewing and acting on tenant requests.

By default, the vCloud Director AMQP service sends unencrypted messages. If you configure it to encrypt these messages using SSL, it verifies the broker's certificate by using the default JCEKS trust store of the Java runtime environment on the vCloud Director server. The Java runtime environment is typically located in the `$JRE_HOME/lib/security/cacerts` directory.

To use SSL with the vCloud Director AMQP service, select **Use SSL** on the AMQP Broker Settings section of the Blocking Tasks page of the vCloud Director Web console provide an SSL certificate pathname or JCEKS trust store pathname and password. If you do not want to validate certificates, because you trust all the computers that connect to the vCloud Director AMQP service, you can select **Accept all certificates**.

For more information about AMQP, see http://www.amqp.org.

Subscribing to Notifications

Notifications of system events are sent to the AMQP message broker that was configured in the system AMQP settings. AMQP client programs can connect to the broker and specify components of the AMQP routing key to indicate their interest in messages based on content. For example, a client can use the routing key to request the broker to send it all messages from a specific organization, or all messages that indicate a failed task. See "Routing Key Format," on page 219.

Processing Messages from Blocking Tasks

Messages from blocking tasks are also sent to the configured message broker, and clients can use the routing key to indicate their interest in these messages. See "Subscribing to Notifications," on page 215. Messages from blocking tasks contain additional information about the task itself. Clients that process these messages can use the vCloud API to authenticate to the system and act on the blocked task.

This chapter includes the following topics:

Configure Notifications and AMQP Settings

The system administrator can enable or disable AMQP notification messages for events in a cloud. The system administrator can also configure settings that the vCloud Director AMQP service uses when it sends messages generated by notifications and blocking tasks.

AMQP broker settings are established when you install and configure RabbitMQ or another AMQP broker to use with vCloud Director. These values include the following items:

- The fully-qualified domain name of the RabbitMQ server host, for example `amqp.example.com`.
- A username and password that are valid for authenticating with RabbitMQ.
- The port at which the broker listens for messages. The default is `5672`.
- The RabbitMQ virtual host. The default is "`/`".

NOTE It is a good practice to test the AMQP settings before you change the configuration. See "Test AMQP Settings," on page 218.

Prerequisites

Verify that you are logged in as a system administrator.

Procedure

1. Retrieve the `SystemSettings` element.
2. Examine the response to locate the links that you can use to retrieve the system's `NotificationsSettings` and `AmqpSettings` elements.

   These links have a `rel` attribute value of `down` and a `type` attribute value of `application/vnd.vmware.admin.notificationsSettings+xml` for `NotificationsSettings` or `application/vnd.vmware.admin.amqpSettings+xml` for `AmqpSettings`, as shown here:

   ```
   <Link href="https://vcloud.example.com/api/admin/extension/settings/amqp"
      rel="down"
      type="application/vnd.vmware.admin.amqpSettings+xml"/>
   ```

3 Enable or disable notifications.

  a Retrieve the `NotificationsSettings` element.

  Make a GET request to the `href` value of the `application/vnd.vmware.admin.notificationsSettings+xml` link.

  b Modify the value of the `EnableNotifications` element to enable or disable notifications.

  c Update the modified element with the new contents.

  PUT the modified element to the `href` value of its `rel="edit"` link.

4 Review or modify system AMQP settings.

  a Retrieve the `AmqpSettings` element.

  Make a GET request to the `href` value of the `application/vnd.vmware.admin.amqpSettings+xml` link described in Step 2.

  b Modify the contents of this element as necessary.

  See the schema reference for details of element contents.

  c Update the modified element with the new contents.

  PUT the modified element to the `href` value of its `rel="edit"` link. See "Example: Update AMQP Settings," on page 217

Example: Update AMQP Settings

This request modifies the AMQP settings for a cloud to require the use of SSL for AMQP connections, and to change the AMQP exchange type from `amq.direct` to `amq.fanout`. RabbitMQ declares these exchange types by default. For more information, see the RabbitMQ documentation.

Request:

```
PUT https://vcloud.example.com/api/admin/extension/settings/amqp
Content-Type: application/vnd.vmware.admin.amqpSettings+xml
...
<?xml version="1.0" encoding="UTF-8"?>
<AmqpSettings
   xmlns="http://www.vmware.com/vcloud/extension/v1.5">
   <AmqpHost />
   <AmqpPort>5672</AmqpPort>
   <AmqpUsername>guest</AmqpUsername>
   <AmqpPassword>Pa55w0rd</AmqpPassword>
   <AmqpExchange>amq.direct</AmqpExchange>
   <AmqpVHost>/</AmqpVHost>
   <AmqpUseSSL>true</AmqpUseSSL>
</AmqpSettings>
```

The response echoes the request, and contains a link you can use to test the settings. The value of `AmqpPassword`, which you must supply when you modify this element, is never returned when you retrieve it.

Response:

```
200 OK
Content-Type: application/vnd.vmware.admin.amqpSettings+xml
...
<AmqpSettings
   xmlns="http://www.vmware.com/vcloud/extension/v1.5"
   type="application/vnd.vmware.admin.amqpSettings+xml"
   href="https://vcloud.example.com/api/admin/extension/settings/amqp"
   ... >
   <vcloud:Link
      rel="test"
      type="application/vnd.vmware.admin.amqpSettingsTest+xml"
      href="https://vcloud.example.com/api/admin/extension/settings/amqp/action/test"/>
   <AmqpHost />
   <AmqpPort>5672</AmqpPort>
   <AmqpUsername>guest</AmqpUsername>
   <AmqpPassword />
   <AmqpExchange>amq.fanout</AmqpExchange>
   <AmqpVHost>/</AmqpVHost>
   <AmqpUseSSL>true</AmqpUseSSL>
</AmqpSettings>
```

Test AMQP Settings

The `settings/amqp/action/test` link of the `AmqpSettings` element allows you to test AMQP settings before configuring them for the cloud.

Prerequisites

- Verify that you are logged in as a system administrator.
- Verify that you know the AMQP broker password.
- Retrieve the `SystemSettings` element. See "Retrieve or Update System Settings," on page 165.

Procedure

1 Examine the `SystemSettings` element to locate the link that you can use to retrieve the system's `AmqpSettings` element.

   This link has a `rel` attribute value of `down` and a `type` attribute value of `application/vnd.vmware.admin.amqpSettings+xml` as shown here:

   ```
   <Link href="https://vcloud.example.com/api/admin/extension/settings/amqp"
      rel="down"
      type="application/vnd.vmware.admin.amqpSettings+xml"/>
   ```

2 Retrieve the `AmqpSettings` element and locate the `settings/amqp/action/test` link it contains.

   The response portion of "Example: Update AMQP Settings," on page 217 includes this link.

3 Create a new `AmqpSettings` element that contains the values you want to test.

   You can use the existing `AmqpSettings` element as a template. Whether you want to test the existing values or create new ones, you must include the AMQP broker password in the `AmqpPassword` element. This element is always returned empty when you retrieve the system's `AmqpSettings`.

4 Test the AMQP settings.

   POST the `AmqpSettings` element to the `settings/amqp/action/test` link described in Step 2.

Example: Test AMQP Settings

This example tests the settings shown in the request portion of "Example: Update AMQP Settings," on page 217

Request:

```
POST https://vcloud.example.com/api/admin/extension/settings/amqp/action/test
Content-Type: application/vnd.vmware.admin.amqpSettings+xml
...
<?xml version="1.0" encoding="UTF-8"?>
<AmqpSettings
   xmlns="http://www.vmware.com/vcloud/extension/v1.5"
   type="application/vnd.vmware.admin.amqpSettings+xml">
   <AmqpHost />
   <AmqpPort>5672</AmqpPort>
   <AmqpUsername>guest</AmqpUsername>
   <AmqpPassword>Pa55w0rd</AmqpPassword>
   <AmqpExchange>systemExchange</AmqpExchange>
   <AmqpVHost>/</AmqpVHost>
   <AmqpUseSSL>true</AmqpUseSSL>
</AmqpSettings>
```

The response is an `AmqpSettingsTest` element whose `Valid` element contains a Boolean indication of whether the settings are valid. This response indicates that they are. If a value in the POSTed `AmqpSettings` element is incorrect, the `AmqpSettingsTest` response has a `Valid` value of `false`.

Response:

```
200 OK
Content-Type: application/vnd.vmware.admin.amqpSettingsTest+xml
...
<vmext:AmqpSettingsTest
   xmlns:vmext="http://www.vmware.com/vcloud/extension/v1.5"
   xmlns:vcloud="http://www.vmware.com/vcloud/v1.5"
   type="application/vnd.vmware.admin.amqpSettingsTest+xml"
   href="https://vcloud.example.com/api/admin/extension/settings/amqp/action/test"
   ... >
   <vcloud:Link
      rel="up"
      type="application/vnd.vmware.admin.amqpSettings+xml"
      href="https://vcloud.example.com/api/admin/extension/settings/amqp" />
   <vmext:Valid>true</vmext:Valid>
</vmext:AmqpSettingsTest>
```

Notification Message Format

All messages that the vCloud Director AMQP service sends contain an AMQP routing key and a `Notification` element.

The `Notification` element is defined in the vCloud API schema. The routing key format is defined by the AMQP specification.

Routing Key Format

The routing key for a vCloud Director AMQP message has the following form:

operationSuccess.entity.org.user.subType1.subType2...subTypeN.[*taskName*]

Routing key components include:

operationSuccess A Boolean value denoting whether the operation that triggered the notification succeeded or failed.

entity The object identifier of the object on which an operation, an event of type `com/vmware/vcloud/event/`, triggered the notification. For more information about object identifiers, see "Objects, References, and Representations," on page 11

org The object identifier of the organization that owns the affected object.

user The object identifier of the user who made the request.

subType1-subType*N* Each *subType* is a single component of the event type name. See "Notification Types," on page 221.

taskName If *entity* is a task or blocking task, the task name is appended to the routing key.

The following routing key, in which the object identifiers are truncated to save space, is an example of a routing key that might have been created for a successful `com/vmware/vcloud/event/vapp/create` event:

```
true.dc6a-xxx.0b8a-xxx.832c-xxx.com.vmware.vcloud.event.vapp.create
```

Notification Headers

The vCloud API defines notification headers and prepends them to every notification.

Table 10-1. Notification Headers

Header	Value
notification.type	See "Notification Types," on page 221.
notification.entityType	The type of vCloud entity is associated with this notification. For example, `vm`.
notification.entityUUID	The object identifier of the object on which an operation, an event of type com/vmware/vcloud/event/, triggered the notification.
notification.orgUUID	The object identifier of the organization that owns the affected object.
notification.userUUID	The object identifier of the user who made the request.
notification.operationSuccess	A Boolean value denoting whether the operation that triggered the notification succeeded or failed.

Example: Notification Message Format

In addition to the routing key, a notification message contains a `Notification` element. This one was generated by a blocking task.

```
<vmext:Notification
   xmlns:vmext="http://www.vmware.com/vcloud/extension/v1.5"
   type="com/vmware/vcloud/event/blockingtask/create"
   eventId="a1440dd8-60ae-46c7-b216-44693bc00c90">
   <vmext:Link
      rel="entityResolver"
      href="https://vcloud.example.com/api/entity/" />
   <vmext:EntityLink
      rel="entity"
      type="vcloud:blockingTask"
      name="vdcComposeVapp"
      id="urn:vcloud:blockingTask:25" />
   <vmext:EntityLink
      rel="down"
      type="vcloud:user"
      name="vcloud"
      id="urn:vcloud:user:44" />
   <vmext:EntityLink
      rel="up"
      type="vcloud:org"
      name="Default"
      id="urn:vcloud:org:70" />
   <vmext:EntityLink
      rel="task"
      type="vcloud:task"
      name="vdcComposeVapp"
      id="urn:vcloud:task:34" />
   <vmext:EntityLink
      rel="task:owner"
      type="vcloud:vapp"
      id="urn:vcloud:vapp:26" />
   <vmext:Timestamp>2011-06-18T14:33:27.787+03:00</vmext:Timestamp>
   <vmext:OperationSuccess>true</vmext:OperationSuccess>
</vmext:Notification>
```

A `Notification` contains an `entityResolver` URL and `EntityLink` elements that provide more information about the `entity`, `org`, and `user` components of the routing key. Attributes of those elements show the `name`, `type`, and `id` of each component. After you authenticate to the cloud as a system administrator, you can retrieve any of the entities represented in an `EntityLink` by making a GET request to a URL you create by appending the value of an `id` attribute to the `entityResolver` URL. See "Retrieve an Object as an Entity," on page 237.

This request retrieves the blocking task that generated the `Notification` in this example.

```
GET https://vcloud.example.com/api/entity/urn:vcloud:blockingtask:25
```

The response to this request is identical to the one shown in the response portion of "Example: Handling a Blocking Task," on page 230.

Notification Types

The value of the `type` attribute of a vCloud Director notification is a string with the form `com/vmware/vcloud/event/`*`object-type`*`/`*`event-type`*.

Table 10-2. User, Group, Role, and Session Events

Type (com/vmware/vcloud/event/)	Description
session/login	A login session was created.
user/import	A user was imported from LDAP.
user/remove	An imported user was removed from the organization.
user/modify	One or more properties of a user were modified.
user/lockout	An account was locked based on the organization's password policy settings.
user/unlock	A locked account was unlocked.
user/lock_expired	The lock on an account has expired.
user/create	A local user was created in an organization.

Table 10-2. User, Group, Role, and Session Events (Continued)

Type (com/vmware/vcloud/event/)	Description
user/delete	A local user was removed from the organization.
group/import	A group was imported from LDAP.
group/remove	A group was removed from an organization.
role/create	A new role was created.
role/modify	An existing role was modified.
role/delete	A role was deleted.

Table 10-3. Organization, Network, Catalog, and vDC Events

Type (com/vmware/vcloud/event/)	Description
org/create	An organization was created.
org/modify	An organization was modified.
org/delete	An organization was deleted.
network/create	A network was created.
network/modify	A network was modified.
network/delete	A network was deleted.
network/deploy	A network was deployed.
network/undeploy	A network was undeployed.
catalog/create	A catalog was created.
catalog/delete	A catalog was deleted.
catalog/modify	One or more properties of a catalog were modified
catalog/publish	A catalog was published.
catalogItem/create	An item was added to a catalog.
catalogItem/delete	An item was removed from a catalog.
vdc/create_request	A request to create a vDC was blocked pending administrative action.
vdc/create	A vDC was created.
vdc/modify	One or more properties of a vDC was modified.
vdc/delete_request	A request to delete a vDC was blocked pending administrative action.
vdc/delete	A vDC was deleted.
vdc/fast_provisioning/modify	The `UsesFastProvisioning` value of a vDC was modified.
vdc/thin_provisioning/modify	The `IsThinProvision` value of a vDC was modified.

Table 10-4. vApp, vApp Template, Vm, and Media Events

Type (com/vmware/vcloud/event/)	Description
vappTemplate/create	A vApp template was created.
vappTemplate/import	A virtual machine was imported from vSphere as a vApp template.
vappTemplate/modify	One or more properties of a vApp template were modified.
vappTemplate/delete	A vApp template was deleted.

Table 10-4. vApp, vApp Template, Vm, and Media Events (Continued)

Type (com/vmware/vcloud/event/)	Description
vappTemplate/create_request	A request to create a vApp template was blocked pending administrative action.
vappTemplate/import_request	A request to import a vApp template was blocked pending administrative action.
vappTemplate/modify_request	A request to modify a vApp template was blocked pending administrative action.
vappTemplate/delete_request	A request to delete a vApp template was blocked pending administrative action.
vapp/create	A vApp was created (instantiated)
vapp/import	A virtual machine was imported from vSphere as a vApp.
vapp/modify	One or more properties of a vApp were modified.
vapp/delete	A vApp was deleted.
vapp/deploy	A vApp was deployed.
vapp/undeploy	A vApp was undeployed.
vapp/runtime_lease_expiry	The runtime lease of a vApp has expired.
vapp/create_request	A request to instantiate a vApp template was blocked pending administrative action.
vapp/import_request	A request to import a vApp was blocked pending administrative action.
vapp/modify_request	A request to modify a vApp was blocked pending administrative action.
vapp/delete_request	A request to delete a vApp was blocked pending administrative action.
vapp/deploy_request	A request to deploy a vApp was blocked pending administrative action.
vapp/undeploy_request	A request to undeploy a vApp was blocked pending administrative action.
vm/create_request	A request to create a virtual machine was blocked pending administrative action.
vapp/quarantine_reject	An uploaded OVF was rejected after quarantine.
vapp/upload_timeout	An OVF upload has timed out.
vm/create	A virtual machine was created by instantiating a vApp.
vm/modify_request	A request to modify a virtual machine was blocked pending administrative action.
vm/modify	One or more properties of a virtual machine were modified.
vm/delete	A virtual machine was deleted.
vm/change_state	The power state of a virtual machine has changed.
vm/deploy_request	A request to deploy a virtual machine was blocked pending administrative action.
vm/deploy	A virtual machine was deployed.
vm/undeploy_request	A request to undeploy a virtual machine was blocked pending administrative action.
vm/undeploy	A virtual machine was undeployed.

Table 10-4. vApp, vApp Template, Vm, and Media Events (Continued)

Type (com/vmware/vcloud/event/)	Description
vm/consolidate_request	A request to consolidate a virtual machine was blocked pending administrative action.
vm/consolidate	A virtual machine was consolidated.
vm/relocate_request	A request to relocate a virtual machine was blocked pending administrative action.
vm/relocate	A virtual machine was relocated.
media/create	A media object was created by upload or import.
media/import	A media object was imported.
media/modify	One or more properties of a media object were modified.
media/delete	A media object was deleted.
media/create_request	A request to create a media object was blocked pending administrative action.
media/import_request	A request to import a media object was blocked pending administrative action.
media/modify_request	A request to modify a media object was blocked pending administrative action.
media/delete_request	A request to delete a media object was blocked pending administrative action.
media/upload_timeout	A media upload has timed out.
media/quarantine_reject	An uploaded media object was rejected after quarantine.

Table 10-5. Other System Events

Type (com/vmware/vcloud/event/)	Description
providerVdc/create_request	A request to create a provider vDC was blocked pending administrative action.
providerVdc/create	A provider vDC was created.
providerVdc/modify	One or more properties of a provider vDC were modified.
providerVdc/delete_request	A request to delete a provider vDC was blocked pending administrative action.
providerVdc/delete	A provider vDC was deleted.
vc/create	A vCenter server was registered.
vc/modify	One or more properties of a registered vCenter server were modified.
vc/delete	A registered vCenter server was registered.
task/create	A task was created
task/start	A non-blocking task has started or a blocking task has resumed
task/abort	A task was aborted
task/fail	A task has failed
blockingtask/create	A task was blocked and a notification created.
blockingtask/resume	A blocking task was resumed.
blockingtask/abort	A blocking task was aborted.
blockingtask/fail	A blocking task was failed.

Table 10-5. Other System Events (Continued)

Type (com/vmware/vcloud/event/)	Description
datastore/modify	One or more properties of a datastore object were modified
datastore/delete	A datastore object was deleted.

Retrieve or Update Blocking Task Settings

Timeout settings, default actions, and related messages for blocking tasks are properties of a cloud. They apply to all organizations in the cloud. Only a system administrator can view or modify them.

When a user requests an operation that is configured to create a blocking task, the system creates a reference to the operation in the cloud's `BlockingTaskReferences` container. The system also sends a message about the task to the configured AMQP broker. A system administrator can retrieve the list of `BlockingTaskReferences` by making a GET request to the system's `blockingTasks` link. An AMQP client can use information in the message to construct a URL that it can use to retrieve the task. See "Example: Notification Message Format," on page 220.

If no action is taken on the blocking task within a specified timeout interval, it is subject to a default action. You can specify the timeout interval and default action for all blocking tasks by modifying the system's `BlockingTaskSettings` element. To configure an operation as a blocking task, add the operation name to the `BlockingTaskOperations` element contained by `BlockingTaskSettings`. See "Task Operations," on page 226.

Prerequisites

- Verify that you are logged in as a system administrator.
- Retrieve the `SystemSettings` element. See "Retrieve or Update System Settings," on page 165.

Procedure

1. Examine the response to locate the link that you can use to retrieve the system's `BlockingTaskSettings` element.

   This link has a `rel` attribute value of `down` and a `type` attribute value of `application/vnd.vmware.admin.blockingTaskSettings+xml`, as shown here:

   ```
   <Link href="https://vcloud.example.com/api/admin/extension/settings/blockingTask"
      rel="down"
      type="application/vnd.vmware.admin.blockingTaskSettings+xml"/>
   ```

2. Retrieve the element.

   Make a GET request to the `href` value of the link.

3. (Optional) Modify the element as needed to change the settings it controls.

   See the schema reference.

4. (Optional) Update the modified element with the new contents.

   PUT the modified element to the `href` value of its `rel="edit"` link. See "Example: Update Blocking Task Settings," on page 225.

Example: Update Blocking Task Settings

This request modifies the blocking task settings for a cloud to set the time-out period to 24 hours and adds media upload as an operation that creates a blocking task. See "Task Operations," on page 226 for a list of operation names.

Request:

```
PUT https://vcloud.example.com/api/admin/extension/settings/blockingTask
Content-Type: application/vnd.vmware.admin.blockingTaskSettings+xml
...
<?xml version="1.0" encoding="UTF-8"?>
<BlockingTaskSettings
   xmlns:vcloud="http://www.vmware.com/vcloud/v1.5"
   xmlns="http://www.vmware.com/vcloud/extension/v1.5">
   <TimeoutAction>abort</TimeoutAction>
   <BlockingTaskOperations>
      <vcloud:Operation>vdcUploadMedia</vcloud:Operation>
   </BlockingTaskOperations>
   <TimeoutInMilliseconds>86400000</TimeoutInMilliseconds>
</BlockingTaskSettings>
```

The response echoes the request, and adds the `href` attributes and `edit` links for the `BlockingTaskSettings` element and the `BlockingTaskOperations` element it contains.

Response:

```
200 OK
Content-Type: application/vnd.vmware.admin.blockingTaskSettings+xml
...
<BlockingTaskSettings
   xmlns="http://www.vmware.com/vcloud/v1.5"
   TimeoutInMilliseconds="86400000"
   type="application/vnd.vmware.admin.blockingTaskSettings+xmll"
   href="https://vcloud.example.com/api/admin/extension/settings/blockingTask"
   xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" ... >
   <Link
      rel="edit"
      type="application/vnd.vmware.admin.blockingTaskSettings+xml"
      href="https://vcloud.example.com/api/admin/extension/settings/blockingTask" />
   <Link
      rel="down"
      type="application/vnd.vmware.admin.taskOperationList+xml"
      href="https://vcloud.example.com/api/admin/extension/settings/blockingTask/operations"/>
   <TimeoutAction>abort</TimeoutAction>
   <BlockingTaskOperations
      type="application/vnd.vmware.admin.taskOperationList+xml"
      href="https://vcloud.example.com/api/admin/extension/settings/blockingTask/operations">
      <Link
         rel="edit"
         type="application/vnd.vmware.admin.taskOperationList+xml"
         href="https://vcloud.example.com/api/admin/extension/settings/blockingTask/operations"/>
      <Operation>vdcUploadMedia</Operation>
   </BlockingTaskOperations>
</BlockingTaskSettings>
```

Task Operations

Requests that you can configure as blocking tasks are represented by task operation names.

To configure a request type as a blocking task, place the operation name in an `Operation` element and add that element to the cloud's `BlockingTaskOperations` element. See "Retrieve or Update Blocking Task Settings," on page 225.

Table 10-6. Media

Operation Name	Description
vdcCopyMedia	Copy a media object.
vdcDeleteMedia	Delete a media object.
vdcUploadMedia	Upload media.
vdcUpdateMedia	Modify one or more properties of a media object.

Table 10-7. Network

Operation Name	Description
networkCreateNetworkPool	Create a network pool.
networkCreateExternalNetwork	Create an external network.
networkDelete	Delete a network.
networkDeleteNetworkPool	Delete a network pool.
networkCreateFencePoolTypeNetworkPool	Create an isolation-backed FencePoolType network pool.
networkUpdateNetwork	Modify one or more properties of a network object.
networkUpdateNetworkPool	Modify one or more properties of a network pool object.
networkUpdateVlanPool	Modify one or more properties of a VlanPoolType network pool object.

Table 10-8. vApp

Operation Name	Description
vappDeploy	Deploy a vApp.
vappPowerOff	Power-off a vApp.
vappRebootGuest	Reboot any virtual machine in a vApp.
vappReset	Reset any virtual machine in a vApp.
vappShutdownGuest	Shut down any virtual machine in a vApp.
vappSuspend	Suspend any virtual machine in a vApp.
vappUndeployPowerOff	Undeploy any virtual machine in a vApp by powering it off.
vappUndeploySuspend	Undeploy any virtual machine in a vApp by suspending it.
vappUpdateVm	Modify one or more properties of a virtual machine.
vappUpgradeHwVersion	Upgrade the hardware version of any virtual machine in a vApp.
vdcCaptureTemplate	Capture a vApp as a vApp template.
vdcComposeVapp	Compose a vApp.
vdcCopyVapp	Copy a vApp.
vdcDeleteVapp	Delete a vApp.
vdcInstantiateVapp	Instantiate a vApp template.
vdcRecomposeVapp	Recompose a vApp.
vdcUpdateVapp	Update or modify any section of a vApp.

Table 10-9. vApp Template

Operation Name	Description
vdcCopyTemplate	Copy a vApp template.
vdcDeleteTemplate	Delete a vApp template.
vdcEnableDownload	Enable a vApp template for download as OVF.
vdcUpdateTemplate	Modify one or more properties of a vApp template.
vdcUploadOvfContents	Upload an OVF package to create a vApp template.

Table 10-10. vDC

Operation Name	Description
rclCreateProviderVdc	Create a provider vDC.
rclDeleteProviderVdc	Delete a provider vDC.
vdcCreateVdc	Create an organization vDC.
vdcDeleteVdc	Delete an organization vDC.
vdcUpdateVdc	Modify one or more properties of an organization vDC.

Table 10-11. Import

Operation Name	Description
importMedia	Import a media object from vSphere.
importSingletonVapp	Import a virtual machine from vCenter as a vApp.
importSingletonTemplate	Import a virtual machine from vCenter as a vApp template.
importIntoExistingTemplate	Import a virtual machine from vCenter to an existing vApp template.
importIntoExistingVapp	Import a virtual machine from vCenter to an existing vApp.

Monitor Blocking Tasks

A system administrator can retrieve a list of all pending and active blocking tasks. Links in the returned `BlockingTask` element allow the administrator to take action on the request.

In addition to being subject to programmatic action by an AMQP client (see "Notification Message Format," on page 219), blocking tasks can be monitored and managed by a system administrator using the vCloud API.

Prerequisites

Verify that you are logged in as a system administrator.

Procedure

1 Retrieve the list of blocking tasks.

   See the request portion of "Example: Retrieve a List of Blocking Tasks," on page 229. If the `BlockingTaskReferences` element contains no `Reference` elements, no blocking tasks are currently active in the system

2 Retrieve an individual blocking task from one of the `Reference` elements in the response.

   See the request portion of "Example: Handling a Blocking Task," on page 230.

3 Use one of the action links in the BlockingTask to take action on the task.

See the response portion of "Example: Handling a Blocking Task," on page 230.

Example: Retrieve a List of Blocking Tasks

Request:

```
GET https://vcloud.example.com/api/admin/extension/blockingTasks/
```

Response:

```
200 OK
Content-Type: application/vnd.vmware.admin.blockingTaskList+xml
...
<vmext:BlockingTaskReferences
   xmlns:vmext="http://www.vmware.com/vcloud/extension/v1.5"
   xmlns:vcloud="http://www.vmware.com/vcloud/v1.5"
   type="application/vnd.vmware.admin.blockingTaskList+xml"
   xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
   ... >
   <vcloud:Link
      rel="up"
      type="application/vnd.vmware.admin.vmwExtension+xml"
      href="https://vcloud.example.com/api/admin/extension" />
   <vcloud:Reference
      type="application/vnd.vmware.admin.blockingTask+xml"
      name="vdcUpdateTemplate"
      href="https://vcloud.example.com/api/admin/extension/blockingTask/28" />
   <vcloud:Reference
      type="application/vnd.vmware.admin.blockingTask+xml"
      name="vdcComposeVapp"
      href="https://vcloud.example.com/api/admin/extension/blockingTask/25" />
   <vcloud:Reference
      type="application/vnd.vmware.admin.blockingTask+xml"
      name="vdcUploadMedia"
      href="https://vcloud.example.com/api/admin/extension/blockingTask/12" />
</vmext:BlockingTaskReferences>
```

Take Action on a Blocking Task

The BlockingTask element includes links that you can use to take action on a blocking task.

A BlockingTask element is primarily a collection of Link elements that allow you to take action on the task. When a user requests an operation that is configured to create a blocking task, the system sends a message about the task to the configured AMQP broker, and also creates a reference to the task in the cloud's BlockingTaskReferences container. A system administrator can retrieve the list of BlockingTask elements by making a GET request to the system's extension/blockingTasks link. See "Monitor Blocking Tasks," on page 228.

After authenticating to the cloud as a system administrator, the AMQP client can retrieve a blocking task. The AMQP client makes a GET request to a URL that the task creates by appending the value of the id attribute of the task to the entityResolver URL in the Notification. See "Example: Notification Message Format," on page 220.

The following actions are allowed:

resume Unblock the task and allow it to continue.

abort End the task, cleaning up any transient objects that it created. Task status is set to ABORTED.

fail End the task, setting the status of any transient objects that it created to ERROR. Task status is set to ERROR.

updateProgress Reset the timeout value and timeout action for an active task. Use this action to keep the task alive when it might become subject to a timeout action.

Prerequisites

Verify that you are logged in as a system administrator.

Procedure

1 Retrieve the list of active blocking tasks.

See "Monitor Blocking Tasks," on page 228. If you are using an AMQP client to handle blocking tasks, skip this step. Each blocking task creates its own AMQP message, which contains a reference to the `BlockingTask`.

2 Retrieve an individual `BlockingTask`.

See the request portion of "Example: Handling a Blocking Task," on page 230.

3 Make a request.

Action	Request
resume	POST a `BlockingTaskOperationParams` element to the `Link` where `rel="resume"`
abort	POST a `BlockingTaskOperationParams` element to the `Link` where `rel="abort"`
fail	POST a `BlockingTaskOperationParams` element to the `Link` where `rel="fail"`
updateProgress	POST a `BlockingTaskUpdateProgressParams` element to the `Link` where `rel="updateProgress"`

Example: Handling a Blocking Task

This request shows how to retrieve a blocking task without using an AMQP client. "Example: Notification Message Format," on page 220 shows how to retrieve the same task using information in the AMQP message.

Request:

```
GET https://vcloud.example.com/api/admin/extension/blockingTask/25
```

Response:

```
200 OK
<vmext:BlockingTask
   xmlns:vmext="http://www.vmware.com/vcloud/extension/v1.5"
   xmlns:vcloud="http://www.vmware.com/vcloud/v1.5"
   status="active"
   timeoutDate="2011-05-07T16:25:18.857+03:00"
   timeoutAction="abort"
   createdTime="2011-05-02T16:25:18.857+03:00"
   name="importSingletonTemplate"
   id="urn:vcloud:blockingTask:25"
   type="application/vnd.vmware.admin.blockingTask+xml"
   href="https://vcloud.example.com/api/admin/extension/blockingTask/25"
   xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
   ... >
   <vcloud:Link
      rel="resume"
      type="application/vnd.vmware.admin.blockingTaskOperationParams+xml"
      href="https://vcloud.example.com/api/admin/extension/blockingTask/25/action/resume" />
   <vcloud:Link
      rel="abort"
      type="application/vnd.vmware.admin.blockingTaskOperationParams+xml"
      href="https://vcloud.example.com/api/admin/extension/blockingTask/25/action/abort" />
   <vcloud:Link
      rel="fail"
      type="application/vnd.vmware.admin.blockingTaskOperationParams+xml"
      href="https://vcloud.example.com/api/admin/extension/blockingTask/25/action/fail" />
   <vcloud:Link
      rel="updateProgress"
      type="application/vnd.vmware.admin.BlockingTaskUpdateProgressOperationParams+xml"

href="https://vcloud.example.com/api/admin/extension/blockingTask/25/action/updateProgress" />
   <vcloud:Link
      rel="up"
      type="application/vnd.vmware.vcloud.task+xml"
      href="https://vcloud.example.com/api/task/42" />
   <vcloud:Organization
      type="application/vnd.vmware.admin.organization+xml"
      name="example"
      href="https://vcloud.example.com/api/admin/org/97" />
   <vcloud:User
      type="application/vnd.vmware.admin.user+xml"
      name="system"
      href="https://vcloud.example.com/api/admin/user/80" />
   <vcloud:TaskOwner
      type="application/vnd.vmware.vcloud.vAppTemplate+xml"
      name=""
      href="https://vcloud.example.com/api/vAppTemplate/vappTemplate-89" />
</vmext:BlockingTask>
```

The following request allows the task to resume with a message indicating administrative approval.

```
POST https://vcloud.example.com/api/admin/extension/blockingTask/25/action/resume
Content-Type: application/vnd.vmware.admin.blockingTaskOperationParams+xml
...
<?xml version="1.0" encoding="UTF-8"?>
<BlockingTaskOperationParams
   xmlns="http://www.vmware.com/vcloud/extension/v1.5" >
   <Message>Approved by system administrator.</Message>
</BlockingTaskOperationParams>
```

Extend The Timeout Expiration of an Active Task

You can use the updateProgress link in a BlockingTask to extend the expiration time of an active task.

Prerequisites

Verify that you are logged in as a system administrator.

Procedure

1. Retrieve the list of active blocking tasks.

   See "Monitor Blocking Tasks," on page 228. If you are using an AMQP client to handle task extension requests, skip this step. Each blocking task creates its own AMQP message, which contains a reference to the BlockingTask mentioned in Step 1.

2. Retrieve an individual BlockingTask.

   See the request portion of "Example: Handling a Blocking Task," on page 230.

3. Provide a new timeout value, relative to now, for the task.

   Create a BlockingTaskUpdateProgressParams element that specifies the number of milliseconds until the task times out. See "Example: Extend The Timeout Expiration of an Active Task," on page 232.

4. POST the BlockingTaskUpdateProgressParams to the updateProgress URL from the BlockingTask.

   The new timeout value is set to now (the time when the updateProgress request is executed) plusTimeoutValueInMilliseconds.

Example: Extend The Timeout Expiration of an Active Task

This request resets the expiration time of the BlockingTask shown in "Example: Handling a Blocking Task," on page 230 to ten minutes after the request is processed.

Request:

```
POST https://vcloud.example.com/api/admin/extension/blockingTask/34/action/updateProgress
Content-Type: application/vnd.vmware.admin.blockingTaskUpdateProgressOperationParams+xml
...
<?xml version="1.0" encoding="UTF-8"?>
<BlockingTaskUpdateProgressParams
   xmlns="http://www.vmware.com/vcloud/extension/v1.5">
   <Message>Giving you ten more minutes...</Message>
   <TimeoutValueInMilliseconds>600000</TimeoutValueInMilliseconds>
</BlockingTaskUpdateProgressParams>
```

The response includes the entire BlockingTask and shows the new value of the timeoutDate attribute. The value assumes that the request was made at time 2011-05-11T11:50:55. This example omits most of the response.

Response:

```
200 OK
...
<vmext:BlockingTask
   xmlns:vmext="http://www.vmware.com/vcloud/extension/v1.5"
   xmlns:vcloud="http://www.vmware.com/vcloud/v1.5"
   status="active"
   timeoutDate="2011-05-11T12:00:55.857+03:00"
   ...
</vmext:BlockingTask>
```

XML Representations in the vCloud API 11

The vCloud API represents objects in a cloud as XML documents in which object properties are encoded as elements and attributes with typed values and an explicit object hierarchy defined by an XML schema. Schema reference information is available as a download.

Client programs of RESTful Web services must be able to request object representations from the server, parse the server's responses to extract the information they contain, and compose requests that, in many cases, are based on the information extracted from a response. Developers of such clients must understand the structure of each representation that might be part of a request or response, and any requirements that the network protocol (HTTP) places on client-server interaction.

Schemas

Each vCloud API object is defined in an XML schema document. Some objects are defined in their own schema documents. Others are defined in the context of the larger schema in which they are used. Still others are defined in a common schema document, from which other schemas inherit. Schema files and reference information about all elements, types, and operations is included in the schema reference download. See "About the Schema Reference Downloadable Archive," on page 19.

vCloud Director uses a validating XML parser that requires elements in XML documents to agree in order and number with the schema. Required elements must appear in request bodies. All elements that appear in request bodies must appear in the order established by the schema, and with content that conforms to the type constraint specified in the schema. Default values, where defined, are supplied for elements that are empty. See "XML Namespace Identifiers," on page 234.

API Versioning

vCloud API schema version information appears in the values of the `xsi:schemaLocation` and `xmlns` attributes in a response document. For example, a response body that used schema version 1.5 would include the following attributes:

```
xmlns="http://www.vmware.com/vcloud/v1.5"
xsi:schemaLocation="https://vcloud.example.com/api/v1.5/schema/master.xsd"
```

To discover the schema versions that a server supports, a client can make an unauthenticated GET request to a well-known URL on the server, as shown in "Example: Versions Request and Response," on page 43.

Extensibility

The vCloud API is designed to be extensible. All complex types that the API defines extend a single abstract type. This type and all types based on it can include zero or more instances of an extension element that can contain an arbitrary number of elements and attributes.

VCloudExtensibleType is an abstract type that all complex types extend that are defined in the vCloud API namespace at http://www.vmware.com/vcloud/v1.5. This type allows you to add custom attributes to any type and supports definition of custom elements within the VCloudExtension element.

The VCloudExtension element has an attribute named required that specifies how clients and servers proceed when they see an unknown extension. All VCloudExtension elements are assumed to require a server that understands them. The required attribute is optional, but if omitted is assumed to be present with a value of true. This extensibility mechanism allows new servers to extend the XML representations native to the vCloud API without requiring existing clients to understand those extensions.

A client might encounter a VCloudExtension element in any response. If the element declares required="true" and the client does not know how to interpret the contents of the element, the client can ignore it, but it must include the VCloudExtension in any request to modify the element that contains it. A server must return a failure when a request includes a VCloudExtension element that declares required="true" but the server does not understand the extension. For more information about VCloudExtension, see the schema reference.

This chapter includes the following topics:

- "XML Namespace Identifiers," on page 234
- "Common vCloud API Attributes," on page 235
- "Retrieve an Object as an Entity," on page 237

XML Namespace Identifiers

Elements used as request or response bodies contain a set of attributes that enable XML validation. The body of a PUT or POST request must contain all XML namespace identifiers required to validate the elements it contains. A response body typically includes all the XML namespace identifiers that the server used to validate it, in addition to other attributes that specify the schema locations searched during validation.

The vCloud API uses these XML namespace identifier attributes and prefixes.

Table 11-1. XML Namespace Identifiers in the vCloud API

Name	Value	Requirement
xmlns	http://www.vmware.com/vcloud/v1.5	Required in all request bodies.
xmlns:vmext	http://www.vmware.com/vcloud/extension/v1.5	Required in request bodies that include elements from the vSphere platform extensions
xmlns:ve	http://www.vmware.com/schema/ovfenv	Required in request bodies that include an ovf:Environmentelement.
xmlns:ovf	http://schemas.dmtf.org/ovf/envelope/1	Required in request bodies that include elements defined in OVF schema dsp8023_1.1.0.xsd.
xmlns:rasd	http://schemas.dmtf.org/wbem/wscim/1/cim-schema/2/CIM_ResourceAllocationSettingData	Required in request bodies that include elements defined in OVF schema CIM_ResourceAllocationSettingData.xsd.
xmlns:oe	http://schemas.dmtf.org/ovf/environment/1	Required in request bodies that include elements defined in OVF schema dsp8027_1.1.0.xsd.
xmlns:vssd	http://schemas.dmtf.org/wbem/wscim/1/cim-schema/2/CIM_VirtualSystemSettingData	Not required in request bodies.

Table 11-1. XML Namespace Identifiers in the vCloud API (Continued)

Name	Value	Requirement
xsi:schemaLocation	An installation-dependent schema location search path. See http://www.w3.org/TR/xmlschema-0/.	Not required in request bodies.
xmlns:xsi	http://www.w3.org/2001/XMLSchema-instance	Not required in request bodies.

XML Namespace Prefixes in Request and Response Bodies

When a request or response includes elements from multiple XML namespaces, each element name is prefixed with a namespace identifier. Unless all elements in a request or response originate in the same XML namespace, these prefixes are required in request bodies, and are always included in response bodies.

The examples omit XML namespace identifiers from most responses. The following fragment shows how some of them appear in a typical response body.

```
<VApp
   xmlns="http://www.vmware.com/vcloud/v1.5"
   ...
   xmlns:ovf="http://schemas.dmtf.org/ovf/envelope/1"
   xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
   xsi:schemaLocation="http://www.vmware.com/vcloud/v1.5
https://vcloud.example.com/api/schema/v1.5/master.xsd">
   ...>
   ...
</VApp>
```

Common vCloud API Attributes

Most vCloud API objects have a number of common attributes. With the exception of `name`, none of these attributes are required in request bodies, and are ignored if included. All of them are included in response bodies.

Object Name

Every object requires a `name` attribute. The string value of this attribute is included in all object references, and can be used as the display name for the object. The value of `name` must be unique within a given scope.

Table 11-2. Requirements for Unique Object Names

Object Type	Name Scope
ProviderVdc	Cloud
Org	Cloud
Vdc	Organization
Catalog	Organization
CatalogItem	Catalog
vAppTemplate	None
vApp	Organization
Vm	vApp
Media	None
Network	Container (Organization, vApp, or cloud)

Object Identifier, Type, and Reference

These attributes are common to all object representations.

id
The object identifier, expressed in URN format. The value of the `id` attribute uniquely identifies the object, persists for the life of the object, and is never reused. The `id` attribute value is intended to provide a context-free identifier that can be used with the vCloud API `entityResolver` (see "Retrieve an Object as an Entity," on page 237), and is also suitable for use by clients that need to access the object using a different API.

type
The object type, specified as a MIME content type.

href
An object reference, expressed in URL format. Because this URL includes the object identifier portion of the `id` attribute value, it uniquely identifies the object, persists for the life of the object, and is never reused. The value of the `href` attribute is a reference to a view of the object, and can be used to access a representation of the object that is valid in a particular context. Although URLs have a well-known syntax and a well-understood interpretation, a client should treat each `href` as an opaque string. The rules that govern how the server constructs `href` strings might change in future releases.

Object Creation Status

Objects such as `VAppTemplate`, `VApp`, and `Vm`, that extend the `ResourceEntity` type have a `status` attribute whose value indicates the state of the object. In this table, YES indicates that a status value is allowed for the object listed in the column header. The `status` value for a `VAppTemplate` or `VApp`, which contain `Vm` objects that each have a `status` attribute of their own, is computed from the status of the contained objects.

Table 11-3. status Attribute

Value	Description	vAppTemplate	vApp	Vm
-1	The object could not be created.	YES	YES	YES
0	The object is unresolved.	YES	YES	YES
1	The object is resolved.	YES	YES	YES
2	The object is deployed.	No	No	No
3	The object is suspended.	No	YES	YES
4	The object is powered on.	No	YES	YES
5	The object is waiting for user input.	No	YES	YES
6	The object is in an unknown state.	YES	YES	YES
7	The object is in an unrecognized state.	YES	YES	YES
8	The object is powered off.	YES	YES	YES
9	The object is in an inconsistent state.	No	YES	YES
10	Children do not all have the same status.	YES	YES	No
11	Upload initiated, OVF descriptor pending.	YES	No	No
12	Upload initiated, copying contents.	YES	No	No
13	Upload initiated , disk contents pending.	YES	No	No
14	Upload has been quarantined.	YES	No	No
15	Upload quarantine period has expired.	YES	No	No

Retrieve an Object as an Entity

You can use the vCloud API entity resolver with an object's id attribute value to retrieve a context-free reference to the object.

Every first-class object that the vCloud API defines includes an id attribute whose value is the object identifier expressed in URN format. The value of the id attribute uniquely identifies the object, persists for the life of the object, and is never reused.

You can append the value of the id attribute to the vCloud API entityResolver URL to retrieve a context-free representation of the underlying object as an Entity element.

Prerequisites

Verify that you are logged in as a system administrator or member of an organization in the cloud.

Procedure

1 Retrieve the current Session object to get the entityResolver URL.

Use a request like this one:

```
GET https://vcloud.example.com/api/session
```

The response is a Session element like the one shown in "Example: Create a Login Session," on page 44. The Session element contains the entityResolver URL in the href of the Link element in this excerpt.

```
<Session ... >
  ...
    <Link
      rel="entityResolver"
      type="application/vnd.vmware.vcloud.entity+xml"
      href="https://vcloud.example.com/api/entity/" />
</Session>
```

2 Append the value of the object's id attribute to the entityResolver URL.

3 Make a GET request to the URL you created in Step 2

See the request portion of "Example: Using the entityResolver URL," on page 237.

Example: Using the entityResolver URL

This example retrieves the Vapp object shown in the excerpt "Example: Object id, type, and href Attributes," on page 12 as an Entity.

Request:

```
GET https://vcloud.example.com/api/entity/urn:vcloud:vapp:490af534-1491-452e-8ed6-a5eb54447dac
```

Response:

```
<Entity
   xmlns="http://www.vmware.com/vcloud/v1.5"
   id="urn:vcloud:vapp:490af534-1491-452e-8ed6-a5eb54447dac"
   name="urn:vcloud:vapp:490af534-1491-452e-8ed6-a5eb54447dac"
   type="application/vnd.vmware.vcloud.entity+xml"
   href="https://vcloud.example.com/api/entity/urn:vcloud:vapp:490af534-1491-452e-8ed6-
a5eb54447dac"
   xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" ...>
   <Link
```

```
      rel="alternate"
      type="application/vnd.vmware.vcloud.vApp+xml"
      href="https://vcloud.example.com/api/vApp/vapp-490af534-1491-452e-8ed6-a5eb54447dac" />
</Entity>
```

Index