

vCloud API Programming Guide

vCloud Director 5.1

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-000750-00

vmware[®]

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

Copyright © 2009–2012 VMware, Inc. All rights reserved. This product is protected by U.S. and international copyright and intellectual property laws. VMware products are covered by one or more patents listed at <http://www.vmware.com/go/patents>.

VMware is a registered trademark or trademark of VMware, Inc. in the United States and/or other jurisdictions. All other marks and names mentioned herein may be trademarks of their respective companies.

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

Contents

vCloud API Programming Guide	7
1 About the VMware vCloud API	9
Object Taxonomy	10
Objects, References, and Representations	12
Links and Link Relations	12
Client Workflow Overview	16
Using the vCloud API with vCloud Director	21
About the vCloud API Examples	22
2 Hello vCloud: A Simplified RESTful Workflow	23
Logging In	24
Find a Catalog and a vDC	26
Retrieve the Contents of a Catalog	27
Retrieve a Catalog Item	28
Retrieve Deployment Information From the vDC	30
Deploy the vApp	31
Get Information About a vApp	34
Displaying the Virtual Machine Console	37
Delete the vApp	38
Log Out	40
3 Exploring a Cloud	41
Summary of vCloud API Browsing Requests	41
Retrieve the Login URL and List of Supported API Versions	43
Create a Login Session Using the Integrated Identity Provider	44
Retrieve a List of Organizations Accessible to You	48
Retrieve an Administrative View of a Cloud	49
Retrieve a List of vSphere Platform Operations and Objects for a Cloud	52
4 Provisioning an Organization	55
Summary of vCloud API Provisioning Requests	56
Upload an OVF Package to Create a vApp Template	57
Download a vApp Template as OVF	66
Upload a Media Image	69
Copying and Moving with the vCloud API	71
Capturing and Importing vApps	72
Cataloging vApp Templates and Media Images	72
Creating and Using Independent Disks	75
View or Change the Owner of an Object	79

- 5 Deploying and Operating vApps 81**
 - Summary of vCloud API vApp and Virtual Machine Operations Requests 83
 - Create a vApp From a Template 85
 - Compose a vApp From Existing Virtual Machines 87
 - Recompose a vApp to Add or Remove Virtual Machines 90
 - Provide User Input Requested by a Virtual Machine 92
 - Attach or Detach an Independent Disk 93
 - Creating and Using vApp Snapshots 95
 - Operate a vApp 96
 - Configuring vApps and Virtual Machines 97

- 6 Creating and Managing Organizations 129**
 - Summary of Administrative Requests 129
 - Administrator Credentials and Privileges 132
 - Organization Administration 132
 - vDC Administration 139
 - Network Administration 147
 - Catalog Administration 176
 - User and Group Administration 179
 - Working With Roles and Rights 187
 - Controlling Access to vApps and Catalogs 191

- 7 Managing and Monitoring a Cloud 193**
 - Summary of vSphere Platform Extension Requests 193
 - Retrieve or Update System Settings 197
 - Attach a vCenter Server 198
 - Finding Available vCenter Resources 200
 - Create a Provider vDC 209
 - Create an External Network 219
 - Create a Network Pool 222
 - Import a Virtual Machine from vCenter 228
 - Relocate a Virtual Machine to a Different Datastore 231
 - Truststore and Keytab Maintenance 233
 - Retrieve the vSphere URL of an Object 236

- 8 Working With Object Metadata 239**
 - Retrieve or Update a Metadata Element 241
 - Retrieve or Update a Metadata Value 244

- 9 Using the Query Service 247**
 - Typed Queries 248
 - Packaged Queries 252
 - Query Parameters 257
 - Add a Metadata Filter to a Query 260

- 10 Configuring and Using Blocking Tasks and Notifications 263**
 - Configure Notifications and AMQP Settings 264
 - Retrieve or Update Blocking Task Settings 273

Monitor Blocking Tasks	276
Take Action on a Blocking Task	277
Extend The Timeout Expiration of an Active Task	279
11 vCloud Director Extension Services	281
Summary of vCloud API Extensibility Requests	282
Register an Extension Service	284
Adding or Removing Service Links	288
Service-Specific Tasks and Events	291
Authorization Framework for Extension Service Operations	294
Localization Framework for Extension Services	302
REST APIs for Extension Services	302
12 XML Representations in the vCloud API	307
XML Namespace Identifiers	309
Common vCloud API Attributes	310
Retrieve an Object as an Entity	312
Index	315

vCloud API Programming Guide

This edition of the *vCloud API Programming Guide* provides information about version 5.1 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
10SEP12	API Version 5.1
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

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.

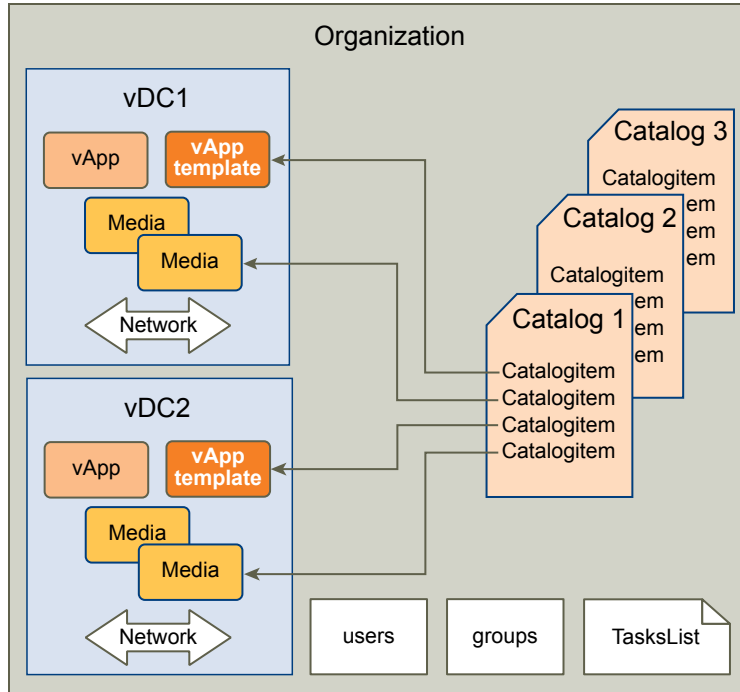
This chapter includes the following topics:

- [“Object Taxonomy,”](#) on page 10
- [“Objects, References, and Representations,”](#) on page 12
- [“Links and Link Relations,”](#) on page 12
- [“Client Workflow Overview,”](#) on page 16
- [“Using the vCloud API with vCloud Director,”](#) on page 21
- [“About the vCloud API Examples,”](#) on page 22

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. User credentials are authenticated by the organization's identity provider, which can be either the integrated identity provider included in vCloud Director or an external SAML-based identity provider.

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 or SAML-based identity provider. Groups must be imported. 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.

Organization vDCs

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

Organization vDC Networks

An organization vDC can be provisioned with one or more networks. These organization vDC networks can be configured to provide direct or routed connections to external networks, or can be isolated from external networks and other organization vDC networks. Routed connections require an Edge Gateway and network pool in the vDC. The Edge Gateway provides firewall, network address translation, static routing, VPN, and load balancing services.

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, including:

- 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 <code>id</code> attribute uniquely identifies the object, persists for the life of the object, and is never reused. The <code>id</code> attribute value is intended to provide a context-free identifier that can be used with the vCloud API <code>entityResolver</code> (see “Retrieve an Object as an Entity,” on page 312).
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 <code>id</code> attribute value, it uniquely identifies the object, persists for the life of the object, and is never reused. The value of the <code>href</code> 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 <code>href</code> as an opaque string. The rules that govern how the server constructs <code>href</code> 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:vcld: 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
abort	Abort this blocking task.	POST
add	Add an item to this container.	POST
alternate	References an alternate representation of this object.	GET
answer	Provide user input requested by a virtual machine.	POST
authorization:check	Check whether an extension service operation is authorized for an entity.	POST
blockingTask	A list of pending blocking task requests in this cloud.	GET
bundle:upload	Upload an extension service localization bundle.	PUT
bundles:cleanup	Remove unused extension service localization bundles.	POST
catalogItem	References the CatalogItem object that refers to this object.	GET
certificate:reset	Removes the SSL certificate used by this service.	POST
certificate:update	Updates the SSL certificate used by this service.	POST
checkCompliance	Check that this virtual machine is using a storage profile of the intended type.	POST
consolidate	Consolidate this virtual machine.	POST
controlAccess	Apply access controls to this object.	POST
copy	Reserved	N/A
deploy	Deploy this vApp.	POST
disable	Disable this object.	POST
discardState	Discard the suspended state of this virtual machine.	POST
disk:attach	Attach an independent disk to this virtual machine.	POST
disk:detach	Detach an independent disk from this virtual machine.	POST
down	References an object contained by this object.	GET

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

rel Attribute Value	Action or Relationship Description	Implied HTTP Verb
down:aclRules	Retrieve the ACL rules for this resource class action.	GET
down:apidefinitions	Retrieve the API definitions for this extension service.	GET
down:apiDefinitions	Retrieve the API definitions for this extension service.	GET
down:apiFilters	Retrieve the API filters for this extension service.	GET
down:extensibility	Add an extension service to the system.	POST
down:fileDescriptors	Retrieve file descriptors for extension services APIs	GET
down:files	Retrieve files for extension services APIs	GET
down:resourceClassActions	Retrieve the actions defined for this extension service resource class.	GET
down:resourceClasses	Retrieve the resource classes defined by this extension service.	GET
down:service		
down:serviceLinks	Retrieve the service links defined by this extension service.	GET
down:serviceResources	Retrieve the list of extension service resources of this class.	
down:services	Retrieve the list of registered extension services.	GET
download:alternate	Reserved	N/A
download:default	References the default location from which this file can be downloaded.	GET
download:identity	References the extended OVF descriptor of this vApp template. The extended OVF descriptor contains additional information such as MAC address, BIOS UUID, and NetworkConfigSection	GET
edgeGateway:configureServices	Update the network services offered by this Edge Gateway.	PUT
edgeGateway:reapplyServices	Reapply (after an update) the network services offered by this Edge Gateway.	POST
edgeGateway:redeploy	Redeploy the vShield Edge supporting this Edge Gateway.	POST
edgeGateway:syncSyslogSettings	Synchronize syslog server addresses used by this Edge Gateway with system defaults.	POST
edgeGateway:upgrade	Upgrade the backing configuration of this Edge Gateway from compact to full.	POST
edgeGateways	List the Edge Gateway objects in this organization vDC.	GET
edit	Modify this object, typically by replacing its current representation with the one in the request body.	PUT
enable	Enable this object.	POST
enterMaintenanceMode	Put this virtual machine into maintenance mode.	POST
entity	Retrieve a representation of the object on which an operation triggered this notification.	GET
entityResolver	Retrieve an object id as a context-free Entity element.	GET
event:create	Create an event in an this organization's event stream.	POST
exitMaintenanceMode	Take this virtual machine out of maintenance mode.	POST
fail	Fail this blocking task.	POST
firstPage	Reference to the first page of a paginated response.	GET

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

rel Attribute Value	Action or Relationship Description	Implied HTTP Verb
installVmwareTools	Install VMware Tools on this virtual machine.	POST
keystore:reset	Removes the keystore used by this service.	POST
keystore:update	Updates the keystore used by this service.	POST
keytab:reset	Removes the keytab used by this service.	POST
keytab:update	Updates the keytab used by this service.	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
merge	Merge one or more Provider vDCs with this Provider vDC.	POST
migrateVms	Migrate virtual machines from this resource pool to a different one.	POST
move	Reserved	N/A
nextPage	Reference to the next page of a paginated response.	GET
orgVdcNetworks	List the organization vDC networks supported by this Edge Gateway.	GET
ova	Reserved	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
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 to add, remove, or reconfigure virtual machines.	POST
reconfigureVm	Update multiple sections of a virtual machine.	POST
reconnect	Reconnect this vCenter Server to the system.	POST
refreshStorageProfiles	Refresh the list of storage profiles that exist on the vCenter service backing this Provider vDC.	POST
refreshVirtualCenter	Refresh the representation of this vCenter server	POST
register	Register a VCenter Server with the system.	POST
relocate	Relocate this virtual machine.	POST
remove	Remove this object.	DELETE
remove:force	Force removal of this object.	DELETE
repair	Repair this host or network.	POST
resourcePoolVmList	List the virtual machines using this resource pool.	GET
resume	Resume this blocking task.	POST
rights	List the service-specific rights created by this extension service.	GET

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

rel Attribute Value	Action or Relationship Description	Implied HTTP Verb
rights:cleanup	Remove service-specific rights no longer used by any extension service.	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
shadowVms	List shadow virtual machines associated with the virtual machines in this vApp template.	GET
snapshot:create	Create a snapshot of the virtual machines in this vApp.	POST
snapshot:removeAll	Remove all snapshots created for the virtual machines in this vApp.	POST
snapshot:revertToCurrent	Revert all virtual machines in this vApp to their current snapshot.	POST
storageProfile	References the storage profile for this object.	GET
syncSyslogSettings	Synchronize syslog server addresses used by this vApp network with system defaults.	POST
task	Retrieve the blocking task that triggered this notification.	GET
task:cancel	Cancel this task.	POST
task:create	Create a task object.	POST
task:owner	Reference to the owner of a task.	GET
truststore:reset	Remove the truststore used by this service.	POST
truststore:update	Update the truststore used by this service.	PUT
undeploy	Undeploy this vApp.	POST
unlock	Unlock this user account.	POST
unregister	Unregister this vCenter Server.	POST
up	References an object that contains this object.	GET
update:resourcePools	Update the resource pools of this Provider vDC	POST
updateProgress	Request an update of this task's progress.	POST
upgrade	Upgrade this ESX/ESXi host.	POST
upload:alternate	Reserved	N/A
upload:default	References the default location to which this object can be uploaded.	PUT
vSphereWebClientUrl	A URL that you can use to view this object with the vSphere Web Client	GET

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, representations 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:

- The set of objects that the API supports, and what they represent; for example, what is a vDC and how does it relate to an organization or catalog?
- How 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 a client refers 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 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,”](#) on page 21.

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 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 `SupportedVersions` document, which lists each version of the vCloud API that the server supports. For each version, the response lists the names and MIME types of the complex types defined in the version's XML namespace, and the version 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 API version that the client supports. The following header indicates that the request is for vCloud API version 5.1:

```
Accept: application/*+xml;version=5.1
```

Valid values for the HTTP `Accept` header in this release are:

version=5.1 The request is from a vCloud API version 5.1 client.

version=1.5 The request is from a vCloud API version 1.5 client.

Requests that specify API version 5.1 cannot access objects that do not exist in the vCloud API version 5.1 namespace.

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. (Requests cannot be compressed.) 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 309.
- 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.

Response Content

Response content depends on the requested operation. The response to a GET request is typically the complete representation of an existing object. The response to a PUT or POST request always contains values for the href, name, and id attributes of the object being created or updated. It also contains at most one Task element that you can retrieve to track the progress of the operation. When the Task completes with a status of success, a GET request to the object's href returns all properties of the object. If the Task completion status is not success, the object is in an indeterminate state, and should be deleted.

HTTP Response Codes

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.
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.
406 Not Acceptable	The resource identified by the request is only capable of generating response entities which have content characteristics not acceptable according to the accept headers sent in the request.
409 Conflict	The object state is not compatible with the requested operation.
415 Unsupported Media Type	The server is refusing to service the request because the entity of the request is in a format not supported by the requested resource for the requested method.
500 Internal Server Error	The request was received but could not be completed because of an internal error at the server.
504 Gateway Timeout	The server, while acting as a gateway or proxy, did not receive a timely response from the upstream server specified by the URI.

Using the vCloud API with vCloud Director

VMware vCloud Director 5.1 supports version 5.1 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 example XML representations. See [“About the Schema Reference,”](#) on page 21. For information about HTTP client programs to use with vCloud Director, see [“REST Client Programs,”](#) on page 21.

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 (Optional) If you want to use the vSphere Web Client to access vCloud API objects on a vSphere server, verify that the vSphere Web Client URL is enabled for all vCenter servers from which you want to retrieve the vSphere URL of an object.

You can manage this feature on the **General** tab of the vSphere Properties page of the vCloud Director Web console.

What to do next

Decide on an HTTP client program to use. See [“REST Client Programs,”](#) on page 21.

REST Client Programs

Any client application that can send HTTPS requests can be an appropriate tool for developing RESTful applications with the vCloud API.

REST client plug-ins are available for most browsers and many IDEs. 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

The *vCloud API Schema Reference* includes reference material for all elements, types, queries, and operations in the vCloud API. It also includes the schema definition files.

The schema reference is available in HTML format in the vCloud Director documentation center.

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 19. 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 5.1 of the vCloud API, is omitted from most examples.

```
Accept: application/*+xml;version=5.1
```

- 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 HTTPS, 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 referenced by 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 an organization vDC to deploy it in and an organization vDC network to connect it to.

6 [Deploy the vApp](#) on page 31

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.

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, undeploy it and power it off, then use an HTTP DELETE request to delete the vApp object.
- 10 [Log Out](#) on page 40
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 43. 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.

NOTE This procedure assumes that you are logging in with credentials managed by the vCloud Director integrated identity provider. Users whose credentials are managed by a SAML identity provider must follow a different login workflow.

Prerequisites

Verify that the following conditions are met:

- You are a member of an organization in the cloud, and have permission to create and operate vApps.
- Your organization contains at least one vDC and one network. For more information about setting up an organization to support the Hello vCloud workflow, see [Chapter 6, “Creating and Managing Organizations,”](#) on page 129.
- Your 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,”](#) on page 55.

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 Using the Integrated](#)

[Identity Provider](#),” on page 45. 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 Using the Integrated Identity Provider”](#) 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 132. Because the `System` organization does not support some of the features demonstrated in the Hello vCloud workflow, you should try this example in another organization.

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=5.1
```

Response:

```
200 OK
x-vcloud-authorization: cn9uYmdugN8E2j96+5Lqrc3YBvFsEgDHXzyfJrJ/6bM=
Content-Type: application/vnd.vmware.vcloud.session+xml;version=5.1
...
<Session
  xmlns="http://www.vmware.com/vcloud/v2.0"
  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 and vDCs.

Prerequisites

Verify that you are logged in to the vCloud API 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 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 ... />
  <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.
- An organization vDC named `ExampleVdc01`, at URL `https://vcloud.example.com/api/vdc/5`, where you can deploy the vApp.

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 referenced by 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 to the vCloud API 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: Retrieve the Contents of a Catalog,”](#) on page 28.

Example: Retrieve 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 to the vCloud API 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: Retrieve 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 an organization vDC to deploy it in and an organization vDC network to connect it to.

Instantiation, deployment, and operation of a vApp all take place in the context of an organization 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 in the organization vDC that the vApp can connect to.

“[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 to the vCloud API 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="Isolated" />
  <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 action URL for `instantiateVAppTemplate`. The `rel` attribute of this link has a value of `add`. It implements an action that adds an object (a vApp) to the vDC.
- A list of `AvailableNetworks` that includes all the networks in the vDC.

Deploy the vApp

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.

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.

Prerequisites

Verify that you are logged in to the vCloud API 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 86. 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 a template, obtained from the href attribute of the Entity contained by the CatalogItem that you retrieved in [“Retrieve a Catalog Item,”](#) on page 28 and supplied 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 vDC 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 vDC network.

For more information about creating networks with the vCloud API, see [“About vCloud Director Networks,”](#) on page 148.

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 to the vCloud API 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 311.
- 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. Actions such as acquiring a screen ticket or thumbnail, and inserting or removing media, are meaningful only in the context of a virtual machine. Other actions, like `shutdown` and `reboot`, can be applied to either object. See [Chapter 5, “Deploying and Operating vApps,”](#) on page 81.

Request:

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

Response:

```

200 OK
Content-Type: application/vnd.vmware.vcloud.vApp+xml
...
<VApp
  ...
  deployed="true"
  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 ... />
  ...
  <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>
    <IpScopes>
      ...
    </IpScopes>
    <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 ... />
...
<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.

Prerequisites

- Verify that you are logged in to the vCloud API 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, undeploy it and power it off, then use an HTTP DELETE request to delete the vApp object.

A deployed vApp has a link that you can use with a POST request to undeploy it and then take a power action such as powering it off or suspending it. 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 to the vCloud API as a system administrator or the object owner.

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 Undeploy the vApp, and specify that it should also be powered off.

Make a POST request to the vApp’s `action/undeploy` link, which has the following form:

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

In the request body, specify that the undeployment include powering off the vApp. See [“Example: Undeploy, Power Off, and Delete a vApp,”](#) on page 39.

- 3 Retrieve the XML representation of the vApp again.

Now that it has been powered off and undeployed, the vApp 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: Undeploy, Power Off, and 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: Undeploy, Power Off, and Delete a vApp

The undeploy request body, an `UndeployVAppParams` element, allows you to specify an `UndeployPowerAction`. This example specifies an `UndeployPowerAction` of `powerOff`. Although `powerOff` is the default `UndeployPowerAction`, we show it here for clarity.

Request:

```
POST https://vcloud.example.com/api/vApp/vapp-7/action/undeploy
Content-Type: application/vnd.vmware.vcloud.undeployVAppParams+xml
...
<?xml version="1.0" encoding="UTF-8"?>
<UndeployVAppParams
  xmlns="http://www.vmware.com/vcloud/v1.5">
  <UndeployPowerAction>powerOff</UndeployPowerAction>
</UndeployVAppParams>
```

Response:

```
202 Accepted
...
<Task
  xmlns="http://www.vmware.com/vcloud/v1.5"
  ...
  operation="Undeploying Virtual Application Linux FTP server (7)"
  ... >
</Task>
```

After the vApp has been undeployed and powered off, its representation includes a link where `rel="remove"`. Make a DELETE request to this link to remove the vApp.

Request:

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

Response:

```
202 Accepted
...
<Task
  ...
  operation="Deleting Virtual Application Linux FTP server (7)"
  expiryTime="2010-09-23T08:00:55.402-07:00"
  ... >
</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 40.

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

You can use HTTP GET requests 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, because 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. The contents of this list is based on your role and privileges.

This chapter includes the following topics:

- [“Summary of vCloud API Browsing Requests,”](#) on page 41
- [“Retrieve the Login URL and List of Supported API Versions,”](#) on page 43
- [“Create a Login Session Using the Integrated Identity Provider,”](#) on page 44
- [“Retrieve a List of Organizations Accessible to You,”](#) on page 48
- [“Retrieve an Administrative View of a Cloud,”](#) on page 49
- [“Retrieve a List of vSphere Platform Operations and Objects for a Cloud,”](#) on page 52

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.

IMPORTANT Request URLs are always available in `Link` elements contained by the representation of the object on which they operate. URL forms shown here are for reference purposes only. Although URLs have a well-known syntax and a well-understood interpretation, a client should treat vCloud API request URLs as opaque strings. The rules that govern how the server constructs these strings might change in future releases.

This summary may not cover all requests in this category. For the complete list of requests, along with detailed information about input and output types, see the Operations lists in the schema reference.

Table 3-1. Summary of vCloud API Browsing Requests

Operation	Request	Request Body	Response
Show login URL and list of supported API versions	GET <i>API-URL/versions</i>	None	SupportedVersions
Log in	POST <i>API-URL/sessions</i>	None	Session
Log out	DELETE <i>API-URL/session</i>	None	200 OK
Log in [DEPRECATED]	POST <i>API-URL/login</i>	None	OrgList
Log out [DEPRECATED]	POST <i>API-URL/login</i>	None	200 OK
Retrieve a list of entry points for browsing operations	GET <i>API-URL/session</i>	None	200 OK
Retrieve a list of organizations to which you have access	GET <i>API-URL/org/</i>	None	OrgList
Retrieve the contents of an organization	GET <i>API-URL/org/id</i>	None	Org
Retrieve properties of a network	GET <i>API-URL/network/id</i>	None	OrgNetwork
Retrieve the contents of a catalog	GET <i>API-URL/catalog/id</i>	None	Catalog
Retrieve properties of a catalog item	GET <i>API-URL/catalogItem/id</i>	None	CatalogItem
Retrieve the contents of a vDC	GET <i>API-URL/vdc/id</i>	None	Vdc
Retrieve properties of a media image	GET <i>API-URL/media/id</i>	None	Media
Retrieve a vApp template	GET <i>API-URL/vAppTemplate/vappTemplate-id</i>	None	VAppTemplate
Retrieve properties of a vApp	GET <i>API-URL/vApp/vapp-id</i>	None	VApp
Retrieve properties of a virtual machine	GET <i>API-URL/vApp/vm-id</i>	None	Vm
Retrieve metadata for an object such as an organization, network, vApp, virtual machine, media image, or independent disk.	GET <i>API-URL/object/id/metadata</i>	None	Metadata
Retrieve a list of IP addresses allocated to a network. [NEW]	GET <i>API-URL/network/id/allocatedAddresses</i>	None	AllocatedIpAddresses

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>1.5</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/v1.5/schema/master.xsd</SchemaLocation>
```

```
    </MediaTypeMapping>
```

```
    <MediaTypeMapping>
```

```
      ...
```

```
    </MediaTypeMapping>
```

```
  </VersionInfo>
```

```
  <VersionInfo>
```

```
    <Version>5.1</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 retrieve 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 Using the Integrated Identity Provider

The vCloud API login mechanism authenticates a user and creates a `Session` object that contains the URLs from which that user can begin browsing. Users who authenticate to the integrated identity provider use basic HTTP authentication.

Prerequisites

NOTE This procedure assumes that you are logging in with credentials managed by the vCloud Director integrated identity provider. Users whose credentials are managed by a SAML identity provider must follow a different login workflow.

- Verify that you know the login URL. See [“Retrieve the Login URL and List of Supported API Versions,”](#) on page 43.
- Verify that you are logging in as a user whose identity is managed by the vCloud Director integrated identity provider.

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 Using the Integrated Identity Provider,”](#) on page 45.

- 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 197.

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 Using the Integrated Identity Provider

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 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
...
<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 48.
vcloud	A link to administrative objects and operations. See Chapter 6, “Creating and Managing Organizations,” on page 129
vmwExtension	A link to the vCloud API extensions, accessible to a system administrator. See Chapter 7, “Managing and Monitoring a Cloud,” on page 193.
queryList	A link to the set of typed queries you can run. See Chapter 9, “Using the Query Service,” on page 247.
entity	A link to the entity resolver. See “Retrieve an Object as an Entity,” on page 312.

Create a Login Session Using a SAML Identity Provider

The vCloud API login mechanism authenticates a user and creates a `Session` object that contains the URLs from which that user can begin browsing. Users who authenticate to a SAML identity provider must acquire and process a security assertion from that identity provider, then submit the processed assertion to the vCloud API login URL.

Prerequisites

NOTE This procedure assumes that you are logging in with credentials managed by a SAML identity provider. Users whose credentials are managed by the vCloud Director integrated identity provider must follow a different login workflow.

- Verify that you know the login URL. See [“Retrieve the Login URL and List of Supported API Versions,”](#) on page 43
- Verify that you are logging in as a user whose identity is managed by the SAML identity provider defined by your organization.

Procedure

- 1 Acquire the SAML assertion from your identity provider.
The system administrator must use the vSphere SSO Service as the identity provider.
- 2 Compress the assertion using GZIP.
- 3 Encode the compressed assertion a MIME Base64 encoding, as specified in RFC 1421.
- 4 Use the login URL to authenticate to the cloud.

POST a request to this URL. The request must include an `Authorization` header that specifies `SIGN` as the authorization method and has the following attributes:

Table 3-2. Authorization Header Attributes and Values

Attribute Name	Attribute Value
token	The compressed, encoded identity assertion from your SAML identity provider.
org	The name of your vCloud Director organization.

See [“Example: Create a Login Session Using a SAML Identity Provider,”](#) on page 47.

5 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 197.

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 Using a SAML Identity Provider

This example shows a login request and response for a user of a SAML identity provider logging in to the Finance organization of a cloud whose login URL is `https://vcloud.example.com/api/sessions`. The user name is included in the encoded credentials.

Request:

```
POST https://vcloud.example.com/api/sessions
Authorization: SIGN token="compressed-encoded-credentials",org="Finance"
Accept: application/*+xml;version=5.1
```

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="bob"
  org="Finance"
  ... >
<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 48.
vcloud	A link to administrative objects and operations. See Chapter 6, “Creating and Managing Organizations,” on page 129
vmwExtension	A link to the vCloud API extensions, accessible to a system administrator. See Chapter 7, “Managing and Monitoring a Cloud,” on page 193.
queryList	A link to the set of typed queries you can run. See Chapter 9, “Using the Query Service,” on page 247.
entity	A link to the entity resolver. See “Retrieve an Object as an Entity,” on page 312.

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 Using the Integrated Identity Provider,”](#) 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 49.

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 Using the Integrated Identity Provider,”](#) 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 50

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" />
```

```
<Link ... />
```

```
...
```

```
<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 Using the Integrated Identity Provider,”](#) 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 52.

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

Request:

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

The response is a `VMWExtension` element containing a number of `Link` elements. This example shows only a subset of `VMWExtension` contents.

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

```


Provisioning an Organization

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

The vCloud API allows you to upload and download OVF packages, and upload media images. Operations are characterized as uploads when they transfer 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 an upload, and a GET request initiates a download. 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 to make them available to other users.

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 creates a copy of 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. You can add cloned templates and media images to catalogs.
Capturing	The vCloud API capture operation creates a vApp template from a vApp. You can add the captured template to a catalog or download it as an OVF package.
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 or download it as an OVF package.

You can also create independent disks that are contained by an organization vDC and can be connected to any virtual machine in that vDC.

This chapter includes the following topics:

- [“Summary of vCloud API Provisioning Requests,”](#) on page 56
- [“Upload an OVF Package to Create a vApp Template,”](#) on page 57
- [“Download a vApp Template as OVF,”](#) on page 66
- [“Upload a Media Image,”](#) on page 69
- [“Copying and Moving with the vCloud API,”](#) on page 71
- [“Capturing and Importing vApps,”](#) on page 72
- [“Cataloging vApp Templates and Media Images,”](#) on page 72
- [“Creating and Using Independent Disks,”](#) on page 75
- [“View or Change the Owner of an Object,”](#) on page 79

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.

IMPORTANT Request URLs are always available in Link elements contained by the representation of the object on which they operate. URL forms shown here are for reference purposes only. Although URLs have a well-known syntax and a well-understood interpretation, a client should treat vCloud API request URLs as opaque strings. The rules that govern how the server constructs these strings might change in future releases.

This summary may not cover all requests in this category. For the complete list of requests, along with detailed information about input and output types, see the Operations lists in the schema reference.

Table 4-1. Summary of Provisioning Requests

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

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

Operation	Request	Request Body	Response
Remove an item from a catalog.	DELETE <i>API-URL/catalog/id/catalogItem/id</i>	None	204 No content
Control access to catalogs.	POST <i>API-URL/catalog/id/action/controlAccess</i>	ControlAccessParams	ControlAccessParams
Retrieve the owner of a media object.	GET <i>API-URL/media/id/owner</i>	None	Owner
Retrieve the owner of a vApp template.	GET <i>API-URL/vAppTemplate/vappTemplate-id/owner</i>	None	Owner
Retrieve the owner of a vApp.	GET <i>API-URL/vApp/id/owner</i>	None	Owner
Update the owner of a vApp.	PUT <i>API-URL/vApp/id/owner</i>	Owner	204 No Content
Create an independent disk in a vDC. [NEW]	POST <i>API-URL/vdc/id/disk</i>	DiskCreateParams	Disk
Retrieve properties of an independent disk. [NEW]	GET <i>API-URL/disk/id</i>	None	Disk
Update an independent disk in a vDC. [NEW]	POST <i>API-URL/disk/id</i>	Disk	Disk
Retrieve a list of all virtual machines attached to an independent disk. [NEW]	GET <i>API-URL/disk/id/attachedVms</i>	None	Vms
Delete an independent disk. [NEW]	DELETE <i>API-URL/disk/id</i>	None	Task

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 81.

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 311.

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 `NewVirtualMac001` and `NewVirtualMac002`.

1 [Initiating the OVF Upload](#) on page 59

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 61

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 62

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 63

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 59.

- 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 59.

- 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 59.

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 59.

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 12.

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 59.

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 62.
- 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 59.

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 59, 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 <code>size</code> attribute of the <code>File</code> element in the OVF descriptor.
bytesTransferred	For all file references other than the descriptor, this attribute is initially set to a value of <code>0</code> , indicating that the upload has not begun. In the <code>File</code> element that references the OVF descriptor, the value of the <code>bytesTransferred</code> attribute is equal to the value of the <code>size</code> attribute, indicating that all the bytes in the descriptor were transferred.
name	The file name, taken from the <code>href</code> attribute of the <code>File</code> 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 197.

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 61.
- Retrieve the upload URLs for all files in the package. See [“Retrieving the Upload URLs,”](#) on page 62.

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 64.

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 64.
- 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 64.

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 65.

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 66
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 68
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 68
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 197.

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 to the vCloud API 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 66.

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 68.

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 67. 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 to the vCloud API 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 69.

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 68 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 57.

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 add link for media in the target vDC.

This link 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 70.

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 70.

- 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 63.

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 228.

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 74 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 contains information extracted from the request, and includes links that an administrator or catalog owner can use to manage the `CatalogItem`.

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`.

Creating and Using Independent Disks

Independent disks are stand-alone virtual disks that you create in organization vDCs. Administrators and users who have adequate rights can create, remove, and update independent disks, and connect them to virtual machines.

When you create an independent disk, it is associated with an organization vDC but not with a virtual machine. After the disk has been created in a vDC, the disk owner or an administrator can attach it to any virtual machine deployed in that vDC. The disk owner can also modify disk properties, detach it from a virtual machine, and remove it from the vDC. The system administrator and organization administrator of the organization that contains the vDC have the same rights to use and modify the disk as the disk owner.

Create or Update an Independent Disk

To create an independent disk in an organization vDC, POST a `DiskCreateParams` element to the vDC's disk link.

To create an independent disk, you must specify its name and size. You can optionally include a description and specify a storage profile to be used by the disk. After you have created the disk, you can modify its name, description, and storage profile.

The owner of a disk is initially the user who created it. To change the owner, see [“View or Change the Owner of an Object,”](#) on page 79.

Procedure

- 1 Choose an organization vDC to contain the disk.
- 2 Create a `DiskCreateParams` element.
You must specify the size (in Megabytes) and name of the independent disk. See the request portion of [“Example: Create an Independent Disk,”](#) on page 76.
- 3 POST the `DiskCreateParams` element you created in [Step 2](#) to the URL for adding disks to the organization vDC.

See the request portion of [“Example: Create an Independent Disk,”](#) on page 76.

Example: Create an Independent Disk

This example adds an independent disk to the organization vDC created in [“Add a vDC to an Organization,”](#) on page 140. Because optional attributes `busType` and `busSubType` are omitted, a SCSI disk is created.

Request:

```
POST https://vcloud.example.com/api/admin/vdc/44/disk
Content-Type: application/vnd.vmware.admin.diskCreateParams+xml
...
<?xml version="1.0" encoding="UTF-8"?>
<DiskCreateParams
  xmlns="http://www.vmware.com/vcloud/v1.5">
  <Disk
    name="500GB-SCSI"
    size="500000000">
    <Description>500 GB SCSI Disk</Description>
  </Disk>
</DiskCreateParams>
```

The response, a subset of which appears here, is a `Disk` element that contains an embedded `Task` that tracks creation of the disk. Because the request did not specify a storage profile for the disk, it uses the default storage profile for the containing organization vDC. The response also includes `Link` elements that enable access to disk operations and metadata. While the disk is under construction, its status remains 0.

Response:

```
200 OK
Content-Type: application/vnd.vmware.vcloud.disk+xml
...
<?xml version="1.0" encoding="UTF-8"?>
<Disk
  xmlns="http://www.vmware.com/vcloud/v1.5"
  size="500000000"
```

```

status="0"
name="500GB-SCSI"
id="urn:vcloud:disk:128"
type="application/vnd.vmware.vcloud.disk+xml"
href="https://vcloud.example.com/api/disk/128"
... >
<Link
  rel="up"
  type="application/vnd.vmware.vcloud.vdc+xml"
  href="https://vcloud.example.com/api/vdc/44" />
<Link
  rel="remove"
  href="https://vcloud.example.com/api/disk/128" />
<Link
  rel="edit"
  type="application/vnd.vmware.vcloud.disk+xml"
  href="https://vcloud.example.com/api/disk/128" />
<Link
  rel="down"
  type="application/vnd.vmware.vcloud.owner+xml"
  href="https://vcloud.example.com/api/disk/128/owner" />
<Link
  rel="down"
  type="application/vnd.vmware.vcloud.vms+xml"
  href="https://vcloud.example.com/api/disk/128/attachedVms" />
<Link
  rel="down"
  type="application/vnd.vmware.vcloud.metadata+xml"
  href="https://vcloud.example.com/api/disk/128/metadata" />
<Description>Independent Disk</Description>
<Tasks>
  <Task
    ...
    operationName="vdcCreateDisk"
    ... >
    ...
  </Task>
</Tasks>
<StorageProfile
  type="application/vnd.vmware.vcloud.vdcStorageProfile+xml"
  name="bronze"
  href="https://vcloud/example.com/api/vdcStorageProfile/128" />
<Owner
  type="application/vnd.vmware.vcloud.owner+xml">
  <User
    type="application/vnd.vmware.admin.user+xml"
    href="https://vcloud.example.com/api/admin/user/120" />
  </Owner>
</Disk>

```

Remove an Independent Disk

To remove an independent disk, verify that no powered-on virtual machines are attached to it, then use a DELETE request to delete it.

A `Disk` element includes a link of the following form, which you can GET to return a list of virtual machines to which the disk is attached.

```
<Link
  rel="down"
  type="application/vnd.vmware.vcloud.vms+xml"
  href="https://vcloud.example.com/api/disk/128/attachedVms" />
```

There are also two queries that you can use to return a list of virtual machines, the disks connected to them, and the vDC that contains them:

vmDiskRelation	Lists this information for <code>Vm</code> and <code>Disk</code> objects that you own.
AdminvmDiskRelation	Lists this information for all <code>Vm</code> and <code>Disk</code> objects in a cloud (system administrators only).

Prerequisites

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

Procedure

- 1 Verify that the independent disk is not connected to any virtual machines.
- 2 Delete the independent disk.
 - Make a DELETE request to the URL in the `rel="remove"` link in the `Disk`.

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

Example: Remove an Independent Disk

Request:

```
DELETE https://vcloud.example.com/api/disk/128
```

Response:

```
202 Accepted
...
<Task
  ...
  operation="Deleting Disk (128)"
  ... >
</Task>
```

View or Change the Owner of an Object

You can view the owner of a VApp, VAppTemplate, Disk, or Media object by making a GET request to the object's owner link. If you have adequate rights, you can change the owner of a Disk or VApp object, but not that of a VAppTemplate or Media object. An administrator can view or change the owner of any object.

The initial owner of a VApp, VAppTemplate, Catalog, Disk, 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. Object-specific rights to change ownership are included in several predefined roles. See [“Predefined Roles and Their Rights,”](#) on page 187.

Prerequisites

- To change the owner of a Disk or VApp object, you must be an organization administrator or system administrator, or a user whose role includes the right to change the owner of these objects.
- To change the owner of a Catalog object, 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 request 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, 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 79.

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

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.

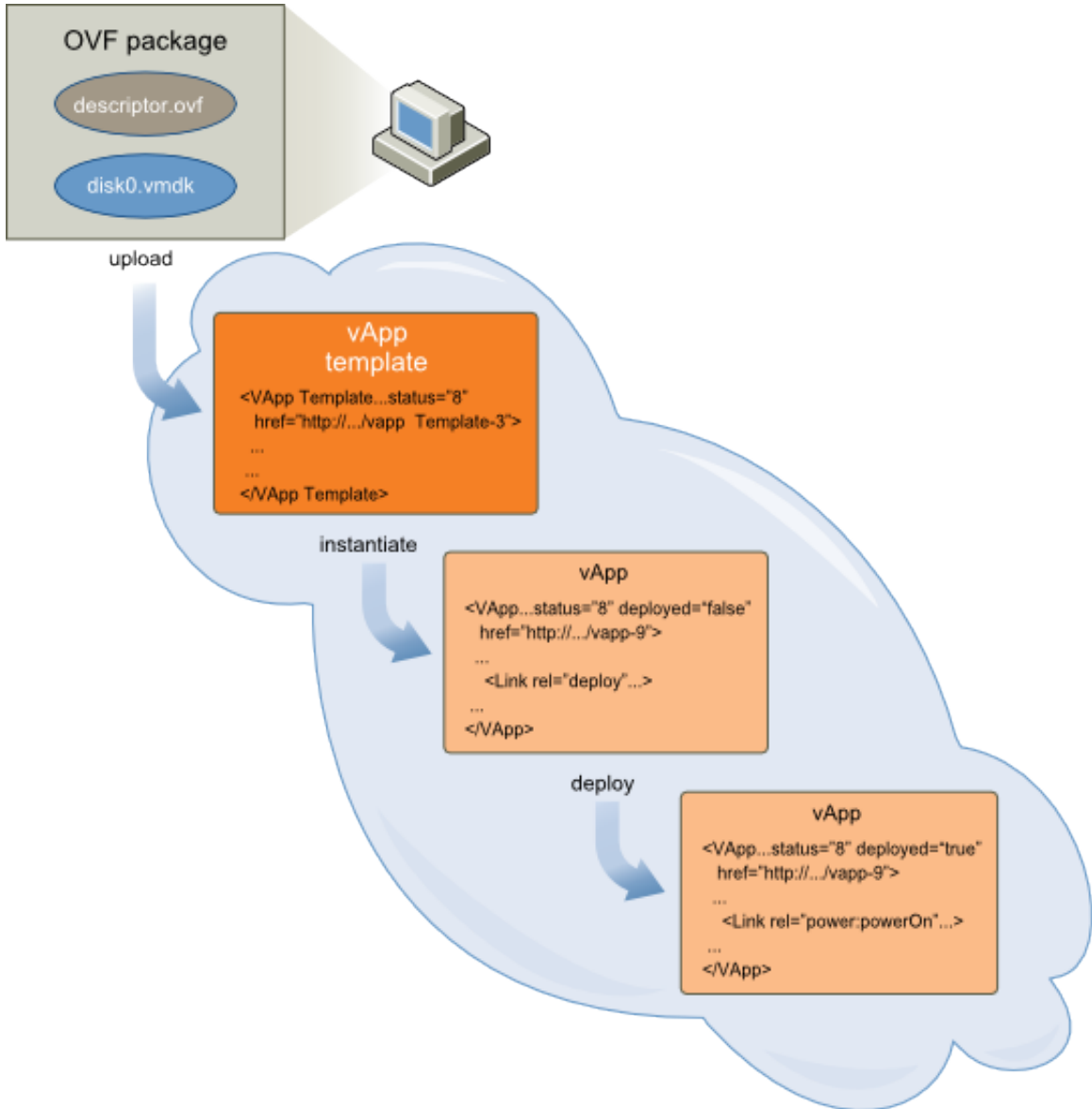
vApp Life Cycle

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



This chapter includes the following topics:

- [“Summary of vCloud API vApp and Virtual Machine Operations Requests,”](#) on page 83
- [“Create a vApp From a Template,”](#) on page 85
- [“Compose a vApp From Existing Virtual Machines,”](#) on page 87
- [“Recompose a vApp to Add or Remove Virtual Machines,”](#) on page 90
- [“Provide User Input Requested by a Virtual Machine,”](#) on page 92
- [“Attach or Detach an Independent Disk,”](#) on page 93
- [“Creating and Using vApp Snapshots,”](#) on page 95

- “Operate a vApp,” on page 96
- “Configuring vApps and Virtual Machines,” on page 97

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`

IMPORTANT Request URLs are always available in Link elements contained by the representation of the object on which they operate. URL forms shown here are for reference purposes only. Although URLs have a well-known syntax and a well-understood interpretation, a client should treat vCloud API request URLs as opaque strings. The rules that govern how the server constructs these strings might change in future releases.

This summary may not cover all requests in this category. For the complete list of requests, along with detailed information about input and output types, see the Operations lists in the schema reference.

Table 5-1. Summary of vApp and Virtual Machine Operations Requests

Operation	Request	Request Body	Response
Instantiate a vApp template to create a vApp	POST <i>API-URL/vdc/id/action/instantiateVAppTemplate</i>	InstantiateVAppTemplateParams	VApp
Change the name or description of a vApp.	PUT <i>API-URL/vApp/vapp-id</i>	VApp	Task
Compose a vApp	POST <i>API-URL/vdc/id/action/composeVApp</i>	ComposeVAppParams	VApp
Recompose a vApp to add or remove virtual machines	POST <i>API-URL/vApp/vapp-id/action/recomposeVApp</i>	RecomposeVAppParams	Task
Deploy a vApp or Virtual Machine	POST <i>VApp-or-Vm-URL/action/deploy</i>	DeployVAppParams	Task
Undeploy a vApp or Virtual Machine	POST <i>VApp-or-Vm-URL/action/undeploy</i>	UndeployVAppParams	Task
Power On a vApp or Virtual Machine	POST <i>VApp-or-Vm-URL/action/powerOn</i>	None	Task
Power Off a vApp or Virtual Machine	POST <i>VApp-or-Vm-URL/action/powerOff</i>	None	Task
Reset a vApp or Virtual Machine	POST <i>VApp-or-Vm-URL/action/reset</i>	None	Task
Suspend a vApp or Virtual Machine	POST <i>VApp-or-Vm-URL/action/suspend</i>	None	Task
Discard the Suspended State of a vApp or Virtual Machine	POST <i>VApp-or-Vm-URL/action/discardSuspendedState</i>	None	Task
Shut Down a vApp or Virtual Machine	POST <i>VApp-or-Vm-URL/action/shutdown</i>	None	Task

Table 5-1. Summary of vApp and Virtual Machine Operations Requests (Continued)

Operation	Request	Request Body	Response
Reboot a vApp or Virtual Machine	POST <i>VApp-or-Vm-URL/action/reboot</i>	None	Task
Retrieve product sections of a vApp or virtual machine	GET <i>VApp-or-Vm-URL/productSections</i>	None	ProductSectionList
Update product sections of a vApp or virtual machine	PUT <i>VApp-or-Vm-URL/productSections</i>	ProductSectionList	Task
Retrieve product sections of a vApp template	GET <i>API-URL/vAppTemplate/vappTemplate-id/productSections</i>	None	ProductSectionList
Retrieve version of VMware Tools installed on a virtual machine	GET <i>Vm-URL/runtimeInfoSection</i>	None	RuntimeInfoSection
Install VMware Tools on a virtual machine	POST <i>Vm-URL/action/installVMwareTools</i>	None	Task
Consolidate a virtual machine	POST <i>Vm-URL/action/consolidate</i>	None	Task
Upgrade the hardware version of a virtual machine	POST <i>Vm-URL/action/upgradeHardwareVersion</i>	None	Task
Insert Media Into a Virtual Machine	POST <i>Vm-URL/action/insertMedia</i>	MediaInsertOrEjectParams	Task
Eject Media from a Virtual Machine	POST <i>Vm-URL/action/ejectMedia</i>	MediaInsertOrEjectParams	Task
List Media Devices of a Virtual Machine	GET <i>Vm-URL/virtualHardwareSection/media</i>	None	RasdItemsList
Get a Request for User Input	GET <i>Vm-URL/question</i>	None	VmPendingQuestion
Provide Requested User Input	POST <i>Vm-URL/question/action/answer</i>	VmQuestionAnswer	204 No Content
Get a Screen Thumbnail for a Virtual Machine	GET <i>Vm-URL/screen</i>	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 <i>Vm-URL/screen/action/acquireTicket</i>	None	ScreenTicket
Attach an independent disk to a virtual machine. [NEW]	POST <i>Vm-URL/disk/action/attach</i>	DiskAttachOrDetachParams	Task
Detach an independent disk from a virtual machine. [NEW]	POST <i>Vm-URL/disk/action/detach</i>	DiskAttachOrDetachParams	Task
Create snapshots of all virtual machines in a vApp. [NEW]	POST <i>API-URL/vApp/vapp-id/action/createSnapshot</i>	CreateSnapshotParams	Task
Remove snapshots of all virtual machines in a vApp. [NEW]	POST <i>API-URL/vApp/vapp-id/action/removeAllSnapshots</i>	None	Task

Table 5-1. Summary of vApp and Virtual Machine Operations Requests (Continued)

Operation	Request	Request Body	Response
Revert all virtual machines in a vApp to their current snapshots. [NEW]	POST <i>API-URL/vApp/vapp-id/action/revertToCurrentSnapshot</i>	None	Task
Retrieve information about vApp snapshots. [NEW]	GET <i>API-URL/vApp/vapp-id/snapshotSection</i>	None	SnapshotSection
Validate the storage profile compliance of a virtual machine. [NEW]	POST <i>API-URL/vApp/vm-id/action/checkCompliance</i>	None	Task
Retrieve a virtual machine storage profile validation report. [NEW]	GET <i>API-URL/vApp/vm-id/complianceResult</i>	None	ComplianceResult

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 to the vCloud API 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 86 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 98.

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 98. 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 98.
 - 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 to the vCloud API 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 88.

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 247.

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 to the vCloud API 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 90.

- 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 247.

Provide User Input Requested by a Virtual Machine

A request for a virtual machine to change state (power on, suspend, reconfigure, and so on) might cause the virtual machine to ask for additional user input before it can complete.

A vApp that contains a Vm awaiting a user response has status="5", and includes a link that you can GET to discover what input is needed.

Prerequisites

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

Procedure

- 1 Find the question link in the target vApp.

This link has the following form:

```
<Link
  rel="down"
  type="application/vnd.vmware.vcloud.vmPendingQuestion+xml"
  href="http://vcloud.example.com/api/vApp/vm-5/question" />
```

- 2 Make a GET request to the URL in that link's href value.

The response is a VmPendingQuestion response that includes the question and the set of possible answers.

- 3 Create a VmQuestionAnswer element that supplies the answer.

See ["Example: Provide User Input Requested by a Virtual Machine,"](#) on page 92.

- 4 POST the element to the question/action/answer link included in the VmPendingQuestion response.

Example: Provide User Input Requested by a Virtual Machine

In this series of examples, a virtual machine that was recently reconfigured in vCenter to add a new parallel port device and then powered on is requesting user input about where to send output from the device. The powerOn request cannot complete until this input is supplied.

The first step is to use the vmPendingQuestion link, shown in [Step 1](#), to get a VmPendingQuestion response that includes the question and the set of possible answers.

Request:

```
GET https://vcloud.example.com/api/vApp/vm-5/question
```

Response:

```
200 OK
Content-type: application/vnd.vmware.vcloud.vmPendingQuestion+xml
...
<VmPendingQuestion
  xmlns="http://www.vmware.com/vcloud/v1.5">
  <Link
    rel="answer"
    type="application/vnd.vmware.vcloud.vmPendingAnswer+xml"
    href="http://vcloud.example.com/api/vApp/vm-5/question/action/answer" />
  <Question>msg.parallel.file.open:Parallel port output file
    "/vmfs/volumes/d6162a46-58e50cab/linuxftp/vm-mgi.log" already
    exists. Do you want to replace it with any newly created content,
    or append new content to the end of the file?
  </Question>
  <QuestionId>50</QuestionId>
  <Choices>
    <Id>0</Id>
    <Text>Append</Text>
  </Choices>
  <Choices>
    <Id>1</Id>
    <Text>Replace</Text>
  </Choices>
  <Choices>
    <Id>2</Id>
    <Text>Cancel</Text>
  </Choices>
</VmPendingQuestion>
```

To supply the answer, POST a `VmQuestionAnswer` element to the `question/action/answer` link of the `Vm`.

Request:

```
POST http://vcloud.example.com/api/vApp/vm-5/question/action/answer
Content-type: application/vnd.vmware.vcloud.vmPendingAnswer+xml
<?xml version="1.0" encoding="UTF-8"?>
<VmQuestionAnswer
  xmlns="http://www.vmware.com/vcloud/v1.5">
  <ChoiceId>2</ChoiceId>
  <QuestionId>50</QuestionId>
</VmQuestionAnswer>
```

Attach or Detach an Independent Disk

Use the `disk/action/attach` or `disk/action/detach` links in a `Vm` to attach or detach an independent disk. You must be the owner of the `Vm` element and the disk.

Every `Vm` element includes links of the following form:

```
<Link
  rel="disk:attach"
  type="application/vnd.vmware.vcloud.diskAttachOrDetachParams+xml"
  href="https://vcloud.example.com/api/vApp/vm-4/disk/action/attach" />
```

```

<Link
  rel="disk:detach"
  type="application/vnd.vmware.vcloud.diskAttachOrDetachParams+xml"
  href="https://vcloud.example.com/api/vApp/vm-4/disk/action/detach" />

```

You can POST a `DiskAttachOrDetachParams` element to one of these URLs to attach or detach an independent disk.

Prerequisites

- Verify that you are logged in to the vCloud API as a system administrator or the object owner.
- Verify that the vDC contains an independent disk. If the vDC does not contain independent disks, you can create one. See [“Create or Update an Independent Disk,”](#) on page 76.

Procedure

- 1 Retrieve a reference to the independent disk to attach to a virtual machine.

The independent disk and the virtual machine must be contained by the same vDC. You can retrieve references to independent disks in a vDC in the following ways:

- Use the query service. A query like this one returns references to all `Disk` objects in the vDC named `MyVdc`.

```

GET https://vcloud.example.com/api/query?
type=disk&format=references&filter=vdcName==MyVdc

```

- Retrieve the vDC and look for `ResourceEntity` elements whose `type` attribute has a value of `application/vnd.vmware.vcloud.disk+xml`. The `href` value of a `ResourceEntity` that represents a disk is a reference to that `Disk` object.

- 2 Create a `DiskAttachOrDetachParams` element.

Provide a reference to the independent disk in the `href` attribute of the `Disk` element.

- 3 POST the `DiskAttachOrDetachParams` to the `disk/action/attach` link of the `Vm`.

Example: Attach an Independent Disk to a Virtual Machine

This request attaches the disk created in [“Create or Update an Independent Disk,”](#) on page 76 to a virtual machine.

Request:

```

POST https://vcloud.example.com/api/vApp/vm-4/disk/action/attach
Content-Type: application/vnd.vmware.vcloud.diskAttachOrDetachParams+xml
...
<?xml version="1.0" encoding="UTF-8"?>
<DiskAttachOrDetachParams
  xmlns="http://www.vmware.com/vcloud/v1.5">
  <Disk
    type="application/vnd.vmware.vcloud.disk+xml"
    href="https://vcloud.example.com/api/disk/128" />
  </DiskAttachOrDetachParams>

```

The response is a `Task`.

Response:

```

<Task
  href="https://vcloud.example.com/api/task/57"
  ...
  status="running"

```

```

    operationName="vappAttachDisk"
    ... />
    ...
</Task>

```

Creating and Using vApp Snapshots

You can take a snapshot of all the virtual machines in a vApp. After the snapshots are taken, you can revert all virtual machines in the vApp to the most recent snapshot, or remove all snapshots.

You can take a snapshot of a vApp whether or not it is powered on. The `createSnapshot` operation always creates a snapshot for each virtual machine in the vApp. Snapshots are created in the order specified for virtual machines in the `StartupSection` of the `VApp` element.

vApp snapshots have the following limitations:

- They do not capture the `ovf:ProductSection` elements in the vApp.
- They do not capture NIC configurations.
- They cannot be created if any virtual machine in the vApp is connected to an independent disk.

Prerequisites

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

Procedure

- 1 (Optional) Retrieve information about current vApp snapshots.

When you create a snapshot, it overwrites any existing snapshots without warning. Before creating a new snapshot, you might want to verify whether there are any existing snapshots. Make a request like this one:

```
GET https://vcloud.example.com/api/vApp/vapp-33/snapshotSection
```

The response is a `SnapshotSection` element containing a link to each of the current snapshots for the vApp.

- 2 Create a `CreateSnapshotParams` request body.

See [“Example: Create a vApp Snapshot,”](#) on page 95.

- 3 POST the `CreateSnapshotParams` to the `action/createSnapshot` link of the vApp.

Example: Create a vApp Snapshot

In this example, the `memory` and `quiesce` options are not specified, so default values default to `true`. The `name` attribute is optional. If you omit it from the request, the system generates a name for the snapshot.

Request:

```
POST https://vcloud.example.com/api/vApp/vapp-33/action/createSnapshot
```

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

```
...
```

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

```
<CreateSnapshotParams
```

```
  xmlns="http://www.vmware.com/vcloud/v1.5"
```

```
  name="snap1">
```

```
    <Description>Demo snapshot</Description>
```

```
</CreateSnapshotParams>
```

The response is a `Task`.

Response:

```
<?xml version="1.0" encoding="UTF-8"?>
<Task
  ...
  operationName="vappCreateSnapshot"
  ... />
  ...
</Task>
```

After the snapshot is created, you can revert the vApp to the state captured in the snapshot.

POST <https://vcloud.example.com/api/vApp/vapp-33/action/revertToCurrentSnapshot>

You can also remove all snapshots.

POST <https://vcloud.example.com/api/vApp/vapp-33/action/removeAllSnapshots>

Neither of these requests has a request body. Both return a Task.

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 the power URL of a VApp 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 the power URL of a Vm affects only that virtual machine.
- See [“Summary of vCloud API vApp and Virtual Machine Operations Requests,”](#) on page 83 for details of each request.

Prerequisites

Verify that you are logged in to the vCloud API 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 72.

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, <code>CustomizeOnInstantiate</code> , that specifies whether guest customization should be run during instantiation. You cannot modify this element if the vApp template has been added to a catalog.
Name and description	You can use a PUT request to update the value of the template's <code>name</code> attribute or the contents of its <code>Description</code> element.
Owner	You can use a PUT request to update the value of the template's <code>Owner</code> element. See “View or Change the Owner of an Object,” on page 79.
ProductSection elements	You can retrieve or update the template's <code>ProductSection</code> 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 120.

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

Configuring a vApp

You can include the following sections in the `InstantiationParams` associated with an `instantiateVAppTemplate`, `composeVApp`, or `recomposeVApp` request. You can also modify them in a deployed 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 <code>InstantiationParams</code> element of an <code>instantiateVAppTemplate</code> 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 configure a virtual machine by updating any of the following sections of a `Vm` element:

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.

There are several ways to update these sections:

- You can include any or all of the sections in the `InstantiationParams` for a `Vm`.
- You can include any or all of the sections in a `reconfigureVm` request.
- You can use the workflow in [“Reconfiguration Workflow,”](#) on page 98 to update individual section or groups of related section.

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 193.

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 the 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 to the vCloud API 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 99.

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

The response portion of [“Example: Configuration Links in a vApp,”](#) on page 99 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 101.

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.

IMPORTANT Request URLs are always available in `Link` elements contained by the representation of the object on which they operate. URL forms shown here are for reference purposes only. Although URLs have a well-known syntax and a well-understood interpretation, a client should treat vCloud API request URLs as opaque strings. The rules that govern how the server constructs these strings might change in future releases.

This summary may not cover all requests in this category. For the complete list of requests, along with detailed information about input and output types, see the Operations lists in the schema reference.

Table 5-2. Summary of vApp Reconfiguration Requests

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

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

Operation	Request	Request Body	Response
Retrieve vApp NetworkConfigSection	GET <i>API-URL/vApp/vapp-id/ networkConfigSection/</i>	None	NetworkConfigSection
Update vApp NetworkConfigSection	PUT <i>API-URL/vApp/vapp-id/ networkConfigSection/</i>	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 the 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 to the vCloud API 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 99.
- 2 In the vApp element's `Children` element, find the `vm` element that represents the virtual machine and retrieve it.
- 3 Examine the response to find the `reconfigureVm` link and `edit` links to modifiable sections.

The response portion of [“Example: Configuration Links in a Vm Element,”](#) on page 101 shows the `reconfigureVm` link and links for each of the modifiable sections 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 99. 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 99.

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">
```

```

...
<Link
  rel="reconfigureVm"
  type="application/vnd.vmware.vcloud.vm+xml"
  name="vm1"
  href="https://vcloud.example.com/api/vApp/vm-4/action/reconfigureVm" />
...
<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.

IMPORTANT Request URLs are always available in Link elements contained by the representation of the object on which they operate. URL forms shown here are for reference purposes only. Although URLs have a well-known syntax and a well-understood interpretation, a client should treat vCloud API request URLs as opaque strings. The rules that govern how the server constructs these strings might change in future releases.

This summary may not cover all requests in this category. For the complete list of requests, along with detailed information about input and output types, see the Operations lists in the schema reference.

Table 5-3. Summary of Vm Reconfiguration Requests

Operation	Request	Request Body	Response
Retrieve the NetworkConnectionSection of a virtual machine	GET <i>API-URL</i> /vApp/vm- <i>id</i> /networkConnectionSection/	None	NetworkConnectionSection
Update the NetworkConnectionSection of a virtual machine	PUT <i>API-URL</i> /vApp/vm- <i>id</i> /networkConnectionSection/	NetworkConnectionSection	Task
Retrieve the GuestCustomizationSection of a virtual machine	GET <i>API-URL</i> /vApp/vm- <i>id</i> /guestCustomizationSection/	None	GuestCustomizationSection

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

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

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

Operation	Request	Request Body	Response
Update network card items in the <code>VirtualHardwareSection</code> of a virtual machine	PUT <i>API-URL/vApp/vm-id/virtualHardwareSection/networkCards</i>	<code>RasdItemsList</code>	Task
Update the name, Description, and any or all of <code>VirtualHardwareSection</code> , <code>OperatingSystemSection</code> , <code>NetworkConnectionSection</code> , <code>GuestCustomizationSection</code> of a virtual machine. [NEW]	POST <i>API-URL/vApp/vm-id/action/reconfigureVm</i>	<code>Vm</code>	Task

vCloud API Custom Attributes

vCloud API custom attributes extend several elements in the `ovf` and `rasd` XML 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
<code>rasd:Connection</code>	10 (Network adapters)	<code>ipAddressingMode</code>	<code>xs:string</code>	IP addressing mode to use for this connection. One of NONE, MANUAL, DHCP, POOL.
<code>rasd:Connection</code>	10 (Network adapters)	<code>ipAddress</code>	<code>xs:string</code>	If <code>ipAddressingMode="MANUAL"</code> , set the IP address here
<code>rasd:Connection</code>	10 (Network adapters)	<code>primaryNetworkConnection</code>	<code>xs:boolean</code>	True if this is the primary network connection of the virtual machine
<code>rasd:HostResource</code>	17 (Hard disks)	<code>capacity</code>	<code>xs:string</code>	Hard disk capacity in megabytes. See "Retrieve or Modify the Hard Disk Configuration of a Virtual Machine," on page 124
<code>rasd:HostResource</code>	17 (Hard disks)	<code>busType</code>	<code>xs:string</code>	read-only

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

Element Name	RASD Resource Type	Attribute Name	Attribute Type	Description
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.

Update Multiple Sections of a Virtual Machine

You can make a single request that updates the name, Description, and any or all of the VirtualHardwareSection, OperatingSystemSection, NetworkConnectionSection, GuestCustomizationSection elements of a virtual machine.

Every Vm element contains a link to a reconfigureVm operation that you can use to update the name, Description, and multiple sections in a single operation. Sections that you omit from the request body are not updated.

Prerequisites

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

Procedure

- 1 Retrieve the Vm element that you want to update.
- 2 Modify the retrieved Vm element.

Modifications can include the name, Description, and any or all of the VirtualHardwareSection, OperatingSystemSection, NetworkConnectionSection, GuestCustomizationSection elements of the Vm.

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 Use the modified Vm as the body of a reconfigureVm request.

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

Example: Update Multiple Sections of a Virtual Machine

This example uses the reconfigureVm operation to accomplish the updates shown in “[Example: Update a NetworkConnectionSection](#),” on page 116 and “[Example: Modify the Guest Customization Section of a Virtual Machine](#),” on page 119 in a single operation.

Request:

```
POST https://vcloud.example.com/api/vApp/vm-4/action/reconfigureVm
Content-Type: application/vnd.vmware.vcloud.vm+xml
...
<?xml version="1.0" encoding="UTF-8"?>
<Vm ... >
  ...
  <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>
  <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>
  ...
</Vm>
```

Response:

```
202 Accepted
Content-Type: application/vnd.vmware.vcloud.task+xml
...
<Task
  ...
  operation="Updating Virtual Application Win2K3 (4)"
  ...>
  ...
</Task>
```

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 to the vCloud API 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 99.

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>
```

```

<IpScopes>
  <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>
</IpScopes>
<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 110 .

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 to the vCloud API 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 108. 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 101.

For more information about these and other network services in vApp networks, see [“Network Services in vApp Networks,”](#) on page 112

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 reference.

Request:

```
PUT https://vcloud.example.com/api/vApp/vapp-7/networkConfigSection/
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">
  <ovf:Info>Configuration parameters for logical networks</ovf:Info>
  <NetworkConfig
    networkName="vAppNetwork">
    <Configuration>
      <IpScopes>
        <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>
      </IpScopes>
      <ParentNetwork
        type="application/vnd.vmware.vcloud.network+xml"
        name="Internet"
        href="https://vcloud.example.com/api/network/54" />
    </Configuration>
  </NetworkConfig>
</NetworkConfigSection>
```

```

<FenceMode>natRouted</FenceMode>
<Features>
  <FirewallService>
    <IsEnabled>true</IsEnabled>
    <FirewallRule>
      <IsEnabled>true</IsEnabled>
      <Description>FTP Rule</Description>
      <Policy>allow</Policy>
      <Protocols>
        <Tcp>true</Tcp>
      </Protocols>
      <DestinationPortRange>21</DestinationPortRange>
      <DestinationIp>10.147.115.1</DestinationIp>
      <SourcePortRange>any</SourcePortRange>
      <SourceIp>any</SourceIp>
      <EnableLogging>>false</EnableLogging>
    </FirewallRule>
    <FirewallRule>
      <IsEnabled>true</IsEnabled>
      <Description>SSH Rule</Description>
      <Policy>allow</Policy>
      <Protocols>
        <Tcp>true</Tcp>
      </Protocols>
      <DestinationPortRange>22</DestinationPortRange>
      <DestinationIp>10.147.115.1</DestinationIp>
      <SourcePortRange>any</SourcePortRange>
      <SourceIp>any</SourceIp>
      <EnableLogging>>false</EnableLogging>
    </FirewallRule>
  </FirewallService>
  <NatService>
    <IsEnabled>true</IsEnabled>
    <NatType>ipTranslation</NatType>
    <Policy>allowTraffic</Policy>
    <NatRule>
      <OneToOneVmRule>
        <MappingMode>automatic</MappingMode>
        <VAppScopedVmId>3963994b-5a0a-48fe-b9ae-7f9a2d8e8e5b</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 101, specifies an IP address that will not be available after the vApp network is reconfigured as shown here. [“Example: Update a NetworkConnectionSection,”](#) on page 116 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 116. 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 116 allows the `FirewallRules` in this example to remain valid.

Network Services in vApp Networks

The `Features` element of a vApp `NetworkConfigSection` defines the network services available to virtual machines in the vApp.

A vApp network can be configured to provide many of the same kinds of services available in an organization vDC network. Configuration parameters for these services are similar to those of their counterparts on an Edge Gateway, but scoped to the needs of a vApp network.

For more information about vCloud Director networks, see [“About vCloud Director Networks,”](#) on page 148. For more information about network services for organization vDC networks, see [“Configure Edge Gateway Services,”](#) on page 156

DHCP Service

A `DhcpService` element defines an IP address range and lease policies for a DHCP service that can be used by to virtual machines in the vApp. Unlike a DHCP service in an Edge Gateway, it can support only a single IP address range, as shown in this example.

```
<DhcpService>
  <IsEnabled>true</IsEnabled>
  <DefaultLeaseTime>3600</DefaultLeaseTime>
  <MaxLeaseTime>7200</MaxLeaseTime>
  <IpRange>
    <StartAddress>192.168.3.2</StartAddress>
    <EndAddress>192.168.3.99</EndAddress>
  </IpRange>
</DhcpService>
```


Firewall Service

A FirewallService element defines firewall rules that, when matched, block or allow incoming or outgoing traffic on the vApp network. A firewall rule in a vApp network can specify the destination as a combination of address and port, or as a specific virtual NIC in a Vm. This FirewallService allows TCP traffic to ports 21 and 22.

```
<FirewallService>
  <IsEnabled>true</IsEnabled>
  <FirewallRule>
    <IsEnabled>true</IsEnabled>
    <Description>FTP Rule</Description>
    <Policy>allow</Policy>
    <Protocols>
      <Tcp>true</Tcp>
    </Protocols>
    <DestinationPortRange>21</DestinationPortRange>
    <DestinationIp>10.147.115.1</DestinationIp>
    <SourcePortRange>any</SourcePortRange>
    <SourceIp>any</SourceIp>
    <EnableLogging>>false</EnableLogging>
  </FirewallRule>
  <FirewallRule>
    <IsEnabled>true</IsEnabled>
    <Description>SSH Rule</Description>
    <Policy>allow</Policy>
    <Protocols>
      <Tcp>true</Tcp>
    </Protocols>
    <DestinationPortRange>22</DestinationPortRange>
    <DestinationIp>10.147.115.1</DestinationIp>
    <SourcePortRange>any</SourcePortRange>
    <SourceIp>any</SourceIp>
    <EnableLogging>>false</EnableLogging>
  </FirewallRule>
</FirewallService>
```

You can see this example in the context of a vApp NetworkConfigSection in [“Example: Update a NetworkConfigSection,”](#) on page 110

An alternate implementation of the second FirewallRule in this example includes a DestinationVm element that specifies the destination as a specific virtual NIC (identified in the VmNicId element) in a specific Vm (identified in the VAppScopedVmId element. The value of VAppScopedVmId 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 101. The IpType is set to assigned, indicating that the NIC retains its assigned IP address. If you set IpType is set to NAT, the IP address of the NIC is its translated address.

```
<FirewallRule>
  <IsEnabled>true</IsEnabled>
  <Description>allow ssh to a specific NIC in a specific Vm</Description>
  <Policy>allow</Policy>
  <Protocols>
    <Tcp>true</Tcp>
  </Protocols>
  <DestinationPortRange>22</DestinationPortRange>
  <DestinationVm>
```

```

    <VAppScopedVmId>3963994b-5a0a-48fe-b9ae-7f9a2d8e8e5b</VAppScopedVmId>
    <VmNicId>0</VmNicId>
    <IpType>assigned</IpType>
  </DestinationVm>
  <SourcePortRange>Any</SourcePortRange>
  <SourceIp>Any</SourceIp>
  <EnableLogging>>false</EnableLogging>
</FirewallRule>

```

NAT Service

A `NatService` element defines network address translation services to virtual machines on the network. This simple `NatService` defines a single rule that implements an IP translation strategy for a single VM.

```

<NatService>
  <IsEnabled>>true</IsEnabled>
  <NatType>ipTranslation</NatType>
  <Policy>allowTraffic</Policy>
  <NatRule>
    <OneToOneVmRule>
      <MappingMode>automatic</MappingMode>
      <VAppScopedVmId>3963994b-5a0a-48fe-b9ae-7f9a2d8e8e5b</VAppScopedVmId>
      <VmNicId>0</VmNicId>
    </OneToOneVmRule>
  </NatRule>
</NatService>

```

You can see this example in the context of a vApp `NetworkConfigSection` in [“Example: Update a NetworkConfigSection,”](#) on page 110

A `NatService` element like this one configures the service to use port forwarding instead of IP translation. Instead of using a `OneToOneVmRule`, which specifies one external IP address to one NIC, it uses a `VmRule` element, which enables port forwarding by allowing one external IP address to be forward to different ports on different virtual machines.

```

<NatService>
  <IsEnabled>>true</IsEnabled>
  <NatType>portForwarding</NatType>
  <Policy>allowTraffic</Policy>
  <NatRule>
    <VmRule>
      <ExternalPort>22</ExternalPort>
      <VAppScopedVmId>3963994b-5a0a-48fe-b9ae-7f9a2d8e8e5b</VAppScopedVmId>
      <VmNicId>0</VmNicId>
      <InternalPort>22</InternalPort>
      <Protocol>TCP</Protocol>
    </VmRule>
  </NatRule>
</NatService>

```

Static Routing Service

A `StaticRoutingService` specifies static routes to other networks. In addition to creating static routes from organization vDC networks on an `EdgeGateway` (see [“Example: Static Routes Between Organization vDC Networks,”](#) on page 161, you can create static routes between vApp networks if they both define the same `ParentNetwork`. 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 vDC 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>
```

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 to the vCloud API 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 101 so that this network connection is compatible with the reconfigured vApp network to which it must connect. See [“Example: Update a NetworkConfigSection,”](#) on page 110. 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`. The element must be present, even if it has no content. 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 to the vCloud API 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 101. 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. Certain values in this element, if omitted or left empty, are inherited from the `OrgGuestPersonalizationSettings` of the organization that owns the virtual machine. See [“Retrieve or Update Organization Settings,”](#) on page 136

The vCloud API also 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 120.

Prerequisites

Verify that you are logged in to the vCloud API 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

This request specifies guest customization values, including the information required to join the virtual machine to a Windows domain.

NOTE If you include a CustomizationScript, it cannot exceed 49,000 characters.

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>example</DomainName>
  <DomainUserName>admin</DomainUserName>
  <DomainUserPassword>Pa55w0rd</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 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 to the vCloud API 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 Username</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 101, 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 to the vCloud API 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 123.

- 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 101 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 to the vCloud API 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 123.

- 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 RasdItemList 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.rasditemlist+xml
```

```
...
```

```
<?xml version="1.0" encoding="UTF-8"?>
<RasdItemList
  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.rasditemlist+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>
</RasdItemList>

```

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>

```

Update the Storage Profile for a Virtual Machine

You can update a VM to revalidate the storage profile it uses or specify a different storage profile. Revalidation of a virtual machine's current storage profile is required whenever the datastore that supports the virtual machine changes.

Every VM element includes a StorageProfile element. The value of the href attribute of that element is a reference to the virtual machine's storage profile. The initial value of this attribute is inherited from the vDC that contains it unless you specify the value when the virtual machine is created. To change the value, you must update the entire VM element that contains it.

NOTE When the system administrator changes the datastore that stores a virtual machine, you must update the VM element as shown in [“Example: Update the Storage Profile for a Virtual Machine,”](#) on page 127, but leave the href of the current StorageProfile unchanged. This action, which replaces the deprecated relocate action, forces revalidation of the existing storage profile. If the current datastore is disabled or no longer supports the specified storage profile, the system relocates the virtual machine to a different datastore that supports the referenced storage profile. After the returned Task completes, the validation and, if necessary, relocation is complete.

Prerequisites

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

Procedure

- 1 Retrieve the VM element.

Make a GET request to the URL in the value of the href attribute of the VM.

- 2 Modify the retrieved Vm to change the StorageProfile reference.

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

The response to this request is a Task element that tracks the relocation of the virtual machine to a datastore in the new storage profile. When the task is complete, the virtual machine has been relocated.

Example: Update the Storage Profile for a Virtual Machine

This example shows a Vm element containing a StorageProfile. The actual update operation requires the entire Vm element, including the StorageProfile, in the request body. Only a small part of the element appears in this example.

Request:

```
PUT https://vcloud.example.com/api/vApp/vm-4
Content-type: application/vnd.vmware.vcloud.vm+xml
...
<?xml version="1.0" encoding="UTF-8"?>
<Vm ...>
...
  <StorageProfile
    type="application/vnd.vmware.vcloud.vdcStorageProfile+xml"
    name="Gold"
    href="https://vcloud.example.com/api/vdcStorageProfile/3" />
  </Vm>
```

Response:

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


Creating and Managing Organizations

The VMware vCloud API supports objects and operations that an organization or a system administrator can use to automate tasks associated with creating and managing organizations and their vDCs, networks, catalogs, 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 49.

This chapter includes the following topics:

- [“Summary of Administrative Requests,”](#) on page 129
- [“Administrator Credentials and Privileges,”](#) on page 132
- [“Organization Administration,”](#) on page 132
- [“vDC Administration,”](#) on page 139
- [“Network Administration,”](#) on page 147
- [“Catalog Administration,”](#) on page 176
- [“User and Group Administration,”](#) on page 179
- [“Working With Roles and Rights,”](#) on page 187
- [“Controlling Access to vApps and Catalogs,”](#) on page 191

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.

IMPORTANT Request URLs are always available in `Link` elements contained by the representation of the object on which they operate. URL forms shown here are for reference purposes only. Although URLs have a well-known syntax and a well-understood interpretation, a client should treat vCloud API request URLs as opaque strings. The rules that govern how the server constructs these strings might change in future releases.

This summary may not cover all requests in this category. For the complete list of requests, along with detailed information about input and output types, see the Operations lists in the schema reference.

GET *API-URL/admin/vdc/id/networks*

Table 6-1. Summary of Administrative Requests

Operation	Request	Request Body	Response
Retrieve an administrative view of an organization.	GET <i>API-URL/admin/org/id</i>	None	AdminOrg
Create an organization.	POST <i>API-URL/admin/orgs</i>	AdminOrg	AdminOrg
Enable an organization.	POST <i>API-URL/admin/org/id/action/enable</i>	None	204 No Content
Disable an organization.	POST <i>API-URL/admin/org/id/action/disable</i>	None	204 No Content
Update an organization.	PUT <i>API-URL/admin/org/id</i>	AdminOrg	AdminOrg
Create an Edge Gateway. [NEW]	POST <i>API-URL/admin/vdc/id/edgeGateways</i>	EdgeGateway	EdgeGateway
Retrieve an Edge Gateway. [NEW]	GET <i>API-URL/admin/edgeGateway/id</i>	None	EdgeGateway
Update Edge Gateway services. [NEW]	POST <i>API-URL/admin/edgeGateway/id/action/configureServices</i>	EdgeGatewayServiceConfiguration	Task
Reapply Edge Gateway Services [NEW]	POST <i>API-URL/admin/edgeGateway/id/action/reapplyServices</i>	None	204 No Content
Redeploy an Edge Gateway [NEW]	POST <i>API-URL/admin/edgeGateway/id/action/redeploy</i>	None	204 No Content
Delete an Edge Gateway. [NEW]	DELETE <i>API-URL/admin/edgeGateway/id</i>	None	Task
Create an organization vDC network. [NEW]	POST <i>API-URL/admin/vdc/id/networks</i>	OrgVdcNetwork	Task
Update an organization vDC network. [NEW]	PUT <i>API-URL/admin/network/id</i>	OrgVdcNetwork	Task
Retrieve a list of networks in this organization vDC. [NEW]	GET <i>API-URL/admin/vdc/id/networks</i>	None	QueryResultRecords
Delete an organization vDC network. [NEW]	DELETE <i>API-URL/admin/network/id</i>	None	Task
Synchronize syslog server settings for a vApp network.	POST <i>API-URL/admin/network/id/action/syncSyslogServerSettings</i>	None	Task
Synchronize syslog server settings for an Edge Gateway. [NEW]	POST <i>API-URL/admin/edgeGateway/id/action/syncSyslogServerSettings</i>	None	Task
Reset a network.	POST <i>API-URL/admin/network/id/action/reset</i>	None	Task
Retrieve the admin view of a provider vDC.	GET <i>API-URL/admin/providervdc/id</i>	None	ProviderVdc

Table 6-1. Summary of Administrative Requests (Continued)

Operation	Request	Request Body	Response
List the organization vDCs supported by a provider vDC.	GET <i>API-URL/admin/providervdc/id/vdcReferences</i>	None	VdcReferences
Create an organization vDC.	POST <i>API-URL/admin/org/id/vdcs</i>	AdminVdc	AdminVdc
Enable an organization vDC.	POST <i>API-URL/admin/vdc/id/action/enable</i>	None	204 No Content
Disable an organization vDC.	POST <i>API-URL/admin/vdc/id/action/disable</i>	None	204 No Content
Delete an organization vDC.	DELETE <i>API-URL/admin/vdc/id</i>	None	Task
Create a catalog.	POST <i>API-URL/admin/org/id/catalogs</i>	AdminCatalog	AdminCatalog
Publish a catalog.	POST <i>API-URL/admin/catalog/id/action/publish</i>	PublishCatalogParams	204 No Content
Update catalog properties.	PUT <i>API-URL/admin/catalog/id</i>	AdminCatalog	AdminCatalog
Delete a catalog.	DELETE <i>API-URL/admin/catalog/id</i>	None	204 No Content
Update the owner of a catalog.	PUT <i>API-URL/admin/catalog/id/owner</i>	Owner	204 No Content
Create or import a user.	POST <i>API-URL/admin/org/id/users</i>	User	User
Update user properties.	PUT <i>API-URL/admin/user/id</i>	User	User
Delete a user.	DELETE <i>API-URL/admin/user/id</i>	None	204 No Content
Import a group.	POST <i>API-URL/admin/org/id/groups</i>	Group	Group
Delete a group.	DELETE <i>API-URL/admin/group/id</i>	None	204 No Content
Update group properties.	PUT <i>API-URL/admin/group/id</i>	Group	Group
Create a role.	POST <i>API-URL/admin/org/id/roles</i>	Role	Role
Update role properties.	PUT <i>API-URL/admin/role/id</i>	Role	Role
Delete a role.	DELETE <i>API-URL/admin/role/id</i>	None	204 No Content
Retrieve a list of tasks owned by organization <i>id</i> .	GET <i>API-URL/tasksList/id</i>	None	Taskslist
Retrieve a task.	GET <i>API-URL/task/id</i>	None	Task
Cancel a task.	POST <i>API-URL/task/id/action/cancel</i>	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, external networks, network pools, and similar system-level objects.

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`.

NOTE When logging in using a SAML identity provider, the system administrator must use the vSphere SSO Service as the identity provider. See [“Create a Login Session Using a SAML Identity Provider,”](#) on page 46.

The System Organization

The `System` organization is created automatically when vCloud Director is installed. Unlike the organizations represented by `Org` and `AdminOrg` objects, the `System` organization cannot contain catalogs, vDCs, groups, or users who are not system administrators.

Example: The System Organization

When a system administrator logs in to the REST API, the `OrgList` in the returned `Session` element contains a link to the `System` organization.

```
<OrgList ... >
  ...
  <Org
    type="application/vnd.vmware.admin.systemOrganization+xml"
    name="System"
    href="https://vcloud.example.com/api/admin/org/123"/>
  ...
</OrgList>
```

Organization Administration

System administrators create organizations and organization administrators, and establish certain organization policies. Organization administrators populate their organization with users and groups, assign roles, and can update most organization policies and properties.

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. User credentials are authenticated by the organization's identity provider, which can be either the integrated identity provider included in vCloud Director or an external SAML-based identity provider.

Create an Organization

To create an organization, a system administrator POSTs an `AdminOrg` element to the cloud's add URL for orgs.

Prerequisites

Verify that you are logged in to the vCloud API 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 133.
- 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 133.

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 133, 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>
  <OrgGeneralSettings>
    <CanPublishCatalogs>true</CanPublishCatalogs>
    <DeployedVMQuota>0</DeployedVMQuota>
    <StoredVmQuota>0</StoredVmQuota>
    <UseServerBootSequence>>false</UseServerBootSequence>
    <DelayAfterPowerOnSeconds>0</DelayAfterPowerOnSeconds>
  </OrgGeneralSettings>
  <OrgLdapSettings>
    <OrgLdapMode>SYSTEM</OrgLdapMode>
    <CustomUsersOu />
  </OrgLdapSettings>
  <OrgEmailSettings>
    <IsDefaultSmtpServer>true</IsDefaultSmtpServer>
    <IsDefaultOrgEmail>true</IsDefaultOrgEmail>
    <FromEmailAddress />
    <DefaultSubjectPrefix />
    <IsAlertEmailToAllAdmins>true</IsAlertEmailToAllAdmins>
  </OrgEmailSettings>
</Settings>
</AdminOrg>

```

The response contains information extracted from 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 unless the create request includes an IsEnabled element with a value of true. A system administrator must enable a disabled AdminOrg before users can log into it. See [“Example: Enable an Organization,”](#) on page 139

The response also includes elements inherited from system defaults, such as OrgPasswordPolicySettings, VAppLeaseSettings, and VAppTemplateLeaseSettings. The full content of these elements appears in the actual response. This example shows only a subset.

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="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" />
  <Link
    rel="down"
    type="application/vnd.vmware.admin.operationLimitsSettings+xml"
    href="https://vcloud.example.com/api/admin/org/26/settings/operationLimitsSettings" />
  <Link

```

```

    rel="edit"
    type="application/vnd.vmware.admin.orgSettings+xml"
    href="https://vcloud.example.com/api/admin/org/26/settings" />
<OrgGeneralSettings
  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>
</OrgGeneralSettings>
<VAppLeaseSettings>
  ...
</VAppLeaseSettings>
<VAppTemplateLeaseSettings>
</VAppTemplateLeaseSettings>
<OrgLdapSettings>
  ...
</OrgLdapSettings>
<OrgEmailSettings>
  ...
</OrgEmailSettings>
<OrgPasswordPolicySettings>
  ...
</OrgPasswordPolicySettings>
<OrgOperationLimitsSettings>
  ...
</OrgOperationLimitsSettings>
<OrgFederationSettings>
  ...
</OrgFederationSettings>
</Settings>
</AdminOrg>

```

Retrieve or Update Organization Settings

Organization settings define organization policies such as default lease settings for vApps and how incorrect login attempts are handled. They also configure how the organization uses services such as email and LDAP, and define the organization's federated identity provider if it uses one.

An AdminOrg element contains an OrgSettings element, which contains the following elements, each of which represents a group of related organization settings. Default settings are inherited from the containing cloud.

GeneralOrgSettings	Specifies quotas for stored and deployed virtual machines owned by members of this organization. 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 service defined in the system <code>LdapSettings</code> or a custom LDAP service defined here.
OrgEmailSettings	Defines whether this organization uses the email service defined in the system <code>EmailSettings</code> or a custom email service defined here.
OrgPasswordPolicySettings	Specifies policies to be followed when a user in this organization enters an incorrect password. Initial values are inherited from the system <code>PasswordPolicySettings</code> .
OrgOperationLimitsSettings	Specifies limits to be placed on simultaneous resource-intensive operations and console sessions for members of this organization.
OrgGuestPersonalizationSettings	Default values for <code>GuestCustomizationSection</code> elements in virtual machines created by this organization. See “Retrieve or Modify the GuestCustomizationSection of a Virtual Machine,” on page 118
OrgFederationSettings	Defines the SAML identity provider used by this organization. An organization can define a SAML identity provider that it shares with other applications or enterprises. Users that authenticate to the identity provider obtain a token that they can then use to log in to the organization. Such a strategy can enable an enterprise to provide access to multiple, unrelated services, including vCloud Director with a single set of credentials, an arrangement often referred to as “single sign-on.” An organization that wants to participate in a federated identity scheme must include an <code>OrgFederationSettings</code> element that contains SAML metadata retrieved from the federation’s identity provider. By default, this element is empty.

NOTE To update or remove `OrgFederationSettings` after a SAML identity provider has been specified, you must include an empty `SAMLMetadata` element in the update request. If this element is not present in the update request, the `OrgFederationSettings` are not changed.

Prerequisites

Verify that you are logged in to the vCloud API 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 138.

Example: Update Organization General Settings

This example updates the GeneralOrgSettings of the organization created in [“Example: Create an Organization,”](#) on page 133. 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 in an update request. 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 contains information extracted from 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 to the vCloud API as a system administrator.
- Retrieve the XML representation of the organization. See [“Retrieve a List of Organizations Accessible to You,”](#) on page 48.

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 133.

Request:

```
POST https://vcloud.example.com/api/admin/org/26/action/enable
```

Response:

```
204 No Content
```

vDC Administration

A newly created organization has no vDCs in it. A system administrator must use system resources to create them.

An organization virtual datacenter (organization vDC) is a deployment environment for virtual systems owned by the containing organization, and an allocation mechanism for resources such as networks, storage, CPU, and memory. In an organization 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 209.
- 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 a `CreateVdcParams` element to an organization's `add` URL for vDCs.

Prerequisites

Verify that you are logged in to the vCloud API as a system administrator.

Retrieve the list of network pools. Several types of organization vDC networks require the vDC to include a network pool, which you can specify when you create or update the vDC. See [“Retrieve a List of External Networks and Network Pools,”](#) on page 204 for information about how to retrieve this list.

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.createVdcParams+xml`, as shown here:

```
<Link
  href="https://vcloud.example.com/api/admin/org/26/vdcparams"
  rel="add"
  type="application/vnd.vmware.admin.createVdcParams+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 in the system 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 a CreateVdcParams request body.
 - a Include an AllocationModel element that specifies how compute resources are allocated by this vDC. Choose one of the following values for AllocationModel:

AllocationVApp Pay as you go. Resources are committed to the organization 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 you retrieve a vApp. Resources available to this kind of organization 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.

- b Include at least one VdcStorageProfile element that specifies a ProviderVdcStorageProfile defined in the Provider vDC you chose in [Step 3](#).
- c Include a NetworkPoolReference element.
The vDC must include a network pool if you want to create routed or isolated networks in it.
- d Include a ProviderVdcReference element that contains a reference to the Provider vDC you chose in [Step 3](#).

See the request portion of [“Example: Create an Organization vDC,”](#) on page 141.

- 5 POST the CreateVdcParams request body to the organization's add link for vdc.
See the request portion of [“Example: Create an Organization vDC,”](#) on page 141.

The server creates the new vDC in the specified organization and returns an AdminVdc element that includes 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: Create an Organization vDC,”](#) on page 141.

Example: Create an Organization vDC

This example adds an AllocationVApp vDC to the organization created in [“Example: Create an Organization,”](#) on page 133. The new organization vDC is provisioned from the Provider vDC created in [“Create a Provider vDC,”](#) on page 209, and includes a storage profile named Silver, which is backed by one of the storage profiles available in the Provider vDC. It also includes a network pool, so that it is capable of supporting routed and isolated organization vDC networks. See [“Retrieve a List of External Networks and Network Pools,”](#) on page 204 for information on how to find a NetworkPoolReference to use.

Request:

```
POST https://vcloud.example.com/api/admin/org/26/vdcparams
Content-Type: application/vnd.vmware.admin.createVdcParams+xml
...
<?xml version="1.0" encoding="UTF-8"?>
<CreateVdcParams
  name="org26vdc1"
  xmlns="http://www.vmware.com/vcloud/v1.5">
  <Description>Example vDC</Description>
  <AllocationModel>AllocationVApp</AllocationModel>
```

```

<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>100</NetworkQuota>
<VdcStorageProfile>
<Enabled>true</Enabled>
  <Units>MB</Units>
  <Limit>20480</Limit>
  <Default>true</Default>
  <ProviderVdcStorageProfile
    href="https://vcloud.example.com/api/admin/pvdcStorageProfile/101" />
</VdcStorageProfile>
<ResourceGuaranteedMemory>1</ResourceGuaranteedMemory>
<ResourceGuaranteedCpu>1</ResourceGuaranteedCpu>
<VCpuInMhz>2048</VCpuInMhz>
<IsThinProvision>false</IsThinProvision>
<NetworkPoolReference
  href="https://vcloud.example.com/api/admin/extension/networkPool/313"/>
<ProviderVdcReference
  name="Main Provider"
  href="https://vcloud.example.com/api/admin/providervdc/35" />
<UsesFastProvisioning>true</UsesFastProvisioning>
</CreateVdcParams>

```

The response, a subset of which appears here, contains information extracted from 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 virtual hardware architectures that the vDC supports. These elements are retrieved from the Provider vDC that you specified when you created the `CreateVdcParams`. 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
    name="task"
    status="running"
    operation="Creating Virtual Datacenter org26vdc1(44)"
    ...
  </Task>
</Tasks>
<AllocationModel>AllocationVApp</AllocationModel>
...
<Capabilities>
  <SupportedHardwareVersions>
    <SupportedHardwareVersion>vmx-04</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 vSphere resource pool that supports the vDC appears in a VCloudExtension element.
- There is an empty ResourceEntities element, because the vDC contains no Media, VAppTemplate, or Disk entities. For information about adding them, see [Chapter 4, “Provisioning an Organization,”](#) on page 55.
- There is an empty AvailableNetworks element. To add networks to this organization vDC, see [“Create an Organization vDC Network,”](#) on page 166.
- 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:VmObjectRef>

```

```

    <vmext:VmServerRef
      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:VmObjectType>RESOURCE_POOL</vmext:VmObjectType>
  </vmext:VmObjectRef>
</VCloudExtension>
<Link
  rel="up"
  type="application/vnd.vmware.admin.organization+xml"
  href="https://vcloud.example.com/api/admin/org/e0b93bca-5dc2-453c-b3dc-bba8067d32b6" />
<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="add"
  type="application/vnd.vmware.admin.edgeGateway+xml"
  href="https://vcloud.example.com/api/admin/vdc/44/edgeGateways" />
<Link
  rel="edgeGateways"
  type="application/vnd.vmware.vcloud.query.records+xml"
  href="https://vcloud.example.com/api/admin/vdc/44/edgeGateways" />
<Link
  rel="add"
  type="application/vnd.vmware.vcloud.orgVdcNetwork+xml"
  href="https://vcloud.example.com/api/admin/vdc/44/networks" />
<Link
  rel="orgVdcNetworks"
  type="application/vnd.vmware.vcloud.query.records+xml"
  href="https://vcloud.example.com/api/admin/vdc/44/networks" />
<Link
  rel="down"
  type="application/vnd.vmware.admin.OrganizationVdcResourcePoolSet+xml"
  href="https://vcloud.example.com/api/admin/extension/vdc/44/resourcePools" />
<Link
  rel="edit"
  type="application/vnd.vmware.admin.updateVdcStorageProfiles+xml"
  href="https://vcloud.example.com/api/admin/vdc/44/vdcStorageProfiles" />
...
<ResourceEntities />
<AvailableNetworks />
...

```



```

<VdcStorageProfiles name="Silver">
  <Units>MB</Units>
  <Limit>20480</Limit>
  <Default>>false</Default>
</VdcStorageProfiles>
...
</AdminVdc>

```

Update Organization vDC Storage Profiles

A system administrator can update the storage profiles that are available in an organization vDC. You can add new storage profiles and remove unused storage profiles.

An organization vDC storage profile allocates a subset of the storage available in a Provider vDC storage profile for use by vApp templates, virtual machines, and media objects in the organization vDC. For each organization vDC storage profile you create, you must specify a storage limit, which cannot exceed the storage available in the Provider vDC storage profile (the value of `CapacityTotal-CapacityUsed` in the `ProviderVdcStorageProfile`). When you update organization vDC storage profiles, you can change the default storage profile and modify the limits on existing storage profiles.

Prerequisites

Verify that you are logged in to the vCloud API as a system administrator.

Procedure

- 1 Retrieve the XML representation of the vDC in the admin view.

Use a request like this one:

```
GET https://vcloud.example.com/api/admin/vdc/44
```

- 2 Examine the `AdminVdc` element to find the `vdcStorageProfiles` link, `VdcStorageProfiles` element, and `ProviderVdcReference` element it contains.

The `vdcStorageProfiles` link has the following form:

```

<Link
  rel="edit"
  type="application/vnd.vmware.admin.updateVdcStorageProfiles+xml"
  href="https://vcloud.example.com/api/admin/vdc/44/vdcStorageProfiles" />

```

- 3 Create an `UpdateVdcStorageProfiles` request body that specifies the details of the update.

To add a storage profile:

- a Select a storage profile from the Provider vDC referenced in the `ProviderVdcReference` element of the vDC you are updating.

The storage profile must not be listed in the `VdcStorageProfiles` element of the vDC you are updating.

- b Include an `AddStorageProfile` element in the `UpdateVdcStorageProfiles` request body.

The `AddStorageProfile` element must specify values for `Units`, `Limit`, and `Default`, and must include a reference to the Provider vDC storage profile on which it is based. You can add multiple storage profiles in a single request. Only one of them can specify `Default` as `true`. If any `AddStorageProfile` element specifies `Default` as `true`, that storage profile becomes the new default storage profile for the vDC.

To remove a storage profile:

- a Examine the `VdcStorageProfiles` element and find the profile to remove.
- b Verify that it is not the default storage profile for the vDC, and that no virtual machines are using it.

You can use the `adminVm` query and filter on the `storageProfileName` attribute to list all storage profiles that are in use.

- c Create an `UpdateVdcStorageProfiles` element that contains a `DeleteStorageProfile` element for each storage profile to remove.

4 POST the `UpdateVdcStorageProfiles` element to the vDC's `vdcStorageProfiles` link.

Example: Update vDC Storage Profiles

This request adds a storage profile to the vDC created in [“Example: Create an Organization vDC,”](#) on page 141. The new storage profile is one of the profiles available from the Provider vDC that backs this organization vDC.

One way to retrieve a list of all the Provider vDC storage profiles available from a specific Provider vDC is to use the query service. This query applies a filter that selects only those storage profiles available from the Provider vDC that backs the organization vDC created in [“Example: Create an Organization vDC,”](#) on page 141.

```
GET https://vcloud.example.com/api/query?type=providerVdcStorageProfile&format=references
&filter=providerVdc==https://vcloud.example.com/api/admin/providervdc/35
```

The response might look something like this:

```
<?xml version="1.0" encoding="UTF-8"?>
<ProviderVdcStorageProfileReferences .../>
...
  <ProviderVdcStorageProfileReference
    type="application/vnd.vmware.admin.pvdcStorageProfile+xml"
    name="Gold"
    id="urn:vcloud:providervdcstorageprofile:101"
    href="https://vcloud.example.com/api/admin/pvdcStorageProfile/101" />
  <ProviderVdcStorageProfileReference
    type="application/vnd.vmware.admin.pvdcStorageProfile+xml"
    name="Silver"
    id="urn:vcloud:providervdcstorageprofile:128"
    href="https://vcloud.example.com/api/admin/pvdcStorageProfile/128" />
  ...
</ProviderVdcStorageProfileReferences>
```

Using information, you can construct the `AddStorageProfile` element in the request body.

Request:

```
POST https://vcloud.example.com/api/admin/vdc/44/vdcStorageProfiles
Content-Type: application/vnd.vmware.admin.updateVdcStorageProfiles+xml
...
<?xml version="1.0" encoding="UTF-8"?>
<UpdateVdcStorageProfiles
  xmlns="http://www.vmware.com/vcloud/v1.5" >
  <AddStorageProfile>
    <Enabled>true</Enabled>
    <Units>MB</Units>
    <Limit>5038</Limit>
    <Default>>false</Default>
```

```

    <ProviderVdcStorageProfile
      href="https://vcloud.example.com/api/admin/pvdcStorageProfile/128" />
  </AddStorageProfile>
</UpdateVdcStorageProfiles>

```

Response:

```

202 Accepted
Content-Type: application/vnd.vmware.vcloud.task+xml
...
<Task ... >
  ...
</Task>

```

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 the vDC.

Prerequisites

Verify that you are logged in to the vCloud API 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.

When you disable an organization vDC, you prevent further use of its compute and storage resources. Running vApps and powered on virtual machines continue to run, but you cannot create or start additional vApps or virtual machines.

- 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 Relocate or delete all vApps, vApp templates, and media in the vDC.
 - c 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.

Network Administration

A newly created organization vDC has no networks in it. After a system administrator has created the required network infrastructure, an organization administrator can create and manage most types of organization vDC networks.

An organization vDC can be provisioned with one or more networks. These organization vDC networks can be configured to provide direct or routed connections to external networks, or can be isolated from external networks and other organization vDC networks. Routed connections require an Edge Gateway and network pool in the vDC. The Edge Gateway provides firewall, network address translation, static routing, VPN, and load balancing services.

About vCloud Director Networks

There are three categories of vCloud Director networks: external networks, organization vDC networks, and vApp networks. Additional infrastructure objects such as Edge Gateways and network pools are required by most categories of networks.

You must be a system administrator to create an external network, a directly connected organization vDC network, a network pool, or an Edge Gateway. An organization administrator can create and modify routed and isolated organization vDC networks, and any user who has vApp Author rights can create and modify a vApp network.

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 vDC 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 that you can use to retrieve details of a vApp network that it contains, as the following example shows.

```
<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 or create a new one, you must find the `NetworkConfigSection` of the `VApp` element and use its `edit` link, as shown in [“Update a vApp Network Configuration,”](#) on page 109.

The configuration of a vApp network, contained in the `NetworkConfig` element of the `InstantiateVAppTemplateParams` request body, 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 vDC 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, and can include the following services:

DhcpService	Provides DHCP services to virtual machines on the network.
FirewallService	Specifies firewall rules that, when matched, block or allow incoming or outgoing network traffic.
NatService	Provides network address translation services to virtual machines on the network.
StaticRoutingService	Specifies static routes to other networks. Requires a routed organization vDC network.

For more information, see [“Network Services in vApp Networks,”](#) on page 112

- Additional modifiable elements like `IpScopes` and `RetainNetInfoAcrossDeployments`, and read-only elements such as `SyslogServerSettings` and `RouterInfo`. For more information about the type and scope of these elements, see the schema reference.
- Network pool resources required by an `isolated` or `natRouted` vApp network are allocated by the system from the pool associated with the vDC in which the vApp is deployed.

Organization vDC Networks

An organization vDC network allows virtual machines in the organization vDC to communicate with each other and to access other networks, including organization vDC networks and external networks, either directly or through an Edge Gateway that can provide firewall and NAT services.

- A direct organization vDC network connects directly to an external network. Only a system administrator can create a direct organization vDC network.
- A routed organization vDC network connects to an external network through an Edge Gateway, which is backed by a vShield Edge device. A routed organization vDC network also requires the containing vDC to include a network pool. After a system administrator has provisioned an organization vDC with an Edge Gateway and associated it with a network pool, organization administrator or system administrators can create routed organization vDC networks in that vDC.
- An isolated organization vDC network does not require an Edge Gateway or external network, but does require the containing vDC to be associated with a network pool. After a system administrator has created an organization vDC with a network pool, organization administrators or system administrators can create isolated organization vDC networks in that vDC.
- Most types of organization vDC networks do not provide any network services. Isolated organization vDC networks can specify a `DhcpPoolService`, which provides DHCP addresses from several pools of IP address ranges. All other services, such as NAT, firewall, and load balancing, are configured by a system administrator on the Edge Gateway to which the network connects.

Table 6-2. Types of Organization vDC Networks and Their Requirements

Organization vDC Network Connection	Description	Requirements
Direct connection to an external network.	Provides direct layer 2 connectivity to machines and networks outside of the organization vDC. Machines outside of this organization vDC can connect directly to machines within the organization vDC.	The cloud must contain an external network.
Routed connection to an external network.	Provides controlled access to machines and networks outside of the organization vDC via an Edge Gateway. System administrators and organization administrators can configure network address translation (NAT) and firewall settings on the gateway to make specific virtual machines in the vDC accessible from an external network.	The vDC must contain an Edge Gateway and a network pool.
No connection to an external network.	Provides an isolated, private network that machines in the organization vDC can connect to. This network provides no incoming or outgoing connectivity to machines outside this organization vDC.	The vDC must contain a network pool.

By default, only virtual machines in the organization vDC that contains the network can use it. When you create an organization vDC network, you can specify that it is shared. A shared organization vDC network can be used by all virtual machines in the organization.

Edge Gateways

An Edge Gateway provides a routed connection between an organization vDC network and an external network. It can provide any of the following services, defined in the `GatewayFeatures` element of the Edge Gateway's Configuration.

FirewallService	Specifies firewall rules that, when matched, block or allow incoming or outgoing network traffic. See “Firewall Service Configurations,” on page 158.
GatewayDhcpService	Provides DHCP services to virtual machines on the network. A variant of this service, <code>DhcpService</code> , is intended to provide DHCP services in vApp networks. See “Gateway DHCP Service Configurations,” on page 166.
GatewayIpsecVpnService	Defines one or more virtual private networks that connect an Edge Gateway to another network in or outside of the cloud.
LoadBalancerService	Distributes incoming requests across a set of servers. See “Load Balancer Service Configurations,” on page 163.
NatService	Provides network address translation services to computers on the network.
StaticRoutingService	Specifies static routes to other networks. See “Static Routing Service Configurations,” on page 161.

For an example of adding services to an Edge Gateway, see [“Configure Edge Gateway Services,”](#) on page 156. For more information about any of these services, see the *vShield Administration Guide*.

External Networks and Network Pools

External networks and network pools are vSphere resources backed by vSphere portgroup, VLAN, or DVswitch objects. A system administrator must create them, as described in [“Create an External Network,”](#) on page 219 and [“Create a Network Pool,”](#) on page 222. You must supply a reference to an external network when you create an Edge Gateway. When you create an organization vDC, you must supply a reference to a network pool if the vDC is to be able to contain routed or isolated networks. See [“Retrieve a List of External Networks and Network Pools,”](#) on page 204

Create an Edge Gateway

An Edge Gateway is a virtual router for organization vDC networks. You can configure it to provide network services such as DHCP, firewall, NAT, static routing, VPN, and load balancing.

You can create an Edge Gateway in either a compact or a full configuration. The full configuration provides increased capacity and performance. The compact configuration requires less memory and fewer compute resources. All services are supported in either configuration. You can enable either configuration for high availability, which enables automatic failover of the Edge Gateway to a backup instance that is running on a separate virtual machine.

An Edge Gateway can support up to ten interfaces. These interfaces are categorized as uplinks when they connect to an external network, and internal interfaces when they connect to an organization vDC network. You must specify at least one uplink interface when you create an Edge Gateway. All uplink interfaces on an Edge Gateway must connect to an external network available in the Provider vDC that backs the organization vDC in which you are creating the Edge Gateway. Internal interfaces are created automatically when you create a routed organization vDC network that connects to an Edge Gateway.

Prerequisites

- Verify that you are logged in to the vCloud API as a system administrator.
- An Edge Gateway requires an organization vDC backed by a Provider vDC that contains at least one external network.

Procedure

- 1 Choose an organization vDC to contain the Edge Gateway.

- 2 Choose an external network to use for the Edge Gateway's initial uplink interface.

This external network must be one of the networks listed in the `AvailableNetworks` element of the `ProviderVDC` that backs the organization vDC in which you are creating the Edge Gateway. Follow these steps to find a suitable external network.

- a Retrieve the XML representation of the organization vDC in which you are creating the Edge Gateway.

Use a request like this one:

```
GET https://vcloud.example.com/api/admin/vdc/44
```

The `ProviderVdcReference` element in the response contains a reference to the Provider vDC that backs this organization vDC.

```
<AdminVdc ...>
...
  <ProviderVdcReference
    type="application/vnd.vmware.admin.providervdc+xml"
    name="PvDC-Example"
    href="https://vcloud.example.com/api/admin/extension/providervdc/35"
  ...
</AdminVdc>
```

- b Retrieve the the XML representation of the Provider vDC.

Use a request like this one:

```
GET https://vcloud.example.com/api/admin/extension/providervdc/35
```

The `AvailableNetworks` element in the response lists the external networks that are available to that Provider vDC, and to all the organization vDCs that it supports.

```
<vmext:VMWProviderVdc ... >
...
  <vcloud:AvailableNetworks>
    <vcloud:Network
      type="application/vnd.vmware.admin.network+xml"
      name="VC1-VLAN48"
      href="https://vcloud.example.com/api/admin/network/297" />
    <vcloud:Network ... />
    <vcloud:Network ... />
  </vcloud:AvailableNetworks>
  ...
</vmext:VMWProviderVdc>
```

Choose an available external network to provide the initial interface for the new Edge Gateway. See [“Example: Create an Edge Gateway,”](#) on page 153 for more information about criteria for choosing an external network.

- 3 Create an `EdgeGateway` element.

In the `GatewayInterfaces` element, create a `GatewayInterface` element that defines an uplink interface.

- Specify `uplink` for the `InterfaceType`.
- Include the external network reference you retrieved in [Step 2](#) in the `Network` element.
- If you plan to create a NAT service or load balancer service in the Edge Gateway, you must create an IP sub-allocation for the uplink by including a `SubnetParticipation` element in the `GatewayInterface` element. IP addresses in the range you specify in this element are reserved for use by this Edge Gateway.

For information about additional elements that an EdgeGateway can contain, see “[Example: Create an Edge Gateway](#),” on page 153 and the schema reference.

- 4 POST the EdgeGateway element to the URL for adding Edge Gateways to the organization vDC.

The server takes the requested action and returns an XML representation of the partially-created object. This representation includes an href attribute, properties specified in the creation request, and an embedded Task element that tracks the creation of the object. When the task completes, the object has been created, and you can use the value of the href attribute with a GET request to retrieve the XML representation of the object.

See the response portion of “[Example: Create an Edge Gateway](#),” on page 153.

Example: Create an Edge Gateway

This example adds an Edge Gateway to the organization vDC created in “[Add a vDC to an Organization](#),” on page 140. The uplink interface specifies one of the networks shown in [Step 2b](#). For the purposes of this example, assume that the external network has a Configuration element that includes this information:

```
<Configuration>
  <IpScopes>
    <IpScope>
      ...
      <Gateway>10.147.115.190</Gateway>
      <Netmask>255.255.255.0</Netmask>
      ...
      <IpRanges>
        <IpRange>
          <StartAddress>10.147.115.129</StartAddress>
          <EndAddress>10.147.115.189</EndAddress>
        </IpRange>
      </IpRanges>
      <AllocatedIpAddresses>
        <IpAddress>10.147.115.129</IpAddress>
        <IpAddress>10.147.115.133</IpAddress>
        <IpAddress>10.147.115.132</IpAddress>
        <IpAddress>10.147.115.130</IpAddress>
      </AllocatedIpAddresses>
      ...
    </IpScope>
  </IpScopes>
  ...
</Configuration>
```

You can see the values from this external network's IpScope reflected in the SubnetParticipation element of the EdgeGateway. The address range specified in the IpRange element of this GatewayInterface must be within the IpRanges of the external network, and cannot include any IP addresses listed in the external network's AllocatedIpAddresses element. You can specify a contiguous IpRange, as we do in this example, or you can include multiple IpRange elements in the IpRanges if you need more IP addresses than are available in a contiguous block.

The external network specified in the GatewayInterface created in this example becomes the default route from this Edge Gateway (UseForDefaultRoute has a value of true). The default DNS service address is inherited from the network specified as the default route.

Request:

```
POST https://vcloud.example.com/api/admin/vdc/44/edgeGateways
Content-Type: application/vnd.vmware.admin.edgeGateway+xml
...
<?xml version="1.0" encoding="UTF-8"?>
<EdgeGateway
  name="theEdge"
  xmlns="http://www.vmware.com/vcloud/v1.5">
  <Description>Example Edge Gateway</Description>
  <Configuration>
    <GatewayBackingConfig>compact</GatewayBackingConfig>
    <GatewayInterfaces>
      <GatewayInterface>
        <Name>uplink1</Name>
        <DisplayName>uplink1</DisplayName>
        <Network
          href="https://vcloud.example.com/api/admin/network/297" />
        <InterfaceType>uplink</InterfaceType>
        <SubnetParticipation>
          <Gateway>10.147.115.190</Gateway>
          <Netmask>255.255.255.0</Netmask>
          <IpRanges>
            <IpRange>
              <StartAddress>10.147.115.155</StartAddress>
              <EndAddress>10.147.115.165</EndAddress>
            </IpRange>
          </IpRanges>
        </SubnetParticipation>
        <UseForDefaultRoute>true</UseForDefaultRoute>
      </GatewayInterface>
    </GatewayInterfaces>
    <HaEnabled>false</HaEnabled>
    <UseDefaultRouteForDnsRelay>false</UseDefaultRouteForDnsRelay>
  </Configuration>
</EdgeGateway>
```

The response is an EdgeGateway element with an embedded Task element that tracks the creation of the Edge Gateway object.

The response includes a number of Link elements that you can use to manage the new Edge Gateway. It also includes an EdgeGatewayServiceConfiguration element that contains a simple FirewallService, which drops all incoming and outgoing packets, effectively blocking all traffic through the Edge Gateway. This service is created by default if you do not specify an EdgeGatewayServiceConfiguration when you create the EdgeGateway. To remove or modify it, see [“Configure Edge Gateway Services,”](#) on page 156.

Response:

```
<?xml version="1.0" encoding="UTF-8"?>
<EdgeGateway
  xmlns="http://www.vmware.com/vcloud/v1.5"
  name="theEdge"
  id="urn:vcloud:gateway:2000"
  type="application/vnd.vmware.admin.edgeGateway+xml"
  href="https://vcloud.example.com/api/admin/edgeGateway/2000" ... >
  <Link
    rel="edit"
```

```

    type="application/vnd.vmware.admin.edgeGateway+xml"
    href="https://vcloud.example.com/api/admin/edgeGateway/2000" />
<Link
  rel="remove"
  href="https://vcloud.example.com/api/admin/edgeGateway/2000" />
<Link
  rel="up"
  type="application/vnd.vmware.admin.vdc+xml"
  href="https://vcloud.example.com/api/admin/vdc/44" />
<Link
  rel="edgeGateway:redeploy"
  href="https://vcloud.example.com/api/admin/edgeGateway/2000/action/redeploy" />
<Link
  rel="edgeGateway:configureServices"
  type="application/vnd.vmware.admin.edgeGatewayServiceConfiguration+xml"
  href="https://vcloud.example.com/api/admin/edgeGateway/2000/action/configureServices" />
<Link
  rel="edgeGateway:reapplyServices"
  href="https://vcloud.example.com/api/admin/edgeGateway/2000/action/reapplyServices" />
<Link
  rel="edgeGateway:syncSyslogSettings"
  href="https://vcloud.example.com/api/admin/edgeGateway/2000/action/syncSyslogServerSettings" />
<Link
  rel="edgeGateway:upgrade"
  href="https://vcloud.example.com/api/admin/edgeGateway/2000/action/upgradeConfig" />
...
<Tasks>
  <Task
    status="running"
    ...
    operation="Creating EdgeGateway theEdge(2000)"
    ...
  </Task>
</Tasks>
<Configuration>
  <GatewayBackingConfig>compact</GatewayBackingConfig>
  <GatewayInterfaces>
    <GatewayInterface>
      <Network
        type="application/vnd.vmware.admin.network+xml"
        name=""
        href="https://vcloud.example.com/api/admin/network/297" />
      <InterfaceType>uplink</InterfaceType>
      <ApplyRateLimit>false</ApplyRateLimit>
      <UseForDefaultRoute>true</UseForDefaultRoute>
    </GatewayInterface>
  </GatewayInterfaces>
  <EdgeGatewayServiceConfiguration>
    <FirewallService>
      <IsEnabled>true</IsEnabled>
      <DefaultAction>drop</DefaultAction>
      <LogDefaultAction>false</LogDefaultAction>
    </FirewallService>
  </EdgeGatewayServiceConfiguration>

```

```

    <HaEnabled>false</HaEnabled>
    <UseDefaultRouteForDnsRelay>false</UseDefaultRouteForDnsRelay>
  </Configuration>
</EdgeGateway>

```

Configure Edge Gateway Services

An organization administrator or system administrator can configure NAT, firewall, and similar services on an existing Edge Gateway by updating its `EdgeGatewayServiceConfiguration`.

The `Configuration` element of an `EdgeGateway` includes an `EdgeGatewayServiceConfiguration` element, which can contain definitions of any of the services listed in “[Edge Gateways](#),” on page 150. Details of service configurations vary, but the mechanism is the same for creating or updating any Edge Gateway service.

You can include an `EdgeGatewayServiceConfiguration` element that defines one or more services when you create an Edge Gateway, or you can create the Edge Gateway without this element, as shown in “[Create an Edge Gateway](#),” on page 151, then update it to add or change services as needed. Note that some services require a reference to one or more Edge Gateway interfaces, and cannot be configured until those interfaces exist.

Prerequisites

Verify that you are logged in to the vCloud API as an organization administrator or system administrator.

Procedure

- 1 Retrieve the XML representation of the Edge Gateway.
- 2 Examine the response to locate the `Link` element that contains the URL for configuring services on the Edge Gateway.

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

```

<Link
  href="https://vcloud.example.com/api/admin/edgeGateway/2000/action/configureServices"
  rel="add"
  type="application/vnd.vmware.admin.edgeGatewayServiceConfiguration+xml"/>

```

- 3 Copy the `EdgeGatewayServiceConfiguration` element from the `EdgeGateway` you retrieved in [Step 1](#).

The `configureServices` action replaces the entire contents of the existing `EdgeGatewayServiceConfiguration` with the one in the request body. Using the existing `EdgeGatewayServiceConfiguration` as the basis for your modifications reduces the chances of unintentional service changes.

- 4 Modify the `EdgeGatewayServiceConfiguration` that you copied in [Step 3](#) to add, remove, or change the services that this Edge Gateway offers.

An `EdgeGatewayServiceConfiguration` element can contain any of the following elements:

- `FirewallService`
- `GatewayDhcpService`
- `GatewayIpsecVpnService`
- `LoadBalancerService`
- `NatService`
- `StaticRoutingService`

- 5 POST the modified `EdgeGatewayServiceConfiguration` element to the URL in the value of the `href` attribute of the `configureServices` link described in [Step 2](#).

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

When the task completes successfully, the EdgeGatewayServiceConfiguration element you POSTed replaces the one you copied in [Step 3](#).

Example: Configure Services on an Edge Gateway

This example replaces the default firewall service on the Edge Gateway created in [“Create an Edge Gateway,”](#) on page 151. For details about this FirewallService, see [“Firewall Service Configurations,”](#) on page 158

Request:

POST <https://vcloud.example.com/api/admin/edgeGateway/2000/action/configureServices>

Content-Type: application/vnd.vmware.admin.edgeGatewayServiceConfiguration+xml

...

```
<?xml version="1.0" encoding="UTF-8"?>
<EdgeGatewayServiceConfiguration
  xmlns="http://www.vmware.com/vcloud/v1.5">
  <FirewallService>
    <IsEnabled>true</IsEnabled>
    <DefaultAction>allow</DefaultAction>
    <LogDefaultAction>>false</LogDefaultAction>
    <FirewallRule>
      <IsEnabled>true</IsEnabled>
      <Description>allow incoming ssh</Description>
      <Policy>allow</Policy>
      <Protocols>
        <Tcp>true</Tcp>
      </Protocols>
      <DestinationPortRange>22</DestinationPortRange>
      <DestinationIp>Internal</DestinationIp>
      <SourcePortRange>Any</SourcePortRange>
      <SourceIp>External</SourceIp>
      <EnableLogging>true</EnableLogging>
    </FirewallRule>
    <FirewallRule>
      <IsEnabled>true</IsEnabled>
      <Description>deny incoming telnet</Description>
      <Policy>drop</Policy>
      <Protocols>
        <Tcp>true</Tcp>
      </Protocols>
      <DestinationPortRange>23</DestinationPortRange>
      <DestinationIp>Internal</DestinationIp>
      <SourcePortRange>Any</SourcePortRange>
      <SourceIp>External</SourceIp>
      <EnableLogging>>false</EnableLogging>
    </FirewallRule>
  </FirewallService>
</EdgeGatewayServiceConfiguration>
```

Response:

```
202 Accepted
Content-Type: application/vnd.vmware.vcloud.task+xml
...
<Task
  href="https://vcloud.example.com/api/task/2120"
  ...
  status="running"
  operation="Updating services EdgeGateway theEdge(2000)"
  ... >
</Task>
```

Firewall Service Configurations

The default FirewallService in an EdgeGatewayServiceConfiguration is enabled and configured to block all incoming traffic. You can modify that FirewallService to allow incoming traffic, block outgoing traffic, or both.

A firewall service configuration includes several important parameters.

Firewall Rules

Each firewall rule specifies a protocol, IP address, and port. Packets that match the criteria in the rule are subject to an action defined in the Policy element of the rule. The action can 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 an EdgeGateway 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.

Port and Address Ranges

These elements in a FirewallRule specify source and destination IP ports and addresses to which the rule applies.

Example: Firewall Service Definition with Two Rules

This fragment of an EdgeGatewayServiceConfiguration defines a firewall service with two rules: one that allows incoming SSH connection, and one that denies incoming Telnet connections. These rules apply to any virtual machine that connects to a network backed by this Edge Gateway. 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.
SourcePortRange	Specify a source IP port or port range, or set to any to match any port.
DestinationPortRange	Specify a destination IP port or port range, or set to any to match any port.
SourceIp	Specify a source IP address, or use one of these strings.

Table 6-3. SourceIp and DestinationIp Values

Value	Result
Any	Matches any IP address
Internal	Matches any IP address originating on an organization vDC network connected to this EdgeGateway. When used in a vApp network, matches any IP address assigned to a virtual machine in the vApp.
External	Matches any IP address originating on an external network connected to this EdgeGateway. When used in a vApp network, matches any IP address except those assigned to a virtual machine in the vApp.

DestinationIp Specify a source IP address, or use one of the strings shown in [Table 6-3](#).

EnableLogging Set to true to log all packets that trigger this rule. See [“Firewall Rule Logging,”](#) on page 158.

Rules are applied to packets in the order in which the FirewallRule elements appear in the FirewallService definition.

NOTE The system assigns an Id value to each rule you create and uses these values when logging rule actions.

```
<FirewallService>
  <IsEnabled>true</IsEnabled>
  <DefaultAction>allow</DefaultAction>
  <LogDefaultAction>>false</LogDefaultAction>
  <FirewallRule>
    <IsEnabled>true</IsEnabled>
    <Description>allow incoming ssh</Description>
    <Policy>allow</Policy>
    <Protocols>
      <Tcp>true</Tcp>
    </Protocols>
    <DestinationPortRange>22</DestinationPortRange>
    <DestinationIp>Internal</DestinationIp>
    <SourcePortRange>Any</SourcePortRange>
    <SourceIp>External</SourceIp>
    <EnableLogging>>false</EnableLogging>
  </FirewallRule>
  <FirewallRule>
    <IsEnabled>true</IsEnabled>
    <Description>deny incoming telnet</Description>
    <Policy>drop</Policy>
    <Protocols>
      <Tcp>true</Tcp>
    </Protocols>
    <DestinationPortRange>23</DestinationPortRange>
    <DestinationIp>Internal</DestinationIp>
    <SourcePortRange>Any</SourcePortRange>
```

```

    <SourceIp>External</SourceIp>
    <EnableLogging>>false</EnableLogging>
  </FirewallRule>
</FirewallService>

```

You can see this fragment in the context of an Edge Gateway configuration in [“Example: Configure Services on an Edge Gateway,”](#) on page 157.

NAT Service Configurations

An Edge Gateway configuration can define a NAT (Network Address Translation) service that translates source or destination IP addresses and port numbers. In the most common case, you associate a NAT service with an uplink interface on an Edge Gateway so that addresses on organization vDC networks are not exposed on the external network.

A NAT service in an `EdgeGatewayServiceConfiguration` can include one or more rules, each of which is expressed in a `GatewayNatRule` element. Each rule translates the original IP address, port, or both, and applies to a network connected to the Edge Gateway. If the network is an uplink (to an external network), the network must include an IP sub-allocation pool.

There are two kinds of rules, as expressed in the value of the `RuleType` element:

SNAT	Source network address translation. This kind of rule translates the packet's source address and, optionally, source IP port to the values you specify.
DNAT	Destination network address translation. This kind of rule translates the packet's destination address and, optionally, destination IP port to the values you specify.

Example: NAT Service

The following fragment of an `EdgeGatewayServiceConfiguration` defines and enables a `NatService` that applies one destination NAT rule and one source NAT rule to the uplink interface defined in [“Example: Create an Edge Gateway,”](#) on page 153. In the DNAT rule, the `OriginalIp` and `OriginalPort` apply to the destination IP address and port of the packet being inspected. In the SNAT rule, the `OriginalIp` and `OriginalPort` apply to the source IP address and port of the packet being inspected. When you create a SNAT rule, you do not need to specify values for `TranslatedPort` and `OriginalPort`, which default to any.

NOTE The system assigns an `Id` value to each rule you create and uses these values when logging rule actions.

```

<?xml version="1.0" encoding="UTF-8"?>
<NatService>
  <IsEnabled>>true</IsEnabled>
  <NatRule>
    <RuleType>DNAT</RuleType>
    <IsEnabled>>true</IsEnabled>
    <GatewayNatRule>
      <Interface
        href="https://vcloud.example.com/api/admin/network/297" />
      <OriginalIp>10.147.115.155</OriginalIp>
      <OriginalPort>any</OriginalPort>
      <TranslatedIp>192.168.0.100-192.168.0.199</TranslatedIp>
      <TranslatedPort>any</TranslatedPort>
      <Protocol>any</Protocol>
      <IcmpSubType>any</IcmpSubType>
    </GatewayNatRule>
  </NatRule>
  <NatRule>

```



```

<RuleType>SNAT</RuleType>
<IsEnabled>>true</IsEnabled>
<GatewayNatRule>
  <Interface
    href="https://vcloud.example.com/api/admin/network/297" />
  <OriginalIp>192.168.0.10</OriginalIp>
  <TranslatedIp>10.147.115.155</TranslatedIp>
  <Protocol>any</Protocol>
</GatewayNatRule>
</NatRule>
</NatService>

```

To add this service to an Edge Gateway, include it in an `EdgeGatewayServiceConfiguration`. See [“Example: Configure Services on an Edge Gateway,”](#) on page 157.

Static Routing Service Configurations

An Edge Gateway or routed vApp network configuration can define a static routing service that specifies one or more static routes.

You can create static routes between two organization vDC networks, or between routed vApp 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 vApp network's `Configuration` or the `GatewayFeatures` element of an Edge Gateway's `Configuration`. A `StaticRoutingService` element can contain zero or more `StaticRoute` elements. Each `StaticRoute` specification requires the following elements.

Name	Name for the route.
Network	Network specification in CIDR notation.
NextHopIp	IP address of the next hop on the route. This address is typically the value in the <code>ExternalIp</code> element of the <code>RouterInfo</code> from the network to which this static route connects.
Interface	Specify <code>internal</code> if <code>NextHopIp</code> contains an IP address in the same network. Specify <code>external</code> if <code>NextHopIp</code> contains an IP address in a different network.
GatewayInterface	Used when configuring a static route in an organization vDC network. Contains a reference to the organization vDC network for which the static route is configured.

Example: Static Routes Between Organization vDC Networks

Assume two routed organization vDC networks, as defined in this fragment of an `EdgeGateway` element.

```

<GatewayInterface>
  <Name>vnic2</Name>
  <DisplayName>routed1</DisplayName>
  <Network
    type="application/vnd.vmware.admin.network+xml"
    name="routed1"
    href="https://vcloud.example.com/api/admin/network/29" />
  <InterfaceType>internal</InterfaceType>
  <SubnetParticipation>
    <Gateway>192.168.3.1</Gateway>
    <Netmask>255.255.255.0</Netmask>
    <IpAddress>192.168.3.1</IpAddress>
  </SubnetParticipation>
  <IpRanges />

```

```

    <UseForDefaultRoute>>false</UseForDefaultRoute>
  </GatewayInterface>
  <GatewayInterface>
    <Name>vnic3</Name>
    <DisplayName>routed2</DisplayName>
    <Network
      type="application/vnd.vmware.admin.network+xml"
      name="routed2"
      href="https://vcloud.example.com/api/admin/network/67" />
    <InterfaceType>internal</InterfaceType>
    <SubnetParticipation>
      <Gateway>172.168.0.1</Gateway>
      <Netmask>255.255.255.0</Netmask>
      <IpAddress>172.168.0.1</IpAddress>
    </SubnetParticipation>
    <IpRanges />
    <UseForDefaultRoute>>false</UseForDefaultRoute>
  </GatewayInterface>

```

- The organization vDC network named routed1 has Gateway address 192.168.3.1.
- The organization vDC network named routed2 has Gateway address 172.168.0.1.
- The Configuration of the vApp network in vApp1 has a RouterInfo element whose ExternalIp value is 192.168.2.100.
- The Configuration of the vApp network in vApp2 has a RouterInfo element whose ExternalIp value is 192.168.1.100.

A `StaticRoutingService` defined by the following fragment creates a static route to vApp1 from network routed1, and a static route to vApp2 from network routed2.

```

<StaticRoutingService>
  <IsEnabled>true</IsEnabled>
  <StaticRoute>
    <Name>TovApp1</Name>
    <Network>192.168.2.0/24</Network>
    <NextHopIp>192.168.3.10</NextHopIp>
    <Interface>Internal</Interface>
    <GatewayInterface
      type="application/vnd.vmware.vcloud.orgVdcNetwork+xml"
      name="routed1"
      href="https://vcloud.example.com/api/admin/network/29" />
    </StaticRoute>
    <StaticRoute>
    <Name>TovApp2</Name>
    <Network>192.168.1.0/24</Network>
    <NextHopIp>172.168.0.10</NextHopIp>
    <Interface>Internal</Interface>
    <GatewayInterface
      type="application/vnd.vmware.vcloud.orgVdcNetwork+xml"
      name="routed2"
      href="https://vcloud.example.com/api/admin/network/67" />
    </StaticRoute>
  </StaticRoutingService>

```

To add this service to an Edge Gateway, include it in an `EdgeGatewayServiceConfiguration`. See [“Example: Configure Services on an Edge Gateway,”](#) on page 157.

Static Routes Between vApp Networks

For an example of a static routing service in a vApp network, see [“Network Services in vApp Networks,”](#) on page 112.

Load Balancer Service Configurations

An Edge Gateway can provide load-balancing services that allow you to distribute incoming requests to a specific external IP address across multiple internal IP addresses. Several load-balancing algorithms are supported.

A load balancer service provides load balancing for TCP, HTTP, and HTTPS traffic. The load balancer accepts incoming IP requests on an external or internal interface, and uses the algorithm you specify to distribute requests across a pool of servers.

To add a load-balancer service to an Edge Gateway, include a `LoadBalancerService` element in the Edge Gateway's `EdgeGatewayServiceConfiguration`.

Example: Load Balancer Service

This fragment of an `EdgeGatewayServiceConfiguration` defines a `LoadBalancerService` that accepts incoming requests at external address `https://192.168.1.100` and balances them across two servers at internal addresses `10.200.100.10` and `10.200.100.11`. The following elements define a `LoadBalancerService`:

- A `Pool` that contains `ServicePort` and `Member` elements. A `LoadBalancerService` must include a `Pool` that defines a `ServicePort` for each protocol on which the load balancer handles incoming requests. You can define up to three `ServicePort` elements, one for each supported protocol (HTTP, HTTPS, TCP). This load balancer handles only SSL (HTTPS) requests, so it requires only one `ServicePort` element in its `Pool`.

You must specify one of the following load-balancing algorithms in the `Algorithm` element of the `ServicePort`.

IP_HASH	Selects a server based on a hash of the source and destination IP address of each packet.
LEAST_CONN	Distributes client requests to multiple servers based on the number of connections already on the server. New connections are sent to the server with the fewest connections.
ROUND_ROBIN	Each server is used in turn according to the weight assigned to it. This is the smoothest and fairest algorithm when the server's processing time remains equally distributed.
URI	The left part of the URI, before the question mark, is hashed and divided by the total weight of the running servers. The result designates which server receives the request, ensuring that a request is always directed to the same server as long as all servers remain available.

The `Pool` in this example also defines an optional `HealthCheck` element that specifies parameters used for periodic verification that all pool members are responding to requests.

Each `Member` element in the `Pool` specifies the `IpAddress` of a virtual machine that provides the service being requested. Incoming requests are balanced across all members of the pool. Because the `Algorithm` specified for this `Pool` is `ROUND_ROBIN`, each `Member` must be assigned a `Weight`.

- A `VirtualServer` element that defines the `Interface`, an internal or external interface defined by the containing `EdgeGateway`, on which requests are accepted.

NOTE The network referenced in the `Interface` element must be configured with an IP sub-allocation.

NOTE Each `Member` of a load balancer `Pool` can contain its own `ServicePort` element. If this element is present in a `Member`, its contents override the `ServicePort` element of the `Pool`.

For more information about `LoadBalancerService` elements and attributes, see the schema reference.

```
<LoadBalancerService>
  <IsEnabled>true</IsEnabled>
  <Pool>
    <Name>HTTPS_pool</Name>
    <ServicePort>
      <IsEnabled>true</IsEnabled>
      <Protocol>HTTPS</Protocol>
      <Algorithm>ROUND_ROBIN</Algorithm>
      <Port>443</Port>
      <HealthCheck>
        <Mode>TCP</Mode>
        <HealthThreshold>2</HealthThreshold>
        <UnhealthThreshold>3</UnhealthThreshold>
        <Interval>5</Interval>
        <Timeout>15</Timeout>
      </HealthCheck>
    </ServicePort>
    <Member>
      <IpAddress>10.200.100.10</IpAddress>
      <Weight>1</Weight>
    </Member>
    <Member>
      <IpAddress>10.200.100.11</IpAddress>
      <Weight>1</Weight>
    </Member>
  </Pool>
  <VirtualServer>
    <IsEnabled>true</IsEnabled>
    <Name>Example Virtual Server</Name>
    <Description>Incoming LoadBalancerService Requests</Description>
    <Interface
      type="application/vnd.vmware.vcloud.orgVdcNetwork+xml"
      href="https://vcloud.example.com/api/admin/network/199" />
    <IpAddress>192.168.1.100</IpAddress>
    <ServiceProfile>
      <IsEnabled>true</IsEnabled>
      <Protocol>HTTPS</Protocol>
      <Port>443</Port>
      <Persistence>
        <Method>SSL_SESSION_ID</Method>
      </Persistence>
    </ServiceProfile>
```

```

    <Logging>true</Logging>
    <Pool>HTTPS_pool</Pool>
  </VirtualServer>
</LoadBalancerService>

```

To add this service to an Edge Gateway, include it in an `EdgeGatewayServiceConfiguration`. See [“Example: Configure Services on an Edge Gateway,”](#) on page 157.

IPsec VPN Service Configurations

An Edge Gateway configuration can define an IPsec virtual private networking (VPN) service to provide secure virtual private networking within an organization, between organization vDC networks, or between an organization vDC network and an external IP address.

An `EdgeGateway` can contain zero or more `GatewayIpsecVpnService` elements, each of which defines VPN tunnels and endpoints.

Example: IPsec VPN Service in an Edge Gateway

```

<GatewayIpsecVpnService>
  <IsEnabled>true</IsEnabled>
  <Tunnel>
    <Name>Example VPN Tunnel</Name>
    <Description />
    <IpsecVpnLocalPeer>
      <Id>3786bb05-dc9a-471b-91cd-554499d45629</Id>
      <Name>gw02</Name>
    </IpsecVpnLocalPeer>
    <PeerIpAddress>10.147.46.68</PeerIpAddress>
    <PeerId>C64E127E-5E86-C57C-17ED-EB175A7A1811</PeerId>
    <LocalIpAddress>10.147.46.66</LocalIpAddress>
    <LocalId>6844BBB4-24E6-7A50-0F29-EB175A7AD899</LocalId>
    <LocalSubnet>
      <Name>nw01</Name>
      <Gateway>192.168.1.1</Gateway>
      <Netmask>255.255.255.0</Netmask>
    </LocalSubnet>
    <PeerSubnet>
      <Name>nw02</Name>
      <Gateway>192.168.2.1</Gateway>
      <Netmask>255.255.255.0</Netmask>
    </PeerSubnet>
    <SharedSecret>L3hithJa3zH7K4q2tH...</SharedSecret>
    <SharedSecretEncrypted>false</SharedSecretEncrypted>
    <EncryptionProtocol>AES256</EncryptionProtocol>
    <Mtu>1500</Mtu>
    <IsEnabled>true</IsEnabled>
  </Tunnel>
</GatewayIpsecVpnService>

```

To add this service to an Edge Gateway, include it in an `EdgeGatewayServiceConfiguration`. See [“Example: Configure Services on an Edge Gateway,”](#) on page 157.

Gateway DHCP Service Configurations

An Edge Gateway configuration can define a DHCP service that assigns IP addresses to clients on organization vDC networks. You can configure multiple address pools, each of which defines a range of IP addresses that are reserved for a specific network.

DHCP services for organization vDC networks are provided by the Edge Gateway to which those network connect. An EdgeGateway can contain an more GatewayDhcpService element. The service can define a pool of IP addresses for each organization vDC connected to the Edge Gateway.

NOTE To provide DHCP services on a vApp network, use the DhcpService element.

Example: Gateway DHCP Service

The following fragment of an EdgeGatewayServiceConfiguration defines and enables a GatewayDhcpService that defines two Pool objects, each of which allocates a range of IP addresses for a single organization vDC network.

```
<GatewayDhcpService>
  <IsEnabled>true</IsEnabled>
  <Pool>
    <IsEnabled>true</IsEnabled>
    <Network
      type="application/vnd.vmware.admin.orgVdcNetwork+xml"
      href="https://vcloud.example.com/api/admin/network/46" />
    <DefaultLeaseTime>3600</DefaultLeaseTime>
    <MaxLeaseTime>7200</MaxLeaseTime>
    <LowIpAddress>10.100.1.100</LowIpAddress>
    <HighIpAddress>10.100.1.150</HighIpAddress>
  </Pool>
  <Pool>
    <IsEnabled>true</IsEnabled>
    <Network
      type="application/vnd.vmware.admin.orgVdcNetwork+xml"
      href="https://vcloud.example.com/api/admin/network/47" />
    <DefaultLeaseTime>3600</DefaultLeaseTime>
    <MaxLeaseTime>7200</MaxLeaseTime>
    <LowIpAddress>10.100.1.200</LowIpAddress>
    <HighIpAddress>10.100.1.250</HighIpAddress>
  </Pool>
</GatewayDhcpService>
```

To add this service to an Edge Gateway, include it in an EdgeGatewayServiceConfiguration. See [“Example: Configure Services on an Edge Gateway,”](#) on page 157.

Create an Organization vDC Network

To add a network to an organization vDC, an administrator POSTs an OrgVdcNetwork element to the vDC's add URL for networks. An organization vDC network with a direct connection to an external network must be created by a system administrator. All other organization vDC network types can be created by either a system administrator or an organization administrator.

The contents of the Configuration element of the OrgVdcNetwork define the properties of the network, including its connections to other networks. See [“Create an Organization vDC Network With a Direct Connection,”](#) on page 167, [“Create an Organization vDC Network With a Routed Connection,”](#) on page 170, and [“Create an Isolated Organization vDC Network,”](#) on page 173.

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 148.

Prerequisites

Verify that you are logged in to the vCloud API as an organization administrator or system administrator.

Procedure

- 1 Retrieve the XML representation of the organization vDC to which you want to add the network.

This request retrieves the admin view of an organization vDC.

```
GET https://vcloud.example.com/api/admin/vdc/44
```

- 2 Examine the response to locate the `Link` element that contains the URL for adding networks to the vDC.

This element has a `rel` attribute value of `add` and a `type` attribute value of `application/vnd.vmware.admin.orgVdcNetwork+xml`, as shown here:

```
<Link
  rel="add"
  type="application/vnd.vmware.admin.orgVdcNetwork+xml"
  href="https://vcloud.example.com/api/admin/vdc/44/networks" />
```

- 3 Create an `OrgVdcNetwork` element.
- 4 POST the `OrgVdcNetwork` element to the URL described in [Step 2](#).

The server takes the requested action and returns a `Task` element that tracks the progress of the request. The `Owner` element of this task includes the `href` attribute of the new network. When the task completes, you can use the value of this attribute with a GET request to retrieve the XML representation of the new network.

Create an Organization vDC Network With a Direct Connection

An organization vDC network with a direct connection provides direct layer 2 connectivity to machines and networks outside of the organization vDC. Machines outside of this organization vDC can connect to machines within the organization vDC directly.

An organization vDC 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 to the vCloud API 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 150.

Procedure

- 1 Choose the external network to which this organization vDC network will connect.

This external network must be one of the ones listed in the `AvailableNetworks` element of the Provider vDC that backs the organization vDC in which you are creating the new network. Follow these steps to find a suitable external network.

- a Retrieve the XML representation of the organization vDC in which you are creating the new network.

Use a request like this one:

```
GET https://vcloud.example.com/api/admin/vdc/44
```

The `ProviderVdcReference` element in the response contains a reference to the Provider vDC that backs this organization vDC.

```
<AdminVdc ...>
  ...
  <ProviderVdcReference
    type="application/vnd.vmware.admin.providervdc+xml"
    name="PvDC-Example"
    href="https://vcloud.example.com/api/admin/extension/providervdc/35"
  ...
</AdminVdc>
```

- b Retrieve the the XML representation of the Provider vDC.

Use a request like this one:

```
GET https://vcloud.example.com/api/admin/extension/providervdc/35
```

The `AvailableNetworks` element in the response lists the external networks that are available to that Provider vDC, and to all the organization vDCs that it supports.

```
<vcloud:AvailableNetworks>
  <vcloud:Network
    type="application/vnd.vmware.admin.network+xml"
    name="VC1-VLAN48"
    href="https://vcloud.example.com/api/admin/network/297" />
  <vcloud:Network ... />
  <vcloud:Network ... />
</vcloud:AvailableNetworks>
```

- 2 Create an `OrgVdcNetwork` element.

- a Specify the `href` of the external network you chose in [Step 1](#) in the `ParentNetwork` element.

The `type` and `name` attributes are optional here.

- b Set the `FenceMode` to `bridged`.

This creates a direct connection to the parent network.

See the request portion of [“Example: Create an Organization vDC Network With a Direct Connection,”](#) on page 169.

- 3 POST the `OrgVdcNetwork` element you created in [Step 2](#) to the URL for adding networks to the organization vDC.

See the request portion of [“Example: Create an Organization vDC Network With a Direct Connection,”](#) on page 169.

The server takes the requested action and returns an XML representation of the partially-created object. This representation includes an href attribute, properties specified in the creation request, and an embedded Task element that tracks the creation of the object. When the task completes, the object has been created, and you can use the value of the href attribute with a GET request to retrieve the XML representation of the object.

See the response portion of [“Example: Create an Organization vDC Network With a Direct Connection,”](#) on page 169.

Example: Create an Organization vDC Network With a Direct Connection

This example adds a directly-connected network to the organization vDC created in [“Add a vDC to an Organization,”](#) on page 140. Because the network has a Configuration whose ParentNetwork element specifies an external network 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/vdc/44/networks
Content-Type: application/vnd.vmware.vcloud.orgVdcNetwork+xml ...
<?xml version="1.0" encoding="UTF-8"?>
<OrgVdcNetwork
  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/network/297" />
    <FenceMode>bridged</FenceMode>
  </Configuration>
</OrgVdcNetwork>
```

Response:

```
201 Created
Content-Type: application/vnd.vmware.vcloud.orgVdcNetwork+xml
...
<OrgVdcNetwork
  xmlns="http://www.vmware.com/vcloud/v1.5"
  name="Internet"
  type="application/vnd.vmware.vcloud.orgVdcNetwork+xml"
  href="https://vcloud.example.com/api/admin/network/54" ...>
  <Link
    rel="edit"
    type="application/vnd.vmware.vcloud.orgVdcNetwork+xml"
    href="https://vcloud.example.com/api/admin/network/54" />
  <Link
    rel="remove"
    href="https://vcloud.example.com/api/admin/network/54" />
  <Link
    rel="up"
    type="application/vnd.vmware.admin.vdc+xml"
    href="https://vcloud.example.com/api/admin/vdc/44" />
  <Link
    rel="down"
    type="application/vnd.vmware.vcloud.metadata+xml"
    href="https://vcloud.example.com/api/admin/network/54/metadata" />
  <Link
    rel="down"
```

```

    type="application/vnd.vmware.vcloud.allocatedNetworkAddress+xml"
    href="https://vcloud.example.com/api/admin/network/54/allocatedAddresses/" />
<Description>Bridged to the public Internet</Description>
<Tasks>
  <Task
    name="task"
    status="running"
    operation="Creating Network Internet(54)"
    ...
  </Task>
</Tasks>
<Configuration>
  ...
</Configuration>
</OrgVdcNetwork>

```

Create an Organization vDC Network With a Routed Connection

An organization vDC network with a routed connection provides controlled access to machines and networks outside of the organization vDC. System administrators and organization administrators can configure network address translation (NAT) and firewall settings on the network's Edge Gateway to make specific virtual machines in the vDC accessible from an external network.

Prerequisites

- Verify that you are logged in to the vCloud API as an organization administrator or system administrator.

Procedure

- 1 Retrieve the list of Edge Gateways in the organization vDC in which you plan to create the routed network.

You can use a query like this one, where *href* is the value of the *href* attribute of your organization vDC:

```
https://vcloud.example.com/api/query?type=edgeGateway&format=records&filter=vdc==href
```

If this organization vDC does not contain any Edge Gateways, or does not contain an Edge Gateway that has the configuration you want, a system administrator can create a new Edge Gateway. See [“Create an Edge Gateway,”](#) on page 151.

- 2 Choose an Edge Gateway that has interface capacity available.

An Edge Gateway can support a maximum of nine internal and external interfaces. At least one of those interfaces is typically consumed by a connection to an external network. Creation of a routed organization vDC network requires the Edge Gateway to have an unused interface available for the new network. To see how many interfaces each Edge Gateway in your organization vDC is using, you can run the query shown in [Step 1](#), then add the values of the *numberOfExtNetworks* and *numberOfOrgNetworks* attributes. If the total is less than 9, the Edge Gateway can accommodate a new routed organization vDC network.

- 3 Create an *OrgVdcNetwork* element.

- a Specify a value of *natRouted* in the *FenceMode* element of the network *Configuration*.

You can specify additional *Configuration* parameters, as noted in the schema reference.

- b Specify the *href* of the Edge Gateway you chose in [Step 1](#) in the *EdgeGateway* element.

See the request portion of [“Example: Create an Organization vDC Network With a Routed Connection,”](#) on page 171.

- 4 POST the `OrgVdcNetwork` element you created in [Step 2](#) to the URL for adding networks to the organization vDC.

See the request portion of [“Example: Create an Organization vDC Network With a Routed Connection,”](#) on page 171

The server takes the requested action and returns an XML representation of the partially-created object. This representation includes an `href` attribute, properties specified in the creation request, and an embedded Task element that tracks the creation of the object. When the task completes, the object has been created, and you can use the value of the `href` attribute with a GET request to retrieve the XML representation of the object.

See the response portion of [“Example: Create an Organization vDC Network With a Routed Connection,”](#) on page 171.

Example: Create an Organization vDC Network With a Routed Connection

This example adds a routed network to the organization vDC created in [“Add a vDC to an Organization,”](#) on page 140. The network connects through the Edge Gateway created in [“Create an Edge Gateway,”](#) on page 151. Because the creation request sets the value of the `IsShared` element to `true`, the new network is made available in all vDCs in this organization.

Request:

```
POST https://vcloud.example.com/api/admin/vdc/44/networks
Content-Type: application/vnd.vmware.vcloud.orgVdcNetwork+xml
...
<?xml version="1.0" encoding="UTF-8"?>
<OrgVdcNetwork
  name="Routed0vDCNet"
  xmlns="http://www.vmware.com/vcloud/v1.5">
  <Description>Routed through an Edge Gateway</Description>
  <Configuration>
    <IpScopes>
      <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>
    </IpScopes>
    <FenceMode>natRouted</FenceMode>
  </Configuration>
  <EdgeGateway
    href="https://vcloud.example.com/api/admin/gateway/2000" />
  <IsShared>true</IsShared>
</OrgVdcNetwork>
```

Response:

201 Created

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

...

```
<OrgVdcNetwork
  xmlns="http://www.vmware.com/vcloud/v1.5"
  name="RoutedOvDCNet"
  type="application/vnd.vmware.vcloud.orgVdcNetwork+xml"
  href="https://vcloud.example.com/api/admin/network/59" ...>
  <Link
    rel="edit"
    type="application/vnd.vmware.vcloud.orgVdcNetwork+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.vdc+xml"
    href="https://vcloud.example.com/api/admin/vdc/44" />
  <Link
    rel="down"
    type="application/vnd.vmware.vcloud.metadata+xml"
    href="https://vcloud.example.com/api/admin/network/59/metadata" />
  <Link
    rel="down"
    type="application/vnd.vmware.vcloud.allocatedNetworkAddress+xml"
    href="https://vcloud.example.com/api/admin/network/59/allocatedAddresses/" />
  <Description>Routed through an Edge Gateway</Description>
  <Tasks>
    <Task
      status="running"
      ...
      operation="Creating Network RoutedOvDCNet(59)"
      ...
      href="https://vcloud.example.com/api/task/999">
    </Task>
  </Tasks>
  <Configuration>
    ...
    <RetainNetInfoAcrossDeployments>false</RetainNetInfoAcrossDeployments>
  </Configuration>
  <EdgeGateway
    type="application/vnd.vmware.admin.edgeGateway+xml"
    name="theEdge"
    href="https://vcloud.example.com/api/admin/edgeGateway/2000" />
  <IsShared>true</IsShared>
</OrgVdcNetwork>
```

NOTE When the Task completes, the new network is represented in the EdgeGateway by a GatewayInterface whose InterfaceType is Internal. Unlike the Uplink interface that you create when you create an EdgeGateway, an internal interface cannot be created explicitly. It is created only as a side-effect of creating a routed organization vDC network.

Create an Isolated Organization vDC Network

An isolated organization vDC network provides an isolated, private network that machines in the organization vDC can connect to. This network provides no connectivity to machines outside this organization vDC. Machines outside of this organization vDC have no connectivity to machines within the organization.

Prerequisites

- Verify that you are logged in to the vCloud API as an organization administrator or system administrator.

Procedure

- 1 Create an OrgNetwork element.

See the request portion of [“Example: Create an Isolated Organization vDC Network,”](#) on page 173.

- 2 POST the OrgNetwork element you created in [Step 1](#) to the URL for adding networks to the organization vDC

See the request portion of [“Example: Create an Isolated Organization vDC Network,”](#) on page 173.

The server takes the requested action and returns an XML representation of the partially-created object. This representation includes an href attribute, properties specified in the creation request, and an embedded Task element that tracks the creation of the object. When the task completes, the object has been created, and you can use the value of the href attribute with a GET request to retrieve the XML representation of the object.

See the response portion of [“Example: Create an Isolated Organization vDC Network,”](#) on page 173.

Example: Create an Isolated Organization vDC Network

This example adds an isolated network to the organization vDC created in [“Add a vDC to an Organization,”](#) on page 140. It includes a ServiceConfig element that configures a DHCP service for the network. This type of DHCP service is identical to the DHCP service supported for a vApp network, and can specify only a single IP address range. No other network services can be created in an isolated organization vDC network.

Request:

```
POST https://vcloud.example.com/api/admin/vdc/44/networks
Content-Type: application/vnd.vmware.vcloud.orgVdcNetwork+xml
...
<?xml version="1.0" encoding="UTF-8"?>
<OrgVdcNetwork
  name="Isolated"
  xmlns="http://www.vmware.com/vcloud/v1.5">
  <Description>Isolated Organization vDC Network</Description>
  <Configuration>
    <IpScopes>
      <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>
    </IpScopes>
  </Configuration>
</OrgVdcNetwork>
```

```

    </IpScopes>
    <FenceMode>isolated</FenceMode>
  </Configuration>
  <ServiceConfig>
    <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>
  </ServiceConfig>
</OrgVdcNetwork>

```

Response:

201 Created

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

...

```

<OrgVdcNetwork
  xmlns="http://www.vmware.com/vcloud/v1.5"
  name="Isolated"
  type="application/vnd.vmware.vcloud.orgVdcNetwork+xml"
  href="https://vcloud.example.com/api/admin/network/60"...>
  <Link
    rel="edit"
    type="application/vnd.vmware.vcloud.orgVdcNetwork+xml"
    href="https://vcloud.example.com/api/admin/network/60" />
  <Link
    rel="remove"
    href="https://vcloud.example.com/api/admin/network/60" />
  <Link
    rel="up"
    type="application/vnd.vmware.admin.vdc+xml"
    href="https://vcloud.example.com/api/admin/vdc/44" />
  <Link
    rel="down"
    type="application/vnd.vmware.vcloud.metadata+xml"
    href="https://vcloud.example.com/api/admin/network/60/metadata" />
  <Link
    rel="down"
    type="application/vnd.vmware.vcloud.allocatedNetworkAddress+xml"
    href="https://vcloud.example.com/api/admin/network/60/allocatedAddresses/" />
  <Description>Isolated Organization Vdc Network</Description>
  <Tasks>
    <Task
      name="task"
      status="running"
      operation="Creating Network Isolated(60)"
      ...
    </Task>
  </Tasks>
  <Configuration>
    ...

```

```

</Configuration>
<ServiceConfig>
  <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>
</ServiceConfig>
</OrgVdcNetwork>

```

Synchronize Syslog Server Settings for an Edge Gateway or vApp 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 each Edge Gateway or vApp network that includes a `FirewallService` for which logging is enabled.

If a system administrator changes the `SyslogServerSettings` for a cloud, all Edge Gateways 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 Edge Gateway, you must be an organization administrator 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 vApp network or Edge Gateway.
- 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/edgeGateway/2000/action/syncSyslogServerSettings"
```

- 3 Make a POST request to the `href` value of 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 an Edge Gateway

This request synchronizes the syslog server settings for the Edge Gateway created in [“Create an Edge Gateway,”](#) on page 151.

```
POST https://vcloud.example.com/api/admin/edgeGateway/2000/action/syncSyslogServerSettings
```

Response:

```
202 Accepted
Content-Type: application/vnd.vmware.vcloud.task+xml
...
<Task ... >
  ...
</Task>
```

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 191. 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 79.

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 177.

- 4 POST the AdminCatalog element to the organization's add URL for catalogs.

See the request portion of [“Example: Create a Catalog,”](#) on page 177.

The server creates an empty catalog. See the response portion of [“Example: Create a Catalog,”](#) on page 177.

Example: Create a Catalog

This example adds a catalog to the organization created in [“Example: Create an Organization,”](#) on page 133.

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 contains information extracted from 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 73.

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 to the vCloud API 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 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 or SAML-based identity provider. Groups must be imported. 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. Users and groups can also be imported from an external identity provider that supports SAML (the Security Assertion Markup Language). Local user accounts are stored in the vCloud Director database and managed by the organization administrator. Imported user accounts are managed by the service from which the user was imported. If an imported user changes his password, contact information, or other account properties, those changes are not effective in vCloud Director until the user is imported again.

About Identity Providers

In vCloud Director, an identity provider is a service that accepts credentials such as a user name and password and authenticates the user as a member of a group or organization. vCloud Director recognizes two kinds of identity providers:

Integrated	The integrated identity provider is a service provided by vCloud Director. It can authenticate users who are created locally or imported from LDAP.
SAML	An organization can define a SAML identity provider that can be used as part of a federated identity strategy. Such a strategy can enable an enterprise to provide access to multiple, unrelated services, including vCloud Director, with a single set of credentials. This sort of authentication strategy is often referred to as "single sign-on." See "Retrieve or Update Organization Settings," on page 136.

The XML representation of a User can include an IdentityProvider element that specifies either INTEGRATED or SAML. If the element is missing or empty, a value of INTEGRATED is assumed

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 180.

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 49. For more information about creating new roles, see ["Create a Role,"](#) on page 189.

Prerequisites

Verify that you are logged in to the vCloud API 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 180.
- 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 133. 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`, whose values are inherited from the organization.

- The `Password` element, which must not be empty when you create a `User`, is never returned.
- The `ProviderType`, which was not specified in the request, defaults to `INTEGRATED`. See [“About Identity Providers,”](#) on page 180.

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>
  <ProviderType>INTEGRATED</ProviderType>
  <IsAlertEnabled>false</IsAlertEnabled>
  <IsDefaultCached>false</IsDefaultCached>
  <IsGroupRole>false</IsGroupRole>
  <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 183. You can also import users individually.

Prerequisites

- Verify that you are logged in to the vCloud API 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 133. 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>
  <ProviderType>INTEGRATED</ProviderType>
  <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 182.

Prerequisites

- Verify that you are logged in to the vCloud API 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 133. 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>

```

Import a User or Group from a SAML Identity Provider

If your organization defines a SAML identity provider in its `OrgFederationSettings`, you cannot import the users or groups as you can from an LDAP service. Instead, you must map the SAML-defined roles of those users and groups to roles defined in your organization.

Unlike imports from an LDAP service, imports from a SAML identity provider do not actually import information from an external database. Instead, the operation creates a mapping between a user or group name in your organization's database and a user or group name defined by your organization's SAML provider. The vCloud Director database stores these mappings, but does not store any data retrieved from the SAML provider.

When a user login presents a SAML token to the organization, user and group names in the token are evaluated using the mappings established by the import operation. This evaluation process can be summarized as follows:

- If the SAML token includes an attribute named `UserName`, try to match the value of that attribute to the value of the `name` attribute of the `User`.
- If the SAML token does not include an attribute named `UserName`, try to match the value of the `NameId` element to the value of the `name` attribute of the `User`.
- If the SAML token includes an attribute named `Groups`, assume that the value of that attribute is a list of group names, and try to match each value in the list to the value of the `name` attribute of a `Group` in the organization. If the
- If the SAML token does not include an attribute named `Groups`, assume that the user is not a member of any group.

Prerequisites

- Verify that you are logged in to the vCloud API as an organization administrator or system administrator.
- Verify that your organization has defined a SAML identity provider in its `OrgFederationSettings`.

Procedure

- 1 Create a `User` or `Group` element that identifies a user or group defined by your organization's SAML provider.
- 2 Include the following line in the `User` or `Group` element.


```
<ProviderType>saml</ProviderType>
```
- 3 POST the element to the organization's users or groups URL.

Example: Import a User from a SAML Identity Provider

This example is identical to the one shown in [“Example: Import a User from an LDAP Database,”](#) on page 182, but includes a `ProviderType` element that specifies the source as the organization's SAML identity provider. It also omits the `IsExternal` element, which is required when importing from LDAP but is ignored when importing from SAML.

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>
  <ProviderType>saml</ProviderType>
  <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>
  <ProviderType>saml</ProviderType>
  <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>
```

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.

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.

NOTE You can create and modify rights associated with extension services, but not those associated with vCloud Director. See [“Create a Service-Specific Right,”](#) on page 300

Predefined Roles and Their Rights

vCloud Director includes predefined roles. Each of these roles includes a set of default rights.

A system administrator can create new roles and modify existing roles, except the System Administrator role.

Table 6-4. 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		

Table 6-4. Default Rights for the Predefined Roles (Continued)

	System Administrator	Organization Administrator	Catalog Author	vApp Author	vApp User	Console Access Only
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		
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				

Table 6-4. Default Rights for the Predefined Roles (Continued)

	System Administrator	Organization Administrator	Catalog Author	vApp Author	vApp User	Console Access Only
Organization: Edit SMTP Settings	X	X				
Organization: Edit Quotas Policy	X	X				
Organization: View Organizations	X	X				
Organization: Edit Organization vDC Network Properties	X	X				
Organization: View Organization vDC Networks	X	X				
Organization: Edit Leases Policy	X	X				
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 to the vCloud API 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 190.

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 133. 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/action/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,”](#) on page 21.

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 79.

Managing and Monitoring a Cloud

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 vSphere resources, 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 132.

This chapter includes the following topics:

- [“Summary of vSphere Platform Extension Requests,”](#) on page 193
- [“Retrieve or Update System Settings,”](#) on page 197
- [“Attach a vCenter Server,”](#) on page 198
- [“Finding Available vCenter Resources,”](#) on page 200
- [“Create a Provider vDC,”](#) on page 209
- [“Create an External Network,”](#) on page 219
- [“Create a Network Pool,”](#) on page 222
- [“Import a Virtual Machine from vCenter,”](#) on page 228
- [“Relocate a Virtual Machine to a Different Datastore,”](#) on page 231
- [“Truststore and Keytab Maintenance,”](#) on page 233
- [“Retrieve the vSphere URL of an Object,”](#) on page 236

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.

IMPORTANT Request URLs are always available in Link elements contained by the representation of the object on which they operate. URL forms shown here are for reference purposes only. Although URLs have a well-known syntax and a well-understood interpretation, a client should treat vCloud API request URLs as opaque strings. The rules that govern how the server constructs these strings might change in future releases.

This summary may not cover all requests in this category. For the complete list of requests, along with detailed information about input and output types, see the Operations lists in the schema reference.

Table 7-1. Summary of vSphere Platform Extension Requests

Operation	Request	Request Body	Response
Retrieve a list of Provider vDCs in the system.	GET <i>API-URL/admin/extension/providerVdcReferences</i>	None	VMWProviderVdcReferences
Retrieve a list of external networks in the system.	GET <i>API-URL/admin/extension/externalNetworkReferences</i>	None	VMWExternalNetworkReferences
Retrieve a list of network pools in the system.	GET <i>API-URL/admin/extension/networkPoolReferences</i>	None	VMWNetworkPoolReferences
Retrieve a list of vCenter servers registered to vCloud Director.	GET <i>API-URL/admin/extension/vimServerReferences</i>	None	VMWVimServerReferences
Retrieve information about a vCenter server.	GET <i>API-URL/admin/extension/vimServer/id</i>	None	VimServer
Retrieve a list of available resource pools on a vCenter server.	GET <i>API-URL/admin/extension/vimServer/id/resourcePoolList</i>	None	ResourcePoolList
Update vCenter server settings.	PUT <i>API-URL/admin/extension/vimServer/id</i>	VimServer	Task
Register a vCenter server and vShield manager.	POST <i>API-URL/admin/extension/action/registervimserver</i>	RegisterVimServerParams	RegisterVimServerParams
Unregister a vCenter server and vShield manager.	POST <i>API-URL/admin/extension/action/unregister</i>	None	Task
Force reconnection to a vCenter server.	POST <i>API-URL/admin/extension/vimServer/id/action/forcevimserverreconnect</i>	None	Task
Retrieve a list of ESX/ESXi hosts in the system.	GET <i>API-URL/admin/extension/hostReferences</i>	None	VMWHostReferences
Retrieve information about a host.	GET <i>API-URL/admin/extension/host/id</i>	None	Host
Prepare a host.	POST <i>API-URL/admin/extension/host/id/action/prepare</i>	PrepareHostParams	Task

Table 7-1. Summary of vSphere Platform Extension Requests (Continued)

Operation	Request	Request Body	Response
Unprepare a host.	POST <i>API-URL/admin/extension/host/id/action/unprepare</i>	None	Task
Enable a host.	POST <i>API-URL/admin/extension/host/id/action/enable</i>	None	Task
Disable a host.	POST <i>API-URL/admin/extension/host/id/action/disable</i>	None	Task
Repair a host.	POST <i>API-URL/admin/extension/host/id/action/repair</i>	None	204 No Content
Upgrade a host.	POST <i>API-URL/admin/extension/host/id/action/upgrade</i>	None	Task
Create a Provider vDC. [NEW]	POST <i>API-URL/admin/extension/providerdcsparms</i>	VMWProviderVdcParams	VMWProviderVdc
Create a Provider vDC. [DEPRECATED]	POST <i>API-URL/admin/extension/providerdcs</i>	VMWProviderVdc	VMWProviderVdc
Retrieve a Provider vDC.	GET <i>API-URL/admin/extension/providerdc/id</i>	None	VMWProviderVdc
Update a Provider vDC.	PUT <i>API-URL/admin/extension/providerdc/id</i>	VMWProviderVdc	VMWProviderVdc
Enable a Provider vDC.	POST <i>API-URL/admin/extension/providerdc/id/action/enable</i>	None	204 No Content
Disable a Provider vDC.	POST <i>API-URL/admin/extension/providerdc/id/action/disable</i>	None	204 No Content
Delete a Provider vDC.	DELETE <i>API-URL/admin/extension/providerdc/id</i>	None	Task
Create an external network.	POST <i>API-URL/admin/extension/externalnets</i>	VMWExternalNetwork	VMWExternalNetwork
Retrieve an external network.	GET <i>API-URL/admin/extension/externalnet/id</i>	None	VMWExternalNetwork
Update an external network.	PUT <i>API-URL/admin/extension/externalnet/id</i>	VMWExternalNetwork	VMWExternalNetwork
Delete an external network.	DELETE <i>API-URL/admin/extension/externalnet/id</i>	None	Task
Create a network pool.	POST <i>API-URL/admin/extension/networkpools</i>	VMWNetworkPool	VMWNetworkPool

Table 7-1. Summary of vSphere Platform Extension Requests (Continued)

Operation	Request	Request Body	Response
Retrieve a network pool.	GET <i>API-URL/admin/extension/networkPool/id</i>	None	VMWNetworkPool
Update a network pool.	PUT <i>API-URL/admin/extension/networkPool/id</i>	VMWNetworkPool	VMWNetworkPool
Delete a network pool.	DELETE <i>API-URL/admin/extension/networkPool/id</i>	None	Task
Import a virtual machine from vCenter as a vApp.	POST <i>API-URL/admin/extension/vimServer/id/importVmAsVapp</i>	ImportVmAsVAppParams	VApp
Import a virtual machine from vCenter as a vApp template.	POST <i>API-URL/admin/extension/vimServer/id/importVmAsVappTemplate</i>	ImportVmAsVAppTemplateParams	VAppTemplate
Retrieve the representation of a datastore.	GET <i>API-URL/admin/extension/datastore/id</i>	None	Datastore
Enable a datastore.	POST <i>API-URL/admin/extension/datastore/id/action/enable</i>	None	Datastore
Disable a datastore.	POST <i>API-URL/admin/extension/datastore/id/action/disable</i>	None	Datastore
Update a datastore.	PUT <i>API-URL/admin/extension/datastore/id</i>	Datastore	Datastore
Delete a datastore.	DELETE <i>API-URL/admin/extension/datastore/id</i>	None	204 No Content
Import a virtual machine from vCenter into an existing vApp.	POST <i>API-URL/admin/extension/vimServer/id/importVmIntoExistingVApp</i>	ImportVmIntoExistingVAppParams	Task
Import a media image from vCenter.	POST <i>API-URL/admin/extension/vimServer/id/importMedia</i>	ImportMediaParams	Media
Enable a resource pool.	POST <i>API-URL/admin/extension/resourcePool/id/action/enable</i>	None	204 No Content
Disable a resource pool.	POST <i>API-URL/admin/extension/resourcePool/id/action/disable</i>	None	204 No Content
Place a vApp in maintenance mode to prevent users from changing vApp metadata.	POST <i>API-URL/vApp/id/action/enterMaintenanceMode</i>	None	204 No Content

Table 7-1. Summary of vSphere Platform Extension Requests (Continued)

Operation	Request	Request Body	Response
Remove a vApp from maintenance mode.	POST <i>API-URL/vApp/id/action/exitMaintenanceMode</i>	None	204 No Content
Relocate a virtual machine in a vApp template to a different datastore. [DEPRECATED]	POST <i>API-URL/vAppTemplate/vm-id/action/relocate</i>	RelocateParams	Task
Relocate a virtual machine to a different datastore. [DEPRECATED]	POST <i>API-URL/vApp/vm-id/action/relocate</i>	RelocateParams	Task
Migrate a virtual machines to a different resource pool. [NEW]	POST <i>API-URL/admin/extension/resourcePool/id/action/migrateVms</i>	MigrateParams	Task
Retrieve a URL that you can use to access a vSphere object with a vSphere client. [NEW]	GET <i>API-URL/admin/extension/vimServer/id/vimObjType/vimObjMoref/vSphereWebClientUrl</i>	None	VSphereWebClientUrl
Merge Provider vDCs. [NEW]	PUT <i>API-URL/admin/extension/provider/vdc/id/action/merge</i>	None	204 No Content
Retrieve the list of storage profiles that exist on a vCenter server. [NEW]	GET <i>API-URL/admin/extension/vimServer/id/storageProfiles</i>	None	VMWStorageProfiles
Refresh the list of storage profiles that exist on a vCenter server. [NEW]	POST <i>API-URL/admin/extension/vimServer/id/action/refreshStorageProfiles</i>	None	Task
Register vCloud Director so that it appears as an extension in the vSphere Client Solutions Manager tab. [NEW]	POST <i>API-URL/admin/extension/vimServer/id/action/refresh</i>	None	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 for specific operations.
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 to the vCloud API 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 to the vCloud API 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 199.

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:VmServer>
    <vmext:ShieldManager
      ...
    </vmext:ShieldManager>
  </vmext:RegisterVmServerParams>

```

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 `VmServerReference` 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 to the vCloud API 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 `VmServerReference` 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 `VmServer` 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.

Table 7-2. vCenter Resource Lists

List URL	List Contents
<code>https://vcloud.example.com/api/admin/extension/vimServer/id/vmsList</code>	References to virtual machines in this vCenter server's inventory
<code>https://vcloud.example.com/api/admin/extension/vimServer/id/hostReferences</code>	References to ESX/ESXi hosts in this vCenter server's inventory
<code>https://vcloud.example.com/api/admin/extension/vimServer/id/resourcePoolList</code>	ResourcePool objects
<code>https://vcloud.example.com/api/admin/extension/vimServer/id/networks</code>	VimObjectRef elements of type DV_PORTGROUP and NETWORK
<code>https://vcloud.example.com/api/admin/extension/vimServer/id/storageProfiles</code>	VMWStorageProfile objects.

Example: Resources on a vCenter Server

Request:

```
GET https://vcloud.example.com/api/admin/extension/vimServer/9
```


Response:

```

200 OK
...
<vmext:VmServer
  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" />
  <vcloud:Link
    rel="down"
    type="application/vnd.vmware.admin.vmwStorageProfiles+xml"
    href="https://vcloud.example.com/api/admin/extension/vimServer/9/storageProfiles" />
  ...
  <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>10.115.124.1</vmext:ShieldManagerHost>
  <vmext:ShieldManagerUserName>admin</vmext:ShieldManagerUserName>
  <vmext:ShieldManagerUserName>admin</vmext:ShieldManagerUserName>
  <vmext:Uuid>44D5DAAA-7F3E-456D-B1CB-8288D7308AD6</vmext:Uuid>
  <vmext:VcProxy>cell1</vmext:VcProxy>
  <vmext:VcVersion>5.0.0</vmext:VcVersion>
  <vmext:UseVsphereService>false</vmext:UseVsphereService>
</vmext:VmServer>

```

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 VmServer 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 200.

Prerequisites

- Verify that you are logged in to the vCloud API as a system administrator.
- Retrieve the XML representation of a vCenter server registered to your cloud. See [“Finding Available vCenter Resources,”](#) on page 200.

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 202. 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:VmObjectType>DATASTORE</vmext:VmObjectType>
    </vmext:VmObjectRef>
</vmext:DataStoreRefs>
</vmext:ResourcePool>
<vmext:ResourcePool
  name="cluster3">
  <vmext:MoRef>resgroup-230</vmext:MoRef>
  <vmext:VmObjectType>RESOURCE_POOL</vmext:VmObjectType>
  <vmext:DataStoreRefs>
    <vmext:VmObjectRef>
      <vmext:VmServerRef
        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:VmObjectType>DATASTORE</vmext:VmObjectType>
    </vmext:VmObjectRef>
  </vmext:DataStoreRefs>
</vmext:ResourcePool>
  ...
</vmext:ResourcePoolList>

```

Retrieve a List of Available Portgroups and Switches from a vCenter Server

You can retrieve the list of available 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`, `DV_PORTGROUP`, and `NETWORK` objects on the server. Objects that have already been consumed in the creation of an external network are not listed. See [“Finding Available vCenter Resources,”](#) on page 200.

Prerequisites

Verify that you are logged in to the vCloud API 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 Available Portgroups and Switches from a vCenter Server,”](#) on page 203. If the list is empty, all network resources on the server are already in use.

Example: Retrieve a List of Available Portgroups and Switches 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 External Networks and Network Pools

You can retrieve a list of external networks and network pools that have been created on a vCenter server registered to a cloud.

A reference to an external network is required when you create an Edge Gateway. A reference to a network pool is usually required when you create an organization vDC. These resources do not exist in a new vCloud Director installation. A system administrator must create them, as described in [“Create an External Network,”](#) on page 219 and [“Create a Network Pool,”](#) on page 222.

Procedure

- 1 Retrieve the XML representation of your cloud.

Use a request like this one:

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

- 2 Examine the response, a `VMWExtension` element, to locate the links to lists of external networks and network pools.

These links have the following form:

```
<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" />
```

- 3 Make a GET request to the link that represents the object type of interest.

See [“Example: Retrieve a List of External Networks,”](#) on page 205.

Example: Retrieve a List of External Networks

Request:

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

Each reference to an external network includes its `type`, `name`, and `href` attributes, as shown in this example.

Response:

```
<vmext:VMWExternalNetworkReferences ... >
...
<vmext:ExternalNetworkReference
  type="application/vnd.vmware.admin.extension.network+xml"
  name="VC0"
  href="https://vcloud.example.com/api/admin/extension/externalnet/85" />
<vmext:ExternalNetworkReference
  type="application/vnd.vmware.admin.extension.network+xml"
  name="VC1"
  href="https://vcloud.example.com/api/admin/extension/externalnet/302" />
...
</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.

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 `vmList` 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 200.

Prerequisites

Verify that you are logged in to the vCloud API 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 206. 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:VmObjectType>VIRTUAL_MACHINE</vmext:VmObjectType>
  </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:VmObjectType>VIRTUAL_MACHINE</vmext:VmObjectType>
  </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:VmObjectType>VIRTUAL_MACHINE</vmext:VmObjectType>
  </vmext:VmObjectRef>
</vmext:VmObjectRefsList>
```

Retrieve a List of Storage Profiles from a vCenter Server

You can retrieve the list of storage profiles that have been defined on a vCenter server registered to a cloud. To retrieve the list, you make a GET request to the server's `storageProfiles` link.

Storage profiles are named configurations of vCenter storage. When you create a Provider vDC, you must specify the name of at least one storage profile to provide storage capacity for that Provider vDC.

The `VMWStorageProfiles` of a `VimServer` element contains an entry for every storage profile that has been defined on the server. See [“Finding Available vCenter Resources,”](#) on page 200.

Prerequisites

- Verify that you are logged in to the vCloud API as a system administrator.
- Retrieve the XML representation of a vCenter server registered to your cloud. See [“Finding Available vCenter Resources,”](#) on page 200.

Procedure

- 1 (Optional) Refresh the cached list of storage profiles.

vCloud Director maintains a cache of the storage profiles that have been created on each of its registered vCenter servers. The cache is refreshed on a regular schedule, and is likely to contain an up-to-date list. If a vCenter administrator has recently created a new storage profile, you can force it to be added to the cache by following these steps.

- a Examine the `VimServer` element to locate its `action/refreshStorageProfiles` link.

The link has the following form:

```
<vcloud:Link
    rel="refreshStorageProfiles"

    href="https://vcloud.example.com/api/admin/extension/vimServer/9/action/refreshStorageProfiles" />
```

- b Refresh the cached list of storage profiles from the vCenter server.

Use a request like this one.

```
POST
https://vcloud.example.com/api/admin/extension/vimServer/9/action/refreshStorageProfiles
```

The response is a `Task`. When the task completes, the cached list of storage profiles on the vCenter server has been updated.

- 2 Examine the `VimServer` element to locate its `storageProfiles` link.

The link has the following form:

```
<vcloud:Link
    rel="down"
    type="application/vnd.vmware.admin.storageProfiles+xml"
    href="https://vcloud.example.com/api/admin/extension/vimServer/9/storageProfiles" />
```

- 3 Retrieve the list of storage profiles.

See [“Example: Retrieve a List of Storage Profiles from a vCenter Server,”](#) on page 208.

Example: Retrieve a List of Storage Profiles from a vCenter Server

Request:

GET https://vcloud.example.com/api/admin/extension/vimServer/9/storageProfiles

This response shows that the specified vCenter server contains three storage profiles, named Gold, Silver, and Bronze. Storage profile details, including datastore identifiers and capacities, are included in the response, but only the value of the name attribute of a VMWStorageProfile is needed when adding that storage profile to a Provider vDC.

Response:

200 OK

Content-Type: application/vnd.vmware.admin.vmwstorageprofiles+xml

...

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

<vmext:VMWStorageProfiles

xmlns:vmext="http://www.vmware.com/vcloud/extension/v1.5"

xmlns:vcloud="http://www.vmware.com/vcloud/v1.5"

...>

...

<vmext:VMWStorageProfile

name="Gold">

<vmext:MoRef>storageProfile-CFAA6D92-36FC-4C16-9B30-FAC79B902371</vmext:MoRef>

<vmext:VmObjectType>STORAGE_PROFILE</vmext:VmObjectType>

<vmext:FreeStorageMb>169203.0</vmext:FreeStorageMb>

<vmext:TotalStorageMb>325120.0</vmext:TotalStorageMb>

<vmext:DataStoreRefs>

<vmext:VmObjectRef>

<vmext:VmServerRef

type="application/vnd.vmware.admin.vmwvirtualcenter+xml"

name=""

href="https://vcloud.example.com/api/admin/extension/vimServer/9" />

<vmext:MoRef>datastore-44</vmext:MoRef>

<vmext:VmObjectType>DATASTORE</vmext:VmObjectType>

</vmext:VmObjectRef>

<vmext:VmObjectRef>

<vmext:VmServerRef

type="application/vnd.vmware.admin.vmwvirtualcenter+xml"

name=""

href="https://vcloud.example.com/api/admin/extension/vimServer/9" />

<vmext:MoRef>datastore-45</vmext:MoRef>

<vmext:VmObjectType>DATASTORE</vmext:VmObjectType>

</vmext:VmObjectRef>

</vmext:DataStoreRefs>

</vmext:VMWStorageProfile>

<vmext:VMWStorageProfile

name="Silver">

...

</vmext:VMWStorageProfile>

<vmext:VMWStorageProfile

name="Bronze">

...

</vmext:VMWStorageProfile>

</vmext:VMWStorageProfiles>

Create a Provider vDC

A Provider vDC is a collection of compute, memory, and storage resources from one vCenter. A Provider vDC provides resources to organization vDCs.

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, storage profiles, and resource pools, or change other properties such as its description. A system administrator cannot change the primary resource pool or vCenter server that was specified when the Provider vDC was created. An organization administrator can retrieve a read-only representation of a Provider vDC in a `ProviderVdc` element. The `ProviderVdc` element includes a subset of the elements and attributes that are visible in the corresponding `VMWProviderVdc`.

Prerequisites

- Verify that you are logged in to the vCloud API as a system administrator.
- Choose a vCenter server to supply a resource pool and storage profiles to this Provider vDC. See [“Finding Available vCenter Resources,”](#) on page 200.

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.createProviderVdcParams+xml`, as shown here:

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

- 3 Create a `VMWProviderVdcParams` element that specifies the properties of the Provider vDC.
 - a Include a `VimServer` element that references the vCenter server that you have chosen to supply a resource pool and storage profiles to this Provider vDC.
 - b Include a `ResourcePoolRefs` element that specifies one resource pool.

The `ResourcePoolRef` must contain the `href` attribute value of the `VimServer` element you created in [Step 3a](#), as well as the `MoRef` and `VimObjectType` values of the resource pool as they appear in the `ResourcePool` and element from the resource pool list. See the request portion of [“Example: Create a Provider vDC,”](#) on page 210.

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.

- c Include at least one `StorageProfile` element that contains the name of a storage profile that has been defined on the vCenter server referenced in the `VimServer` element you created in [Step 3a](#).
- 4 POST the `VMWProviderVdc` element you created in [Step 3](#) to the URL described in [Step 2](#).
See the request portion of [“Example: Create a Provider vDC,”](#) on page 210.

The server creates and enables the Provider vDC and returns a `VMWProviderVdc` element that includes information derived from 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.
- Each storage profile you specified becomes the basis for a `ProviderVdcStorageProfile` object, and can be retrieved from the Provider vDC after it has been created, or by using a `providerVdcStorageProfile` query.
- A `VxlanPoolType` network pool is created on the vCenter server referenced by the `VimServer` element you created in [Step 3a](#) and implicitly attached to the new Provider vDC. You can create other types of network pools on that vCenter server if you want them to be available in the Provider vDC. See [“Create a Network Pool,”](#) on page 222.

Example: Create a Provider vDC

This request creates a Provider vDC specifying a resource pool extracted from the response portion of [“Example: Retrieve a List of Resource Pools from a vCenter Server,”](#) on page 202 and a storage profile extracted from [“Example: Retrieve a List of Storage Profiles from a vCenter Server,”](#) on page 208. The vCenter server that provides the resources is referenced in the `VimServerRef` and `VimServer` elements.

Request:

```
POST https://vcloud.example.com/api/admin/extension/providerVdcParams
Content-Type: application/vnd.vmware.admin.createProviderVdcParams+xml
...
<?xml version="1.0" encoding="UTF-8"?>
<vmext:VMWProviderVdcParams
  xmlns:vmext="http://www.vmware.com/vcloud/extension/v1.5"
  xmlns:vcloud="http://www.vmware.com/vcloud/v1.5"
  name="PvDC-Example">
  <vcloud:Description>Example Provider vDC</vcloud:Description>
  <vmext:ResourcePoolRefs>
    <vmext:VimObjectRef>
      <vmext:VimServerRef
        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
    href="https://vcloud.example.com/api/admin/extension/vimServer/9" />
  <vmext:StorageProfile>Gold</vmext:StorageProfile>
</vmext:VMWProviderVdcParams>
```

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` and `SupportedHardwareVersions`, and a list of `HostReferences` identifying the ESX hosts that provide the resources.

Response:

```
201 Created
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:vccloud="http://www.vmware.com/vccloud/v1.5"
status="0"
name="PvDC-Example"
id="urn:vccloud:providervdc:35"
type="application/vnd.vmware.admin.vmwprovidervdc+xml" ... >
<vccloud:Link
  rel="up"
  type="application/vnd.vmware.admin.vmwProviderVdcReferences+xml"
  href="https://vccloud.example.com/api/admin/extension/providerVdcReferences" />
<vccloud:Link
  rel="alternate"
  type="application/vnd.vmware.admin.providervdc+xml"
  href="https://vccloud.example.com/api/admin/providervdc/35" />
<vccloud:Link
  rel="update:resourcePools"
  type="application/vnd.vmware.admin.resourcePoolSetUpdateParams+xml"
href="https://vccloud.example.com/api/admin/extension/providervdc/35/action/updateResourcePools"
/>
  <vccloud:Link
    rel="down"
    type="application/vnd.vmware.admin.vmwProviderVdcResourcePool+xml"
    href="https://vccloud.example.com/api/admin/extension/providervdc/35/resourcePools" />
  <vccloud:Link
    rel="down"
    type="application/vnd.vmware.admin.vmwStorageProfiles+xml"
href="https://vccloud.example.com/api/admin/extension/providervdc/35/availableStorageProfiles" />
  <vccloud:Link
    rel="down"
    type="application/vnd.vmware.admin.vdcReferences+xml"
    href="https://vccloud.example.com/api/admin/providervdc/35/vdcReferences" />
  <vccloud:Description>Example Provider vDC</vccloud:Description>
  <vccloud:Tasks>
    <vccloud:Task
      ...
      operation="Creating Provider Virtual Datacenter PvDC-Example(35)">
      ...
    </vccloud:Task>
  </vccloud:Tasks>
  <vccloud:ComputeCapacity>
    <vccloud:Cpu>
      <vccloud:Units>MHZ</vccloud:Units>
      <vccloud:Allocation>0</vccloud:Allocation>
      <vccloud:Total>0</vccloud:Total>
      <vccloud:Used>0</vccloud:Used>
      <vccloud:Overhead>0</vccloud:Overhead>
    </vccloud:Cpu>
    <vccloud:Memory>
      <vccloud:Units>MB</vccloud:Units>
      <vccloud:Allocation>0</vccloud:Allocation>
      <vccloud:Total>0</vccloud:Total>
      <vccloud:Used>0</vccloud:Used>
      <vccloud:Overhead>0</vccloud:Overhead>
    </vccloud:Memory>
  </vccloud:ComputeCapacity>

```

```

</vcloud:ComputeCapacity>
<AvailableNetworks>
  <Network
    type="application/vnd.vmware.admin.network+xml"
    name="VC1-VLAN48"
    href="https://vcloud.example.com/api/admin/network/297" />
</AvailableNetworks>
<StorageProfiles>
  <ProviderVdcStorageProfile
    type="application/vnd.vmware.admin.pvdcStorageProfile+xml"
    name="Gold"
    href="https://vcloud.example.com/api/admin/pvdcStorageProfile/101" />
</StorageProfiles>
<Capabilities>
  <SupportedHardwareVersions>
    <SupportedHardwareVersion>vmx-04</SupportedHardwareVersion>
    <SupportedHardwareVersion>vmx-07</SupportedHardwareVersion>
    <SupportedHardwareVersion>vmx-08</SupportedHardwareVersion>
  </SupportedHardwareVersions>
</Capabilities>
<IsEnabled>true</IsEnabled>
<vcloud:NetworkPoolReferences>
  <vcloud:NetworkPoolReference
    type="application/vnd.vmware.admin.networkPool+xml"
    name="VXLAN01"
    href="https://vcloud.example.com/api/admin/extension/networkPool/69c0a96e-1151-439d-
b8f6-2e2a11785c9f" />
</vcloud:NetworkPoolReferences>
<vmext:DataStoreRefs />
<vmext:ResourcePoolRefs>
  <vmext:VimObjectRef>
    <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:VimObjectRef>
</vmext:ResourcePoolRefs>
<vmext:VimServer
  type="application/vnd.vmware.admin.vmwvirtualcenter+xml"
  name="ConfigWizard Configured vCenter"
  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>
</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 to the vCloud API 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 213.

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 210. 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
...
<?xml version="1.0" encoding="UTF-8"?>
<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
    primary="true">
    <vcloud:Link
      rel="migrateVms"
```

```

href="https://vcloud.example.com/api/admin/extension/resourcePool/64/action/migrateVms" />
  <vcloud:Link
    rel="resourcePoolVmList"
    href="https://vcloud.example.com/api/admin/extension/resourcePool/64/vmList" />
  <vmext:ResourcePoolVmObjectRef>
    <vmext:VmServerRef
      type="application/vnd.vmware.admin.vmwvirtualcenter+xml"
      name="ConfigWizard Configured vCenter"
      href="https://vcloud.example.com/api/admin/extension/vimServer/9" />
    <vmext:MoRef>resgroup-235</vmext:MoRef>
    <vmext:VmObjectType>RESOURCE_POOL</vmext:VmObjectType>
  </vmext:ResourcePoolVmObjectRef>
  <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:VMWProviderVdcResourcePool>
</vmext:VMWProviderVdcResourcePoolSet>

```

Update Provider vDC Resource Pools

A system administrator can update the resource pool set of an existing Provider vDC to add or remove resource pools. Adding resource pools allows organization vDCs that reference the Provider vDC to consume additional resources during periods of high demand. Removing resource pools frees the underlying resources.

When you create a Provider vDC, it initially contains a single resource pool, called the primary resource pool. Adding secondary resource pools allows a Provider vDC to support resource elasticity in organization vDCs that use the AllocationPool or AllocationVApp (pay as you go) allocation model. Resource elasticity allows an organization vDC's compute resources to grow or shrink on demand.

All of a Provider vDC's resource pools must come from the same vCenter. See [“Retrieve a List of Resource Pools from a vCenter Server,”](#) on page 201.

Prerequisites

Verify that you are logged in to the vCloud API 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/updateResourcePools"/>

```

- 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 Update the resource pool set.

To add resource pools:

- a Choose another resource pool from the same vCenter server.
- b Create an `UpdateResourcePoolSetParams` element that contains an `AddItem` element for each resource pool to add.

To remove resource pools:

- a Examine the resource pool list and find the pool to remove.
- b Verify the pool is not the primary resource pool, and that no virtual machines are using it.
If necessary, use the `action/migrateVms` link to migrate virtual machines to another resource pool.
- c Create an `UpdateResourcePoolSetParams` element that contains a `DeleteItem` element for each resource pool to remove.

- 5 POST the `UpdateResourcePoolSetParams` element to the Provider vDC's `resourcePools` link.

Example: Update Provider vDC Resource Pools

This request adds a resource pool to the Provider vDC created in [“Example: Create a Provider vDC,”](#) on page 210. 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 201 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:VmServerRef
      type="application/vnd.vmware.admin.vmwvirtualcenter+xml"
      href="https://vcloud.example.com/api/admin/extension/vimServer/9" />
    <vmext:MoRef>resgroup-230</vmext:MoRef>
    <vmext:VmObjectType>RESOURCE_POOL</vmext:VmObjectType>
  </vmext:AddItem>
</vmext:UpdateResourcePoolSetParams>
```

Response:

```
202 Accepted
Content-Type: application/vnd.vmware.vcloud.task+xml
...
<Task ... >
  ...
</Task>
```

This request removes one of the two resource pools (a secondary resource pool) shown in [“Retrieve a Provider vDC Resource Pool Set,”](#) on page 213. 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>
```

Update Provider vDC Storage Profiles

A system administrator can update the storage profiles that are included in a Provider vDC. New storage profiles can be added, and unused storage profiles can be removed.

Storage profiles can be created by a vCenter administrator on any vCenter server that supports the profile-driven storage feature. A Provider vDC can provide access to any of the storage profiles that have been created on its vCenter server (the one referenced in its `vmext:VimServer` element). A vCloud Director system administrator must specify at least one vCenter storage profile when creating a Provider vDC, and can add or remove storage profiles later as needed. Organization vDCs reference Provider vDC storage profiles in much the same way that Provider vDCs reference vCenter storage profiles. Media and Disk objects, as well as vApps and virtual machines reference organization vDC storage profiles by name.

Prerequisites

Verify that you are logged in to the vCloud API 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 Examine the `VMWProviderVdc` element to find the `storageProfiles` link and the `vmext:VimServer` element.

The `storageProfiles` link has the following form:

```
<Link
  rel="edit"
  type="application/vnd.vmware.admin.updateProviderVdcStorageProfiles+xml"
  href="https://vcloud.example.com/api/admin/providervdc/35/storageProfiles" />
```


- 3 Create an `UpdateProviderVdcStorageProfiles` request body that specifies the details of the update.

To add a storage profile:

- a Choose another storage profile from the vCenter server referenced in the `vmext:VimServer` element you located in [Step 2](#).
- b Create an `UpdateProviderVdcStorageProfiles` element that contains an `AddStorageProfile` element for each storage profile to add.

To remove a storage profile:

- a Retrieve the Provider vDC's `availableStorageProfiles` list and find the name of the profile to remove.

Use a request like this one:

```
GET
https://vcloud.example.com/api/admin/extension/providervdc/35/availableStorageProfiles
```

- b Verify that no organization vDCs are using the storage profile you want to remove.
- c Create an `UpdateProviderVdcStorageProfiles` element that contains a `RemoveStorageProfile` element for each storage profile to remove.

- 4 POST the `UpdateProviderVdcStorageProfiles` element to the Provider vDC's `storageProfiles` link.

The server returns a `Task` element that you can use to track the progress of the update. When the update is complete, the Provider vDC includes the updated set of storage profiles in its `StorageProfiles` element. Each storage profile you added becomes the basis for a `ProviderVdcStorageProfile` object, and can be retrieved from the Provider vDC after it has been created, or by using a `providerVdcStorageProfile` query.

Example: Update Provider vDC Storage Profiles

This request adds a storage profile named `Bronze` to the Provider vDC created in [“Example: Create a Provider vDC,”](#) on page 210. The new storage profile is hosted on the same vCenter server that hosts the existing storage profile. See [“Retrieve a List of Storage Profiles from a vCenter Server,”](#) on page 207 for an example that lists the storage profiles available on that server.

Request:

```
POST https://vcloud.example.com/api/admin/extension/providervdc/35/action/updateStorageProfiles
Content-Type: application/vnd.vmware.admin.updateProviderVdcStorageProfiles+xml
```

```
...
<?xml version="1.0" encoding="UTF-8"?>
<vmext:UpdateProviderVdcStorageProfiles
  xmlns:vmext="http://www.vmware.com/vcloud/extension/v1.5"
  xmlns:vcloud="http://www.vmware.com/vcloud/v1.5">
  <vmext:AddStorageProfile>Bronze</vmext:AddStorageProfile>
</vmext:UpdateProviderVdcStorageProfiles>
```

Response:

```
202 Accepted
Content-Type: application/vnd.vmware.vcloud.task+xml
```

```
...
<Task ... >
  ...
</Task>
```

Merge Provider vDCs

You can merge one or more Provider vDCs with an existing Provider vDC. The merged Provider vDC contains the union of all resources from the contributing Provider vDCs

In earlier releases of vCloud Director, a Provider vDC could be backed by no more than one resource pool. This release removes that limitation, and allows you to merge existing Provider vDCs to create a single Provider vDC that is backed by multiple resource pools. When you merge Provider vDCs, you select one or more Provider vDCs as contributors and one Provider vDC as the target of the merge. When the merge is complete:

- The target Provider vDC includes the networks, network pools, storage profiles, resource pools, and datastores from all of the contributors.
- Organization vDCs that were backed by the contributors are now backed by the target.

Prerequisites

Verify that you are logged in to the vCloud API as a system administrator.

Procedure

- 1 Identify the Provider vDCs to merge.

Select contributors that are not backed by the same resource pool as the target. You can use a query like this one to discover the resource pools backing each of your Provider vDCs.

```
GET https://vcloud.example.com/api/query?type=providerVdcResourcePoolRelation&format=records
```
- 2 Construct a ProviderVdcMergeParams element that references each of the contributor provider vDCs.

See the request portion of [“Example: Merge Provider vDCs,”](#) on page 218.
- 3 Make a POST request to the action/merge link of the target Provider vDC and supply the ProviderVdcMergeParams as the request body.

Example: Merge Provider vDCs

This request includes a ProviderVdcReference for each of two contributors. The request URL specifies the target Provider vDC.

Request:

```
POST https://vcloud.example.com/api/admin/extension/providervdc/35/action/merge
Content-Type: application/vnd.vmware.admin.providerVdcMergeParams+xml
...
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<vmext:ProviderVdcMergeParams
  type="application/vnd.vmware.admin.mergeParams+xml"
  xmlns:vmext="http://www.vmware.com/vcloud/extension/v1.5"
  xmlns:vcloud="http://www.vmware.com/vcloud/v1.5">
  <vmext:ProviderVdcReference
    type="application/vnd.vmware.admin.vmwprovidervdc+xml"
    name="PvDC-VC001"
    href="https://vcloud.example.com/api/admin/extension/providervdc/46" />
  <vmext:ProviderVdcReference
```

```

    type="application/vnd.vmware.admin.vmwproviderdc+xml"
    name="PvDC-VC005"
    href="https://vcloud.example.com/api/admin/extension/providerdc/45" />
</vmext:ProviderVdcMergeParams>

```

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 a read-only representation of an external network to examine its properties.

Prerequisites

- Verify that you are logged in to the vCloud API as a system administrator.
- Retrieve the list of available portgroups. See [“Retrieve a List of Available Portgroups and Switches from a vCenter Server,”](#) on page 203.

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 220.

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 Available Portgroups and Switches from a vCenter Server,”](#) on page 203.

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">
  <vcloud:Description>ExternalNet</vcloud:Description>
  <vcloud:Configuration>
    <vcloud:IpScopes>
      <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>example.com</vcloud:DnsSuffix>
      </vcloud:IpScope>
    </vcloud:IpScopes>
    <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 referenced by organization vDCs provide support for isolated and routed vApp networks and organization vDC networks. To create a network pool, a system administrator must specify backing network resources on a vCenter server registered to the cloud.

A network pool object represents a collection of vSphere network resources that are contained by a Provider vDC and available to be associated with organization vDCs backed by that Provider vDC. All network pools created on a vCenter server are available in each Provider vDC that references the server. Traffic on each network in a pool is isolated at layer 2 from all other networks.

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. After a network pool has been associated with an organization vDC (typically when the vDC is created), network resources from the pool are consumed as needed to create isolated or routed organization vDC networks or vApp networks in the vDC.

NOTE When you create a Provider vDC, a `VxlanPoolType` network pool is created automatically on the vCenter server that backs the Provider vDC. This pool is given a name derived from the name of the containing Provider vDC and attached to it at creation. You cannot delete or modify this network pool. You cannot create a `vVxlanPoolType` network pool by any other method. If you rename a Provider vDC, its `VxlanPoolType` network pool is automatically renamed.

vSphere VXLAN networks are based on the IETF draft VXLAN standard. These networks support local-domain isolation equivalent to what is supported by vSphere isolation-backed networks. In addition, they provide:

- logical networks spanning layer 3 boundaries
 - logical networks spanning multiple racks on a single layer 2
 - broadcast containment
 - higher performance
 - greater scale (up to 16 million network addresses)
-

All network pools are defined by a `VMWNetworkPool` element. This element can have one of several 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 223, [“Create an Isolation-Backed Network Pool,”](#) on page 225, and [“Create a Portgroup-Backed Network Pool,”](#) on page 227.

Prerequisites

Verify that you are logged in to the vCloud API 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 to the vCloud API as a system administrator.
- Verify that you know your cloud's add URL for `networkPools`. See [“Create a Network Pool,”](#) on page 222.
- Verify that at least one vCenter server attached to your cloud has network resources available. See [“Retrieve a List of Available Portgroups and Switches from a vCenter Server,”](#) on page 203

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 portion of [“Example: Create a VLAN-Backed Network Pool,”](#) on page 223.
- 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 223.

Example: Create a VLAN-Backed Network Pool

Use the query service to retrieve a list of DV Switch objects available on vCenter servers registered to this cloud.

```
https://vcloud.example.com/api/query?type=dvSwitch&format=records
```

The query response includes the values you'll need for the `VimServerRef` and `MoRef` elements. The `VimObjectType` for a DV Switch is always `DV_SWITCH`.

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. Before creating an isolation-backed network pool, consider using the VXLAN pool that is created automatically when you create a Provider vDC.

vSphere VXLAN networks are based on the IETF draft VXLAN standard. These networks support local-domain isolation equivalent to what is supported by vSphere isolation-backed networks. In addition, they provide:

- logical networks spanning layer 3 boundaries
- logical networks spanning multiple racks on a single layer 2
- broadcast containment
- higher performance
- greater scale (up to 16 million network addresses)

Prerequisites

- Verify that you are logged in to the vCloud API as a system administrator.
- Verify that you know your cloud's add URL for networkPools. See [“Create a Network Pool,”](#) on page 222.
- Verify that at least one vCenter server attached to your cloud has network resources available. See [“Retrieve a List of Available Portgroups and Switches from a vCenter Server,”](#) on page 203

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 225.

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 to the vCloud API as a system administrator.
- Verify that you know your cloud's add URL for networkPools. See [“Create a Network Pool,”](#) on page 222.
- Verify that at least one vCenter server attached to your cloud has network resources available. See [“Retrieve a List of Available Portgroups and Switches from a vCenter Server,”](#) on page 203

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 227.
- 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 222.
See the request portion of [“Example: Create a Portgroup-Backed Network Pool,”](#) on page 227.

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 to the vCloud API as a system administrator.
- Identify the virtual machine to import. See [“Retrieve a List of Virtual Machines from a vCenter Server,”](#) on page 205.

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 to the vCloud API as a system administrator.
- Identify the virtual machine to import. See [“Retrieve a List of Virtual Machines from a vCenter Server,”](#) on page 205.

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 206. 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 to the vCloud API as a system administrator.
- Identify the virtual machine to import. See [“Retrieve a List of Virtual Machines from a vCenter Server,”](#) on page 205.

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 206 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.

NOTE Beginning with this release of the vCloud API, the `action/relocate` operation in a `Vm` is deprecated. If the datastore that contains a virtual machine has been disabled by the system administrator or is no longer associated with virtual machine's designated storage profile, you must update the `Vm` element that represents the virtual machine. That update revalidates the storage profile and relocates the virtual machine if necessary. See [“Update the Storage Profile for a Virtual Machine,”](#) on page 126.

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 to the vCloud API 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:VmObjectType>DATASTORE</vmext:VmObjectType>
    </vmext:VmObjectRef>
</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 233.

- 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 233.

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>

```

Truststore and Keytab Maintenance

You can use the vCloud API to upload and manage SSL certificates, keystores, and Kerberos keytabs for your organization's LDAP service.

The OrgLdapSettings element includes links that you can use to manage the organization's LDAP truststore and keystore by uploading new certificates and keytabs.

```

<Link
    rel="certificate:update"
    type="application/vnd.vmware.admin.certificateUpdateParams+xml"

href="https://vcloud.example.com/api/admin/org/26/settings/ldap/action/updateLdapCertificate" />
<Link
    rel="certificate:reset"

href="https://vcloud.example.com/api/admin/org/26/settings/ldap/action/resetLdapCertificate" />
<Link
    rel="keystore:update"

```

```

    type="application/vnd.vmware.admin.keystoreUpdateParams+xml"
    href="https://vcloud.example.com/api/admin/org/26/settings/ldap/action/updateLdapKeyStore" />
<Link
  rel="keystore:reset"
  href="https://vcloud.example.com/api/admin/org/26/settings/ldap/action/resetLdapKeyStore" />
<Link
  rel="keytab:update"
  type="application/vnd.vmware.admin.sspiKeytabUpdateParams+xml"

href="https://vcloud.example.com/api/admin/org/26/settings/ldap/action/updateLdapSspiKeytab" />
<Link
  rel="keytab:reset"
  href="https://vcloud.example.com/api/admin/org/26/settings/ldap/action/resetLdapSspiKeytab" />

```

Similar links contained by the SystemSettings element allow the system administrator to manage the system LDAP truststore and keystore.

```

<vcloud:Link
  rel="certificate:update"
  type="application/vnd.vmware.admin.certificateUpdateParams+xml"

href="https://vcloud.example.com/api/admin/extension/settings/ldapSettings/action/updateLdapCertificate" />
<vcloud:Link
  rel="certificate:reset"

href="https://vcloud.example.com/api/admin/extension/settings/ldapSettings/action/resetLdapCertificate" />
<vcloud:Link
  rel="keystore:update"
  type="application/vnd.vmware.admin.keystoreUpdateParams+xml"

href="https://vcloud.example.com/api/admin/extension/settings/ldapSettings/action/updateLdapKeyStore" />
<vcloud:Link
  rel="keystore:reset"

href="https://vcloud.example.com/api/admin/extension/settings/ldapSettings/action/resetLdapKeyStore" />
<vcloud:Link
  rel="keytab:update"
  type="application/vnd.vmware.admin.sspiKeytabUpdateParams+xml"

href="https://vcloud.example.com/api/admin/extension/settings/ldapSettings/action/updateLdapSspiKeytab" />
<vcloud:Link
  rel="keytab:reset"

href="https://vcloud.example.com/api/admin/extension/settings/ldapSettings/action/resetLdapSspiKeytab" />

```

All of these links implement similar operations. They either upload a new certificate, keytab, or keystore, or reset or remove an existing one. vCloud Director imposes limits on upload sizes.

Table 7-3. Truststore, Certificate, and Keytab Upload Limits

Upload Type	Maximum Size in Megabytes
vCenter truststore	10
LDAP certificate	2
LDAP keystore	2
LDAP SSPI keytab	2
AMQP certificate	2
AMQP truststore	10

Prerequisites

Verify that you are logged in to the vCloud API as a system administrator.

Procedure

- 1 Create the request body that the request requires.
For details on these request bodies, see the schema reference.
- 2 POST the request body to the request URL.
The response includes an `uploadLocation` parameter whose value is a URL to which you can upload the certificate, keytab, or keystore.
- 3 Use a PUT request to upload the certificate, keytab, or keystore.

Example: Upload an SSL Certificate for an Organization LDAP Service

This example uploads an SSL certificate whose size is 2048 bytes. The first step obtains an upload URL by POSTing a `CertificateUpdateParams` element to the organization's `settings/ldap/action/updateLdapCertificate` URL.

Request:

```
POST: https://vcloud.example.com/api/admin/org/26/settings/ldap/action/updateLdapCertificate
Content-type: application/vnd.vmware.admin.certificateUpdateParams+xml
...
<?xml version="1.0" encoding="UTF-8"?>
<CertificateUpdateParams
  fileSize="2048"
  xmlns="http://www.vmware.com/vcloud/v1.5">
</CertificateUpdateParams>
```

The response contains an `uploadLocation` parameter whose value is a URL to which you can upload the certificate.

Response:

```
<CertificateUploadSocket
  xmlns="http://www.vmware.com/vcloud/v1.5"
  uploadLocation="https://vcloud.example.com/transfer/53bc1/ldapCertificate">
  <Task
    ...
    status="running"
    operation="Updating LDAP certificate Organization (7)"
    ...
  </Task>
</CertificateUploadSocket>
```

To upload the certificate, make a PUT request to the `uploadLocation` URL and supply the certificate in the request body.

Request:

```
PUT https://vcloud.example.com/transfer/53bc1/ldapCertificate
Content-length: 2048
...serialized contents of certificate...
```

EOF

Response:

200 OK

Retrieve the vSphere URL of an Object

If you know the `VimObjectType` and `MoRef` of an object represented in the vCloud API, you can use that information to retrieve a URL that you can use to access the object with the vSphere Web Client.

Using the vSphere Web Client to examine an object vCloud Director uses can help a system administrator diagnose problems with resource consumption and allocation. To retrieve the vSphere URL of an object, you must construct a request URL in the following format.

```
API-URL/admin/extension/vimServer/id/vimObjType/vimObjMoref/vSphereWebClientUrl
```

- *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. The `vimServer` object that has this *id* must be the one that hosts the object that *vimObjType* and *vimObjMoref* identify.
- *vimObjType* is the vSphere object type, expressed as one of the following strings:
 - CLUSTER_COMPUTE_RESOURCE
 - DATASTORE
 - DATASTORE_CLUSTER
 - DV_PORTGROUP
 - DV_SWITCH
 - FOLDER
 - HOST
 - NETWORK
 - RESOURCE_POOL
 - STORAGE_PROFILE (available from vCenter 5.1 and later)
 - VIRTUAL_MACHINE
- *vimObjMoref* is the vSphere managed object reference of the object, as returned by the vCloud API.

For an example request URL, see the request portion of [“Example: Retrieve the vSphere URL of a Resource Pool,”](#) on page 237. All of the information you need to construct the URL is available when you retrieve the XML representation of any of the supported object types. See [“Example: Retrieve a Resource Pool Set,”](#) on page 213, [“Example: Retrieve a List of Available Portgroups and Switches from a vCenter Server,”](#) on page 203, and [“Example: Retrieve a List of Storage Profiles from a vCenter Server,”](#) on page 208.

NOTE See [“Mapping a vCloud Director Object to a vSphere Object,”](#) on page 237 for a list of vCloud Director objects and corresponding vSphere objects.

Prerequisites

Verify that you are logged in to the vCloud API as a system administrator.

Verify that the vSphere Web Client URL is enabled for all vCenter servers from which you want to retrieve the vSphere URL of an object. You can manage this feature on the **General** tab of the vSphere Properties page of the vCloud Director Web console.

Procedure

- 1 Retrieve the `VimObjectType` and `MoRef` elements of the object, and the `VimServerRef` element of the vSphere server that hosts the object.

- 2 Construct a request URL that includes the `VimServerRef`, `VimObjectType`, and `MoRef` elements.

You can use the value of the `href` attribute of the `VimServerRef` element as the `vimServer/id` part of the request URL.

- 3 Make a GET request to the URL you constructed in [Step 2](#).

The response contains a URL you can use with the vSphere Web Client. See [“Example: Retrieve the vSphere URL of a Resource Pool,”](#) on page 237.

Example: Retrieve the vSphere URL of a Resource Pool

This request retrieves the vSphere URL of one of the resource pools referenced in [“Example: Retrieve a Resource Pool Set,”](#) on page 213.

Request:

GET

```
https://vcloud.example.com/api/admin/extension/vimServer/9/RESOURCE_POOL/resgroup-195/vSphereWebClientUrl
```

Response:

```
<?xml version="1.0" encoding="UTF-8"?>
<vmext:VSphereWebClientUrl
  xmlns:vmext="http://www.vmware.com/vcloud/extension/v1.5"
  ...>
  <vmext:URL>https://10.147.22.134:8443/vsphere-client/#extensionId=vsphere.core.sp.summary;...
  </vmext:URL>
</vmext:VSphereWebClientUrl>
```

Use the URL that the `vmext:URL` element contains to access the object with the vSphere Web Client. The URL is truncated in this example.

Mapping a vCloud Director Object to a vSphere Object

When using the vSphere Web Client to examine a vCloud Director object, you can use this table to map the vCloud Director to its representation in vSphere.

Table 7-4. Mapping a vCloud Director Object to a vSphere Object

vCloud Director Object	vSphere Object
Provider vDC:Hosts	Host
Provider vDC:Datastores	Datastore
Provider vDC:Storage Profiles	Storage Profile
Provider vDC:External Network (portgroup-backed)	Standard virtual switch
Provider vDC:External Network(VLAN-backed)	Distributed virtual switch

Table 7-4. Mapping a vCloud Director Object to a vSphere Object (Continued)

vCloud Director Object	vSphere Object
Provider vDC:Resource Pool (root resource pool or Cluster)	Cluster
Provider vDC:Resource Pool (sub-resource pool only)	Resource Pool
External Network (Portgroup-backed)	Standard virtual switch
Direct Org VDC Networks (Portgroup)	Standard virtual switch
External Network (VLAN-backed)	Distributed virtual switch
Direct Org VDC Networks (VLAN)	Distributed virtual switch
Network Pool (VLAN-backed)	Distributed virtual switch
Network Pools (Portgroup-backed)	Standard virtual switch
vCenter	vCenter Server
Resource Pool (root resource pool or Cluster)	Cluster
Resource Pool (sub-resource pool only)	Resource Pool
Host	Host
Datastores	Datastore
Datastore Clusters	Datastore Cluster
Storage Profiles	N/A
vDSwitch	Distributed virtual switch
Port Group	Standard virtual switch
Home:vApp	Folder
vApp	Folder
vApp:vApp Diagram	Folder
vApp:vApp Diagram:VM	VM
vApp:VM List:VM	VM
vApp:Networking:vApp Networks	Distributed Port Group
VM	VM
Expired Items:vApps	Folder
Expired Items:vApp Templates	Folder
Catalog:vApp Template	Folder
Catalog:vApp Template:VM	VM
Catalog:vApp Template:Shadow VMs	VM
Organization vDC:vApp	Folder
Organization vDC:vApp Template	Folder
Organization vDC:Storage Profiles	Storage Profile
Organization vDC:Org VDC Networks (VLAN-backed)	Distributed virtual switch
Organization vDC:Org VDC Networks (Portgroup-backed)	Standard virtual switch
Organization vDC:Resource Pools (root resource pool or Cluster)	Cluster
Organization vDC:Resource Pools (sub-resource pool only)	Resource Pool

Working With Object Metadata

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 up to 1024 `name=value` pairs that the owner of an object can create, retrieve, update, and delete. It also contains an additional group of `name=value` pairs that are under the control of the system administrator.

Object metadata gives cloud operators and cloud tenants a flexible way to apply user-defined 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, and can be included in query filter expressions (see [“Add a Metadata Filter to a Query,”](#) on page 260).

vCloud API Object Metadata Links

The representation of any object that supports 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

A `Metadata` element can contain up to 1024 `MetadataEntry` elements. Each `MetadataEntry` includes a single `name=value` pair, represented in its `Key` and `TypedValue` elements. Key names are specified by `Key` element contents. A key name is a UTF-8 (Unicode) string encoded as described in RFC3986 (pct-encoded).

The key name must be unique within the scope of the containing `Metadata` element. Because key names are implicitly qualified by the `Domain` value of the containing `MetadataEntry`, the two key names in this example are considered to be unique.

```
<Metadata
  ...
  <MetadataEntry>
    <Domain
      visibility="READONLY">SYSTEM</Domain>
    <Key>Foo</Key>
    ...
  </MetadataEntry>
  <MetadataEntry>
    <Domain>GENERAL</Domain>
```

```

    <Key>Foo</Key>
    ...
  </MetadataEntry>
</Metadata>

```

See “Access Control for vCloud API Object Metadata,” on page 240 for more information about the Domain element.

The type of a Value is expressed in the `xsi:type` attribute of the containing TypedValue element. Values have various restrictions, based on their type.

Table 8-1. Metadata TypedValue Types

Type Name	Restrictions on Value	Size of Value
MetadataStringValue	UTF-8 character set. Strings longer than 1000 characters cannot be searched for in a query.	Depends on length of string.
MetadataNumberValue	Must be a signed 8-byte integer.	8 bytes
MetadataDateTimeValue	UTC date and time in the format defined by http://www.w3.org/TR/xmlschema-2/#http://www.w3.org/TR/xmlschema-2//dateTime . For example, 2012-06-18T12:00:00-05:00	8 bytes
MetadataBooleanValue	Must be one of: 1, 0, true, false	1 byte

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 element in the same Domain, 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 element in the same Domain, the MetadataEntry containing that Key is replaced.

Access Control for vCloud API Object Metadata

The Domain element of a MetadataEntry controls access to that entry by users and system administrators. There are two access domains.

GENERAL

User access to metadata in the GENERAL domain is controlled by the user's rights to the object to which the Metadata is attached. The owner of the object can create, read, update, or delete any MetadataEntry in the GENERAL domain. A MetadataEntry that does not include a Domain element is considered to be in the GENERAL domain.

SYSTEM

Metadata in the SYSTEM domain can be created, updated, and deleted only by a system administrator. User access to metadata in the SYSTEM domain is controlled by the value of the visibility attribute of the Domain element.

Table 8-2. Domain Visibility Attribute Values

Value	Meaning
PRIVATE	The metadata is visible to system administrators only.
READONLY	the metadata is visible to system administrators and the object owner.

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 UTF-8 characters.
Metadata size	The size of all Metadata for an object, computed as the sum of all Key and TypedValue UTF-8 strings in all MetadataEntry elements in the GENERAL domain, cannot exceed 128 KB. An additional 16KB of MetadataEntry content can be created in the SYSTEM domain.
MetadataEntry limit	The total metadata associated with an object cannot exceed 1024 MetadataEntry elements in the GENERAL domain and 128 MetadataEntry elements in the SYSTEM domain.

This chapter includes the following topics:

- [“Retrieve or Update a Metadata Element,”](#) on page 241
- [“Retrieve or Update a Metadata Value,”](#) on page 244

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 contains 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 TypedValue. 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 244.

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 8-3. Content Restrictions for Key

Cannot Contain	Cannot Start with	Cannot End with
/. ./	./	/.
/. /	.. /	/. .

Prerequisites

Verify that you have rights to read or modify the metadata. See [“Access Control for vCloud API Object Metadata,”](#) on page 240

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 239.

4 POST the Metadata element to the `rel="add"` link described in [Step 2](#).

See [“Example: Update a Metadata Element,”](#) on page 242.

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. In this example, the `Metadata` element contains `MetadataEntry` elements for which the `Domain` is `SYSTEM`. Only the system administrator can update these elements.

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
  xmlns="http://www.vmware.com/vcloud/v1.5"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  type="application/vnd.vmware.vcloud.metadata+xml">
  <MetadataEntry
    type="application/vnd.vmware.vcloud.metadata.value+xml">
    <Domain
      visibility="READONLY">SYSTEM</Domain>
    <Key>Organization Web Page</Key>
    <TypedValue
      xsi:type="MetadataStringValue">
      <Value>http://internal.example.com/orgs/Finance</Value>
    </TypedValue>
  </MetadataEntry>
  <MetadataEntry
    type="application/vnd.vmware.vcloud.metadata.value+xml">
    <Domain
```

```

        visibility="PRIVATE">SYSTEM</Domain>
    <Key>LOS</Key>
    <TypedValue
        xsi:type="MetadataStringValue">
        <Value>bronze</Value>
    </TypedValue>
</MetadataEntry>
</Metadata>

```

The response is a Task.

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 MetadataEntry elements.

GET <https://vcloud.example.com/api/admin/org/26/metadata>

```

...
<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/SYSTEM/Organization+Web+Page" />
    <Link
      rel="remove"
      type="application/vnd.vmware.vcloud.metadata.value+xml"
      href="https://vcloud.example.com/api/org/26/metadata/SYSTEM/Organization+Web+Page" />
    <Link
      rel="up"
      type="application/vnd.vmware.vcloud.metadata+xml"
      href="https://vcloud.example.com/api/org/26/metadata" />
  <Domain
    visibility="READONLY">SYSTEM</Domain>
  <Key>Organization Web Page</Key>

```

```

    <TypedValue
      xsi:type="MetadataStringValue">
      <Value>http://internal.example.com/orgs/Finance</Value>
    </TypedValue>
  </MetadataEntry>
</MetadataEntry>
  <Link
    rel="edit"
    type="application/vnd.vmware.vcloud.metadata.value+xml"
    href="https://vcloud.example.com/api/org/26/metadata/SYSTEM/LOS" />
  <Link
    rel="remove"
    type="application/vnd.vmware.vcloud.metadata.value+xml"
    href="https://vcloud.example.com/api/org/26/metadata/SYSTEM/LOS" />
  <Link
    rel="up"
    type="application/vnd.vmware.vcloud.metadata+xml"
    href="https://vcloud.example.com/api/org/26/metadata" />
  <Domain
    visibility="PRIVATE">SYSTEM</Domain>
  <Key>LOS</Key>
  <TypedValue
    xsi:type="MetadataStringValue">
    <Value>bronze</Value>
  </TypedValue>
</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 to the vCloud API as a system administrator or the object owner.
- Retrieve the object's `Metadata` element. See [“Retrieve or Update a Metadata Element,”](#) on page 241

Procedure

- 1 Examine the retrieved `Metadata` element to find the `MetadataEntry` you want.

Each `MetadataEntry` contains a link of the following form, which you can use when updating the `Value` of the entry:

```

<Link
  rel="edit"
  type="application/vnd.vmware.vcloud.metadata.value+xml"
  href="https://vcloud.example.com/api/admin/object/id/metadata/DOMAIN/key-name" />

```

For example:

```

<Link
  ...
  href="https://vcloud.example.com/api/admin/org/26/metadata/SYSTEM/LOS/>

```

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 retrieve the value, make a GET request to URL in the value of the href attribute of the MetadataEntry. The response is a MetadataValue element.
- To update the value, make a PUT request to the URL in the edit link described in [Step 1](#), and supply a MetadataValue element as the request body. See [“Example: Update a Metadata Value,”](#) on page 245

NOTE The href value of a MetadataEntry in the SYSTEM domain includes the SYSTEM qualifier. The href value of a MetadataEntry in the GENERAL domain does not include a qualifier.

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 242) to a new value of silver. Note that because this MetadataValue is contained by a MetadataEntry where the Domain is SYSTEM and visibility is PRIVATE, only a system administrator can update it.

Request:

```
PUT https://vcloud.example.com/api/admin/org/26/metadata/SYSTEM/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"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <TypedValue
    xsi:type="MetadataStringValue">
    <Value>silver</Value>
  </TypedValue>
</MetadataValue>
```

The response is a task.

Response:

```
<Task
  ...
  operation="Updating metadata for Organization (26)"
  ... >
  ...
</Task>
```


Using the Query Service

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. A *name* can be either a static or dynamic. Static names are predefined object property names, and are returned in an object-specific `QueryRecordResultType` element. For a list of these names, see the `Queries` reference pages in the *vCloud API Schema Reference*. Dynamic names are user-defined metadata key names. Dynamic *name=value* pairs are not returned unless the object includes a non-empty `Metadata` element. See [Chapter 8, “Working With Object Metadata,”](#) on page 239.
- An `idrecords` format that is identical to the `records` format, except that object reference *values* are returned in `id` format rather than `href` format. See [“Objects, References, and Representations,”](#) on page 12.
- 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,”](#) on page 248
- [“Packaged Queries,”](#) on page 252
- [“Query Parameters,”](#) on page 257
- [“Add a Metadata Filter to a Query,”](#) on page 260

Typed Queries

Typed queries require you to construct a request URL that specifies an object type and optional parameters. Use this URL with a GET request to return query results.

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 257.
- *filter* is an optional filter expression. At most one filter expression is allowed. See “[Filter Expressions](#),” on page 257.

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 made. 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 257.

Query Reference

The vCloud API schema reference includes reference material for all queries.

Example: Listing All Queries

You can retrieve a summary list of all typed queries types accessible to the currently authenticated user by making a request like this one:

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

The response is a QueryList element that contains a Link for every query. The href value of each Link is a URL you can GET to run the query specifying the default format.

```
<?xml version="1.0" encoding="UTF-8"?>
<QueryList
  type="application/vnd.vmware.vcloud.query.queryList+xml"
  href="https://vcloud.example.com/api/query"
  ... >
  <Link
    rel="down"
    type="application/vnd.vmware.vcloud.query.references+xml"
    name="organization"
    href="https://vcloud.example.com/api/query?type=organization&format=references" />
  <Link
    rel="down"
    type="application/vnd.vmware.vcloud.query.records+xml"
    name="organization"
    href="https://vcloud.example.com/api/query?type=organization&format=records" />
```



```

<Link
  rel="down"
  type="application/vnd.vmware.vcloud.query.idrecords+xml"
  name="organization"
  href="https://vcloud.example.com/api/query?type=organization&format=idrecords" />
<Link
  rel="down"
  type="application/vnd.vmware.vcloud.query.references+xml"
  name="adminOrgNetwork"
  href="https://vcloud.example.com/api/query?type=adminOrgNetwork&format=references" />
...
</QueryList>

```

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.

Example: Simple Typed Query

This simple typed query retrieves a list of all users in your organization and returns a response in the 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&page=1&pageSize=25&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&page=1&pageSize=25&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>

```

This simple typed query retrieves the same list of all users in your organization and returns a response in the 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&page=1&pageSize=25&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&page=1&pageSize=25&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>

```

Example: Query With Parameters

This query retrieves a list of all users in your organization and returns a response in the records format. The query includes a `sortAsc=name` parameter, so the result set is 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&page=1&pageSize=25&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&page=1&pageSize=25&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>

```

This query retrieves a list of all users in your organization and returns a response in the records format. The query includes a `filter=ldapUser==true` parameter, so the result set lists the subset of users who are imported from LDAP. Note that you can filter on a record (attribute) value even though you specify the references format.

```
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&page=1&pageSize=25&format=references&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/query?type=adminUser&page=1&pageSize=25&format=re
cords&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 257.
- *filter* is an optional filter expression. At most one filter expression is allowed. See “[Filter Expressions](#),” on page 257.

Query Categories

Packaged queries are divided into the following categories:

User queries	The queries have the form <i>API-URL/object-type/query</i> . Any user can run these queries.
Administrator queries	The queries have the form <i>API-URL/admin/object-type/query</i> . An organization administrator can run these queries.
Extension queries	The queries have the form <i>API-URL/admin/extension/object-type/query</i> . A system administrator can run these queries.

The response is a `QueryList` element that contains a `Link` for every query. The `href` value of each `Link` is a URL you can GET to run the query specifying the default format.

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 made. 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 257.

Table 9-1. Packaged Queries

Query URL	Result Set
<i>API-URL/catalogs/query</i>	All catalogs in your organization that you have rights to view or modify
<i>API-URL/mediaList/query</i>	All media that you can view or modify
<i>API-URL/vAppTemplates/query</i>	All vApp templates that you can view or modify
<i>API-URL/vApps/query</i>	All vApps that you can view or modify
<i>API-URL/vms/query</i>	All virtual machines that you can view or modify
<i>API-URL/admin/groups/query</i>	Groups in all organizations in the system
<i>API-URL/admin/users/query</i>	Users in all organizations in the system
<i>API-URL/admin/strandedUsers/query</i>	Stranded users in the organization. When you delete an LDAP group, users who were imported from that group become stranded and cannot log in.
<i>API-URL/admin/roles/query</i>	All roles defined in the system
<i>API-URL/admin/rights/query</i>	All rights defined in the system
<i>API-URL/admin/orgs/query</i>	All organizations visible to you
<i>API-URL/admin/vdcs/query</i>	All vDCs in the system
<i>API-URL/admin/extension/hostReferences/query</i>	ESX/ESXi hosts registered to the system
<i>API-URL/admin/extension/datastores/query</i>	All datastores in the system
<i>API-URL/admin/extension/externalNetworkReferences/query</i>	External networks in the system
<i>API-URL/admin/extension/networkPoolReferences/query</i>	All network pools in the system
<i>API-URL/admin/extension/providerVdcReferences/query</i>	All provider vDCs in the system
<i>API-URL/admin/extension/vimServerReferences/query</i>	All vCenter servers registered to the system
<i>API-URL/admin/extension/orgNetworks/query</i>	All organization networks in the system
<i>API-URL/admin/extension/vapps/query</i>	All vApps in the system
<i>API-URL/admin/extension/orgVdcs/query</i>	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&pageSize=25&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&page=1&pageSize=25&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&pageSize=25&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&page=1&pageSize=25&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&pageSize=25&format=records&filter=numberOfTemplates!
=0&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&page=1&pageSize=25&format=references&filter=numberOfTemplates!
=0&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 9-2. Query Parameters

Parameter Name	Parameter Description	Default
<code>type</code>	The type of the query. Type names are case-sensitive. See “Query Types,” on page 248.	None. This parameter is required for all typed queries, and is not allowed for any packaged query.
<code>sortAsc=attribute-name</code>	Sort results by <i>attribute-name</i> in ascending order. <i>attribute-name</i> cannot include metadata.	Sorted by database ID
<code>sortDesc=attribute-name</code>	Sort results by <i>attribute-name</i> in descending order. <i>attribute-name</i> cannot include metadata.	Sorted by database ID
<code>page</code>	If the query results span multiple pages, return this page.	<code>page=1</code>
<code>pageSize</code>	Number of results per page, to a maximum of 128.	<code>pageSize=25</code>
<code>format</code>	One of the following types: references Returns a reference to each object, including its <code>name</code> , <code>type</code> , and <code>href</code> attributes. records Returns all database records for each object, with each record as an attribute. idrecords Identical to the <code>records</code> format, except that object references are returned in <code>id</code> format rather than <code>href</code> format.	<code>format=records</code>
<code>fields</code>	Comma-separated list of attribute names or metadata key names to return. For example, <code>fields=name,isEnabled</code> or <code>fields=metadata:rank</code> . See “Specifying Metadata in a Query or a Filter Expression,” on page 258.	Returns all static attribute names. Returns metadata only if the object has a non-empty Metadata element.
<code>offset</code>	Integer value specifying the first record to return. Record numbers < <code>offset</code> are not returned.	<code>offset=0</code>
<code>filter</code>	Filter expression. See Table 9-3	None

Filter Expressions

For queries that do not examine object metadata, you can filter results using string matching or numeric comparison operations. A filter comprises one or more subexpressions drawn from the following set of operators.

Table 9-3. Query Filter Expressions

Operator	Example	Operation
==	<i>attribute==value</i>	Matches. The example evaluates to true if <i>attribute</i> has a value that matches <i>value</i> in a case-sensitive comparison. NOTE Asterisk (*) characters that appear anywhere in <i>value</i> are treated as wildcards that match any character string. When <i>value</i> includes wildcards, the comparison with <i>attribute</i> becomes case-insensitive.
!=	<i>attribute!=value</i>	Does not match. The example evaluates to true if <i>attribute</i> has a value that does not match <i>value</i> in a case-sensitive comparison. Wildcard characters are not allowed.
;	<i>attribute1==value1;attribute2!=value2</i>	Logical AND. The example evaluates to true only if <i>attribute1</i> has a value that matches <i>value1</i> and <i>attribute2</i> has a value that does not match <i>value2</i> in a case-sensitive comparison.
,	<i>attribute1==value1,attribute2==value2</i>	Logical OR. The example evaluates to true if <i>attribute1</i> has a value that matches <i>value1</i> or <i>attribute2</i> has a value that matches <i>value2</i> in a case-sensitive comparison.
=gt=	<i>attribute=gt=value</i>	Greater than. The example evaluates to true if <i>attribute</i> has a value that is greater than <i>value</i> . Both <i>attribute</i> and <i>value</i> must be of type <code>int</code> , <code>long</code> , or <code>dateTime</code> .
=lt=	<i>attribute=lt=value</i>	Less than. The example evaluates to true if <i>attribute</i> has a value that is less than <i>value</i> . Both <i>attribute</i> and <i>value</i> must be of type <code>int</code> , <code>long</code> , or <code>dateTime</code> .
=ge=	<i>attribute=ge=value</i>	Greater than or equal to. The example evaluates to true if <i>attribute</i> has a value that is greater than or equal to <i>value</i> . Both <i>attribute</i> and <i>value</i> must be of type <code>int</code> , <code>long</code> , or <code>dateTime</code> .
=le=	<i>attribute=le=value</i>	Less than or equal to. The example evaluates to true if <i>attribute</i> has a value that is less than or equal to <i>value</i> . Both <i>attribute</i> and <i>value</i> must be of type <code>int</code> , <code>long</code> , or <code>dateTime</code> .

Specifying Metadata in a Query or a Filter Expression

Because metadata values are dynamic and metadata names have an optional domain qualifier queries that filter metadata must specify a qualified name, a value, and the type of the value.

- The domain must be specified for any `MetadataEntry` that is in the `SYSTEM` domain. If no `DOMAIN` is specified, the query returns the value for key name in the `GENERAL` domain, if it exists.
- The type of the value must be specified, using one of the following keywords.

Table 9-4. Metadata Type Specifiers in Query Filters

Type Name as Specified in TypedValue	Type Name as a Filter Expression Keyword
<code>MetadataStringValue</code>	<code>STRING</code>
<code>MetadataNumberValue</code>	<code>NUMBER</code>
<code>MetadataDateTimeValue</code>	<code>DATETIME</code>
<code>MetadataBooleanValue</code>	<code>BOOLEAN</code>

For queries that examine object metadata, you can filter query results using numeric comparison operations when a metadata value has type NUMBER or DATETIME. Because object metadata types are dynamic, filter expressions for metadata queries require additional parameters that define the *attribute* part of the expression as metadata, and specify the type of the *value* part of the expression.

Table 9-5. Metadata Query Filter Expressions

Operator	Example	Operation
=gt=	metadata:attribute=gt=[NUMBER DATETIME]:value	Greater than. The example evaluates to true if <i>value</i> is of type NUMBER or DATETIME and the value of the metadata key named <i>attribute</i> is greater than <i>value</i> .
=lt=	metadata:attribute=lt=[NUMBER DATETIME]:value	Less than. The example evaluates to true if <i>value</i> is of type NUMBER or DATETIME and the value of the metadata key named <i>attribute</i> is less than <i>value</i> .
=ge=	metadata:attribute=ge=[NUMBER DATETIME]:value	Greater than or equal to. The example evaluates to true if <i>value</i> is of type NUMBER or DATETIME and the value of the metadata key named <i>attribute</i> is greater than or equal to <i>value</i> .
=le=	metadata:attribute=le=[NUMBER DATETIME]:value	Less than or equal to. The example evaluates to true if <i>value</i> is of type NUMBER or DATETIME and the value of the metadata key named <i>attribute</i> is less than or equal to <i>value</i> .

You can specify up to 8 metadata fields in a single query. You cannot use a metadata value as a sort key when sorting query output.

The syntax for specifying metadata in a fields parameter is:

```
fields=metadata[@SYSTEM]:key-name
```

For example:

```
fields=metadata:rank
```

```
fields=metadata@SYSTEM:expiry
```

The syntax for specifying a metadata field in a filter expression is:

```
metadata[@SYSTEM]:key-name operator type-keyword:value
```

For example:

```
metadata:rank=ge=NUMBER:1
```

```
metadata@SYSTEM:expiry=le=DATETIME:2012-06-18T12:00:00-05:00
```

Grouping Filter Subexpressions

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. Use the schema reference to find the attribute names included in a particular result set. For most queries, the name of the element that holds a result is derived from the name and format of the query. For example, the result of an `adminGroup` query in `records` format is an `AdminGroupRecord` element. You can search for result set elements by entering the element name in the **Quick Index** window. For each result set element, the reference page includes a table of attribute names, with their types and descriptions.

Add a Metadata Filter to a Query

All packaged queries and most typed queries return object metadata if it exists for an object in the result set. You can add metadata-specific filter criteria to a query.

To query an object's metadata, add a metadata field specifier to the query. This specifier follows the `fields` parameter and consists of the string `metadata:` followed by the name of a metadata key.

Procedure

- 1 Create the filter expression.
- 2 Add a `fields` parameter that specifies metadata.

The syntax for specifying a metadata field is:

```
fields=metadata:key-name [, metadata:key-name ] ...
```

- 3 Add a filter expression to the query.

All of the standard filter expressions are supported for metadata queries. In addition, a set of numeric comparisons are available for values of type `NUMBER` or `DATETIME`.

Example: A Query With a Metadata Filter

Assume that the system administrator adds metadata specifying an expiry date and a priority rank to every organization in a cloud. The key name for the expiry date is `expiry`, and its value is expressed as a `MetadataDateTimeValue`. The key name for the rank is `rank`, and its value is expressed as a `MetadataNumberValue`. The following query includes a `fields` parameter specifying those metadata keys by name.

NOTE You must use a `metadata@SYSTEM` parameter to specify metadata keys in the `SYSTEM` domain. Metadata in the `GENERAL` domain does not require a qualifier. Up to eight metadata parameters can be included in a query.

Request:

```
GET https://vcloud.example.com/api/query?
type=organization&format=records&fields=metadata:rank,metadata@SYSTEM:expiry"
```

The response includes an `OrgRecord` for each organization in the cloud. Each `OrgRecord` includes a `Metadata` element that contains `MetadataEntry` elements for each of the requested keys.

Response:

```
<QueryResultRecords
  xmlns="http://www.vmware.com/vcloud/v1.5"
  total="3"
  pageSize="25"
  page="1"
  ...
  <OrgRecord
```

```

storedVMQuota="0"
numberOfVdcs="5"
...
href="https://vcloud.example.com/api/org/70"
numberOfRunningVMs="3">
<Metadata>
  <MetadataEntry>
    <Domain
      visibility="PRIVATE">SYSTEM</Domain>
    <Key>expiry</Key>
    <TypedValue
      xsi:type="MetadataDateTimeValue">
        <Value>2012-05-01T00:00:00.000-04:00</Value>
      </TypedValue>
    </MetadataEntry>
  <MetadataEntry>
    <Key>rank</Key>
    <TypedValue
      xsi:type="MetadataNumberValue">
        <Value>1</Value>
      </TypedValue>
    </MetadataEntry>
  </Metadata>
</OrgRecord>
<OrgRecord
  storedVMQuota="0"
  numberOfVdcs="2"
  ...
  href="https://vcloud.example.com/api/org/77"
  numberOfRunningVMs="0">
  <Metadata>
    <MetadataEntry>
      <Key>expiry</Key>
      <TypedValue
        xsi:type="MetadataDateTimeValue">
          <Value>2012-06-01T00:00:00.000-04:00</Value>
        </TypedValue>
      </MetadataEntry>
    <MetadataEntry>
      <Key>rank</Key>
      <TypedValue
        xsi:type="MetadataNumberValue">
          <Value>2</Value>
        </TypedValue>
      </MetadataEntry>
    </Metadata>
  </OrgRecord>
</QueryResultRecords>

```

To filter the query result set so that it contains only those organizations whose expiry date is later than May 1, 2012, add a filter for the expiry value, as this example shows.

Request:

```
GET https://vcloud.example.com/api/query?
type=organization&format=records&fields=metadata:rank,metadata@SYSTEM:expiry"
&filter="metadata@SYSTEM:expiry=lt=DATETIME:2012-05-01T00:00:00.000-04:00
```

Response:

```
<QueryResultRecords
  xmlns="http://www.vmware.com/vcloud/v1.5"
  total="3"
  pageSize="25"
  page="1"
  ...
  <OrgRecord
    storedVMQuota="0"
    numberOfVdcs="2"
    ...
    href="https://vcloud.example.com/api/org/77"
    numberOfRunningVMs="0">
    <Metadata>
      <MetadataEntry>
        <Key>expiry</Key>
        <TypedValue
          xsi:type="MetadataDateTimeValue">
            <Value>2012-06-01T00:00:00.000-04:00</Value>
        </TypedValue>
      </MetadataEntry>
      <MetadataEntry>
        <Key>rank</Key>
        <TypedValue
          xsi:type="MetadataNumberValue">
            <Value>2</Value>
        </TypedValue>
      </MetadataEntry>
    </Metadata>
  </OrgRecord>
</QueryResultRecords>
```

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 Extensibility page of the vCloud Director Web console, and provide either of the following:

- an SSL certificate pathname
- a JCEKS trust store pathname and password

If you do not need to validate the AMQP broker's certificate, 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 267.

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 263. 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,”](#) on page 264
- [“Retrieve or Update Blocking Task Settings,”](#) on page 273
- [“Monitor Blocking Tasks,”](#) on page 276
- [“Take Action on a Blocking Task,”](#) on page 277
- [“Extend The Timeout Expiration of an Active Task,”](#) on page 279

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 266.

Prerequisites

Verify that you are logged in to the vCloud API 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 265

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 includes information supplied in the request, and contains a link you can use to test the settings. The value of `AmqpPassword`, which you must supply when you modify `AmqpSettings` (even if you are not changing its value), is never returned when you retrieve `AmqpSettings`.

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.direct</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 to the vCloud API as a system administrator.
- Verify that you know the AMQP broker password.
- Retrieve the `SystemSettings` element. See [“Retrieve or Update System Settings,”](#) on page 197.

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 265 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 265

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 <code>com/vmware/vcloud/event/</code> , triggered the notification. For more information about object identifiers, see “ Objects, References, and Representations ,” on page 12.
org	The object identifier of the organization that owns the affected object.
user	The object identifier of the user who made the request.
subType1-subTypeN	Each <i>subType</i> is a single component of the event type name. See “ Notification Types ,” on page 269.
taskName	If <i>entity</i> 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
<code>notification.type</code>	See “ Notification Types ,” on page 269.
<code>notification.entityType</code>	The type of vCloud entity is associated with this notification. For example, <code>vm</code> .
<code>notification.entityUUID</code>	The object identifier of the object on which an operation, an event of type <code>com/vmware/vcloud/event/</code> , triggered the notification.
<code>notification.orgUUID</code>	The object identifier of the organization that owns the affected object.
<code>notification.userUUID</code>	The object identifier of the user who made the request.
<code>notification.operationSuccess</code>	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 312.

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 278.

Notification Types

The value of the type attribute of a vCloud Director notification is a string of 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/complete	A task has completed.
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.

Table 10-5. Other System Events (Continued)

Type (com/vmware/vcloud/event/)	Description
blockingtask/fail	A blocking task was failed.
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 268.

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 274.

Prerequisites

- Verify that you are logged in to the vCloud API as a system administrator.
- Retrieve the `SystemSettings` element. See [“Retrieve or Update System Settings,”](#) on page 197.

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 273.

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 274 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 contains information extracted from 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+xml"
  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 273.

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 267), blocking tasks can be monitored and managed by a system administrator using the vCloud API.

Prerequisites

Verify that you are logged in to the vCloud API 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 277. 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 278.

- 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 278.

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"
  ... >
<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 276.

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 268.

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 <code>ABORTED</code> .

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 to the vCloud API as a system administrator.

Procedure

- 1 Retrieve the list of active blocking tasks.
See [“Monitor Blocking Tasks,”](#) on page 276. 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 278.
- 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 268 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
Content-Type: application/vnd.vmware.admin.blockingTask+xml
...
<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"
  createTime="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">
</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 to the vCloud API as a system administrator.

Procedure

- 1 Retrieve the list of active blocking tasks.
See [“Monitor Blocking Tasks,”](#) on page 276. 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 278.
- 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 280.
- 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) plus `timeoutValueInMilliseconds`.

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 278 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
Content-Type: application/vnd.vmware.admin.blockingTask+xml
...
<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>
```


vCloud Director Extension Services

vCloud Director and the vCloud API include a framework for integration of extension services that a vCloud API client can access as though they were native services. In addition to service-specific objects or operations they provide, extension services can implement new operations for existing API objects.

A vCloud Director extension service is a program that presents a REST interface to vCloud API clients. When you register an extension service with the vCloud API, you specify one or more URL patterns that the vCloud Director REST service treats as extension requests. When it receives an extension request, the vCloud Director REST service creates an AMQP notification with a service-specific exchange and routing key, and sends it to the vCloud Director AMQP service. Each extension service subscribes to AMQP notifications that have its service-specific routing key. A service processes its notifications, takes whatever actions they require, and returns a response to the AMQP service, where the vCloud Director REST service retrieves it and uses its contents to generate a response to the client that made the request.

Message Routing

Extension services use the vCloud Director AMQP service to communicate with vCloud Director. Every extension service must register a unique AMQP routing key, which vCloud Director prepends to AMQP messages destined for that service. To collect replies from services, vCloud Director creates a single reply exchange for all services, creates a separate reply queue for each cell, and binds each of those queues to the reply exchange.

vCloud Director extension services can also be vCloud API clients, authenticating to the vCloud API and making their own REST requests to the vCloud API URL. This type of interaction is required when creating tasks and events that track the progress of requests made to the service. It is also required by services that operate on vCloud Director objects like vApps and virtual machines.

Creating Events and Tasks

The vCloud API extension framework implements operations that allow an extension service to create and update an organization's lists of tasks and events, so the status of asynchronous events running in extension services can be displayed with the same kinds of information posted by native services.

Authorization Framework

All requests to extension services are processed through the vCloud Director authentication framework. A user making a request to an extension service must be authenticated by vCloud Director as a system administrator or a member of a vCloud Director organization.

An extension service can add service-specific rights and associate those rights with operations on its own objects or with operations it adds to vCloud API objects .

Service APIs

An extension service can define its own request and response body elements if it needs to. API schema files can be specified as part of service registration or can be added later. Schema files can reside at any location that is reachable by the extension service.

Support for Idempotent Operations

Most requests to extension services can include an `operationKey` attribute, which is a string meant to uniquely identify the operation so that if it is invoked multiple times, the result will be the same as if it had been invoked only once. The following types support use of the `operationKey` attribute:

- `AclRuleType`
- `ApiDefinitionType`
- `ApiFilterType`
- `FileDescriptorType`
- `ResourceClassActionType`
- `ResourceClassType`
- `ServiceLinkType`
- `ServiceResourceType`
- `ServiceType`

This chapter includes the following topics:

- [“Summary of vCloud API Extensibility Requests,”](#) on page 282
- [“Register an Extension Service,”](#) on page 284
- [“Adding or Removing Service Links,”](#) on page 288
- [“Service-Specific Tasks and Events,”](#) on page 291
- [“Authorization Framework for Extension Service Operations,”](#) on page 294
- [“Localization Framework for Extension Services,”](#) on page 302
- [“REST APIs for Extension Services,”](#) on page 302

Summary of vCloud API Extensibility Requests

vCloud API extensibility requests allow you to develop extension 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.

IMPORTANT Request URLs are always available in `Link` elements contained by the representation of the object on which they operate. URL forms shown here are for reference purposes only. Although URLs have a well-known syntax and a well-understood interpretation, a client should treat vCloud API request URLs as opaque strings. The rules that govern how the server constructs these strings might change in future releases.

This summary may not cover all requests in this category. For the complete list of requests, along with detailed information about input and output types, see the Operations lists in the schema reference.

Table 11-1. Summary of vCloud API Extensibility Requests

Operation	Request	Request Body	Response
Register an extension service with vCloud Director	POST <i>API-URL/admin/extension/service</i>	Service	Service
Retrieve a list of registered extension services.	GET <i>API-URL/admin/extension/service</i>	None	ExtensionServices
Update a registered extension service.	PUT <i>API-URL/admin/extension/service/id</i>	Service	Service
Delete a registered extension service.	DELETE <i>API-URL/admin/extension/service/id</i>	None	204 No Content
Register extension service links with vCloud Director objects.	POST <i>API-URL/admin/extension/service/id/links</i>	ServiceLink	ServiceLink
Create resource class definitions for a registered extension service.	POST <i>API-URL/admin/extension/service/id/resourceClasses</i>	ResourceClass	ResourceClass
Associate actions with a resource class.	POST <i>API-URL/admin/extension/service/resourceClass/id/resourceClassActions</i>	ResourceClassAction	ResourceClassAction
Create an ACL rule for a resource class.	POST <i>API-URL/admin/extension/service/resourceClassAction/id/aclRules</i>	AclRule	AclRule
Associate a right with an extension service.	POST <i>API-URL/admin/extension/service/id/rights</i>	Right	RightRight
Update a right associated with an extension service.	PUT <i>API-URL/admin/right/id</i>	Right	Right
Delete rights no longer associated with a role or ACL rule after the associated extension service has been deleted.	POST <i>API-URL/admin/extension/service/action/clearUnusedRights</i>	None	204 No Content
List the access rights for a specific combination of user and entity	POST <i>API-URL/admin/user/id/entityRights</i>	EntityReferences	UserEntityRights
Create a Task object in the organization/ <i>id</i>	POST <i>API-URL/tasksList/id</i>	Task	Task
Update a Task object	PUT <i>API-URL/task/id</i>	Task	Task
Post an extension service event to the system.	POST <i>API-URL/admin/org/id/events</i>	Event	204 No Content
Initiate the upload of a localization bundle for an extension service.	POST <i>API-URL/admin/extension/service/id/localizationBundles</i>	BundleUploadParams	BundleUploadSocket

Table 11-1. Summary of vCloud API Extensibility Requests (Continued)

Operation	Request	Request Body	Response
Create API definitions for a registered extension service.	POST <i>API-URL/admin/extension/service/id/definitions</i>	ApiDefinition	ApiDefinition
Create file descriptors for a registered extension service.	POST <i>API-URL/admin/extension/service/definition/id/files</i>	FileDescriptor	FileDescriptor

Register an Extension Service

Register an extension service to specify its namespace, AMQP exchange and routing key, and URL patterns. You can specify additional service properties during registration or update them later.

An extension service typically authenticates with the vCloud API as a system administrator, then registers itself by POSTing a *Service* element to the system's `.../api/admin/extension/service` URL. A *Service* element must include the following elements.

Namespace	The service namespace, which must be unique among all registered extension services. Service namespace names that follow the naming convention used for Java packages (for example, <code>com.example.service.backup</code>) are more likely to be unique. If a service tries to register a namespace that is already registered, registration fails.
RoutingKey	The AMQP routing key that vCloud Director must use when routing messages to the service.
Exchange	The AMQP exchange name that vCloud Director must use when routing messages to the service. The service must create the specified exchange on the AMQP service that vCloud Director uses. The exchange type must be <code>direct</code> .
ApiFilters	Specifies one or more URL patterns that the vCloud Director REST service must treat as extension requests. URL patterns can include regular expressions that <code>java.util.regex.Pattern</code> recognizes. See “Create an API Filter for an Extension Service,” on page 302.

Registration can also specify the following optional properties:

- Definitions of *Link* elements that the service adds to the representations of vCloud API objects. See [“Adding or Removing Service Links,”](#) on page 288
- Authorization framework for controlling access to the service's objects and operations. See [“Authorization Framework for Extension Service Operations,”](#) on page 294.
- Locations of schema files if the service provides its own API. See [“REST APIs for Extension Services,”](#) on page 302

You can also create or update these properties after you register the service.

Prerequisites

Verify that you are logged in to the vCloud API 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/extension
```

- 2 Examine the response to find the Link for adding extension services.

This URL is present in the `VMWExtension` element, and has the following form.

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

- 3 Construct a Service element.

See the request portion of “[Example: Register an Extension Service](#),” on page 285.

- 4 POST the Service element to the URL described in [Step 2](#).

See the request portion of “[Example: Register an Extension Service](#),” on page 285.

Example: Register an Extension Service

This request registers an extension service named `SDK-BackupExtension`. The request specifies the service namespace and routing key, and several URL patterns to be used as API filters. Messages for the service are sent to the AMQP exchange named `sdkext` with routing key `sdkbackup`.

NOTE If the specified exchange does not exist on the AMQP service that vCloud Director uses, an internal server error occurs whenever vCloud Director receives a request that matches one of the service's API filters.

This request also includes several `ServiceLink` elements. For information about the contents of these elements, see “[Adding or Removing Service Links](#),” on page 288.

Request:

POST `https://vcloud.example.com/api/admin/extension/service`

Content Type: `application/vnd.vmware.admin.service+xml`

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

```
<vmext:Service
```

```
  xmlns="http://www.vmware.com/vcloud/v1.5"
```

```
  xmlns:vmext="http://www.vmware.com/vcloud/extension/v1.5"
```

```
  name="SDK-BackupExtension">
```

```
    <vmext:Namespace>org.example.vcd.backup</vmext:Namespace>
```

```
    <vmext:Enabled>true</vmext:Enabled>
```

```
    <vmext:AuthorizationEnabled>true</vmext:AuthorizationEnabled>
```

```
    <vmext:RoutingKey>backup</vmext:RoutingKey>
```

```
    <vmext:Priority>50</vmext:Priority>
```

```
    <vmext:Exchange>sdkext</vmext:Exchange>
```

```
    <vmext:ApiFilters>
```

```
      <vmext:ApiFilter>
```

```
        <vmext:UrlPattern>(/api/org/.*/backups)|(/api/vApp/vapp-.*/backups)|
```

```
(/api/vApp/vapp-.*/action/backup)|(/api/backup/.*)</vmext:UrlPattern>
```

```
      </vmext:ApiFilter>
```

```
    </vmext:ApiFilters>
```

```
    <vmext:ServiceLinks>
```

```
      <vmext:ServiceLink>
```

```
        <vmext:LinkHref>{baseUri}org/{resourceId}</vmext:LinkHref>
```

```
        <vmext:MimeType>application/vnd.example.vcd-backup.backupList+xml</vmext:MimeType>
```

```
        <vmext:Rel>down</vmext:Rel>
```

```
        <vmext:ResourceType>application/vnd.example.vcd-backup.org+xml</vmext:ResourceType>
```

```
      </vmext:ServiceLink>
```

```
      <vmext:ServiceLink>
```

```
        <vmext:LinkHref>{baseUri}api/vApp/vapp-{resourceId}/backups</vmext:LinkHref>
```

```

    <vmext:MimeType>application/vnd.example.vcd-backup.backupList+xml</vmext:MimeType>
    <vmext:Rel>down</vmext:Rel>
    <vmext:ResourceType>application/vnd.example.vcd-backup.vApp+xml</vmext:ResourceType>
  </vmext:ServiceLink>
  <vmext:ServiceLink>
    <vmext:LinkHref>{baseUri}vApp/vapp-{resourceId}/action/backup</vmext:LinkHref>
    <vmext:MimeType>application/vnd.example.vcd-backup.createBackupParams
+xml</vmext:MimeType>
    <vmext:Rel>backup</vmext:Rel>
    <vmext:ResourceType>application/vnd.example.vcd-backup.vApp+xml</vmext:ResourceType>
  </vmext:ServiceLink>
</vmext:ServiceLinks>
</vmext:Service>

```

The server registers the service and returns a Service element that includes information derived from the contents you POSTed, and a set of Link elements that you can use to access, remove, disable, or modify the extension service.

Response:

201 Created

Content Type: application/vnd.vmware.admin.service+xml

...

```

<vmext:Service
  xmlns:vmext="http://www.vmware.com/vcloud/extension/v1.5"
  xmlns:vcloud="http://www.vmware.com/vcloud/v1.5"
  name="SDK-BackupExtension"
  id="urn:vcloud:externalService:45"
  type="application/vnd.vmware.admin.service+xml"
  href="https://vcloud.example.com/api/admin/extension/service/45"
  ... >
  <vcloud:Link
    rel="remove"
    href="https://vcloud.example.com/api/admin/extension/service/45" />
  <vcloud:Link
    rel="edit"
    type="application/vnd.vmware.admin.service+xml"
    href="https://vcloud.example.com/api/admin/extension/service/45" />
  <vcloud:Link
    rel="rights"
    type="application/vnd.vmware.admin.rights+xml"
    href="https://vcloud.example.com/api/admin/extension/service/45/rights" />
  <vcloud:Link
    rel="down:serviceLinks"
    type="application/vnd.vmware.vcloud.query.records+xml"
    href="https://vcloud.example.com/api/admin/extension/service/45/links" />
  <vcloud:Link
    rel="bundle:upload"
    type="application/vnd.vmware.admin.bundleUploadParams+xml"
    href="https://vcloud.example.com/api/admin/extension/service/localizationbundles" />
  <vcloud:Link
    rel="add"
    type="application/vnd.vmware.admin.serviceLink+xml"
    href="https://vcloud.example.com/api/admin/extension/service/45/links" />
  <vcloud:Link
    rel="down:apiFilters"

```

```

    type="application/vnd.vmware.vcloud.query.records+xml"
    href="https://vcloud.example.com/api/admin/extension/service/45/apifilters" />
<vcloud:Link
  rel="add"
  type="application/vnd.vmware.admin.apiFilter+xml"
  href="https://vcloud.example.com/api/admin/extension/service/45/apifilters" />
<vcloud:Link
  rel="add"
  type="application/vnd.vmware.admin.apiDefinition+xml"
  href="https://vcloud.example.com/api/admin/extension/service/45/apidefinitions" />
<vcloud:Link
  rel="down:apiDefinitions"
  type="application/vnd.vmware.vcloud.query.records+xml"
  href="https://vcloud.example.com/api/admin/extension/service/45/apidefinitions" />
<vcloud:Link
  rel="add"
  type="application/vnd.vmware.admin.resourceClass+xml"
  href="https://vcloud.example.com/api/admin/extension/service/45/resourceclasses" />
<vcloud:Link
  rel="down:resourceClasses"
  type="application/vnd.vmware.vcloud.query.records+xml"
  href="https://vcloud.example.com/api/admin/extension/service/45/resourceclasses" />
<vcloud:Link
  rel="authorization:check"
  type="application/vnd.vmware.admin.authorizationCheckParams+xml"
  href="https://vcloud.example.com/api/admin/extension/service/45/authorizationcheck" />
  ...
</vmext:Service>

```

The following elements are never returned as part of a response body. Instead, they are returned as Link elements in the body of their container.

- AclRules
- ApiDefinitions and Files
- ApiFilters
- ResourceClassActions
- ResourceClasses
- ServiceLinks
- ServiceResources

For example, to retrieve the contents of the ServiceLinks element POSTed with the request body in this example, GET the URL in this Link:

```

<vcloud:Link
  rel="down:serviceLinks"
  type="application/vnd.vmware.vcloud.query.records+xml"
  href="https://vcloud.example.com/api/admin/extension/service/45/links" />

```

Adding or Removing Service Links

A service can add its own `Link` elements to the representations of vCloud API objects. You can create these service links when you register a service. You can also add or remove the links after you register the service.

You can create multiple service links as part of registering a service. After you register a service, you can add or remove individual links. Service links typically appear in the representations of all objects of a specific type, but you can constrain them to appear in a particular object of that type. Service links are not included in object representations unless the service that created them is enabled.

NOTE You cannot update a service link, but you can remove existing links and create links.

Prerequisites

Verify that you are logged in to the vCloud API as a system administrator.

Add a Service Link

You can add a service link to an existing service.

A `ServiceLink` element must contain the following child elements:

LinkHref	The value of <code>href</code> attribute of the <code>Link</code> . This can be any URI, and can include the variables <code>{baseUri}</code> and <code>{resourceId}</code> . When constructing the <code>href</code> value of the <code>Link</code> , vCloud Director replaces <code>{baseUri}</code> with the vCloud Director REST API base URL, and replaces <code>{resourceId}</code> with the UUID portion of the <code>id</code> attribute value of the resource in which the <code>Link</code> is inserted. The following example might expand to the string <code>https://vcloud.example.com/org/17</code> .
	<pre><LinkHref> {baseUri}/org/{resourceId} </LinkHref></pre>
MimeType	The value, specified as a MIME content type, of the <code>type</code> attribute of the <code>Link</code> .
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 <code>href</code> value as a request URL.
ResourceType	The object type, specified as a MIME content type, of the object in which the <code>Link</code> appears.

NOTE You can constrain the `Link` to appear in a specific resource by including a `ResourceId` element in the `ServiceLink`. This element contains the `id` of the resource in which the `Link` will appear. This resource must be of the type specified in the `ResourceType` element of the `ServiceLink`.

Prerequisites

Verify that you are logged in to the vCloud API as a system administrator.

Procedure

- 1 Retrieve the XML representation of the service.

This request retrieves the XML representation of the service created in [“Example: Register an Extension Service,”](#) on page 285:

```
GET https://vcloud.example.com/api/admin/extension/service/45
```


- 2 Examine the response to find the Link for adding service links.

This Link has the following form:

```
<vcloud:Link
  rel="add"
  type="application/vnd.vmware.admin.serviceLink+xml"
  href="https://vcloud.example.com/api/admin/extension/service/44/links" />
```

- 3 Create a ServiceLink element that specifies the properties of the new link.
- 4 POST the ServiceLink element to the URL described in [Step 2](#)

See [“Example: Add a Service Link,”](#) on page 289.

Example: Add a Service Link

This request adds a ServiceLink to the service created in [“Example: Register an Extension Service,”](#) on page 285

Request:

POST https://vcloud.example.com/api/admin/extension/service/45/links

Content-type: application/vnd.vmware.admin.serviceLink+xml

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

```
<vmext:ServiceLink
  xmlns:vmext="http://www.vmware.com/vcloud/extension/v1.5"
  xmlns:vcloud="http://www.vmware.com/vcloud/v1.5"
  type="application/vnd.vmware.admin.serviceLink+xml">
  <vmext:LinkHref>{baseUri}vApp/vapp-{resourceId}/action/deleteBackup</vmext:LinkHref>
  <vmext:MimeType>application/vnd.example.vcd-backup.createBackupParams+xml</vmext:MimeType>
  <vmext:Rel>deleteBackup</vmext:Rel>
  <vmext:ResourceType>application/vnd.example.vcd-backup.vApp+xml</vmext:ResourceType>
</vmext:ServiceLink>
```

Response:

200 OK

Content-type: application/vnd.vmware.admin.serviceLink+xml

...

```
<vmext:ServiceLink ...>
```

...

```
</vmext:ServiceLink>
```

Delete a Service Link

Delete a service link when you no longer want it to appear in the representation of an vCloud API objects, or when you want to replace it with a new service link.

When you retrieve the list of service links associated with a service, the response is a QueryResultRecords element in which each service link is represented as a ServiceLinkRecord element. The value of the href attribute of a ServiceLinkRecord is a URL you can use to retrieve or delete the service link.

Prerequisites

Verify that you are logged in to the vCloud API as a system administrator.

Procedure

- 1 Retrieve the XML representation of the service.

This request retrieves the XML representation of the service created in [“Example: Register an Extension Service,”](#) on page 285:

```
GET https://vcloud.example.com/api/admin/extension/service/45
```

- 2 Examine the response to find the Link for listing service links.

This Link has the following form:

```
<vcloud:Link
  rel="down:serviceLinks"
  type="application/vnd.vmware.vcloud.query.records+xml"
  href="https://vcloud.example.com/api/admin/extension/service/44/links" />
```

- 3 Make a GET request to the link described in [Step 2](#).
- 4 Examine the response to find the ServiceLinkRecord that represents the service link to delete.
- 5 Make a DELETE request to the URL in the href attribute value of that ServiceLinkRecord.

Example: Delete a Service Link

Start by getting the service's list of service links.

Request:

```
GET https://vcloud.example.com/api/admin/extension/service/45
```

Response:

```
<?xml version="1.0" encoding="UTF-8"?>
<QueryResultRecords
  total="3"
  pageSize="25"
  page="1"
  name="serviceLink"
  type="application/vnd.vmware.vcloud.query.records+xml"
  ... >
  ...
  <ServiceLinkRecord
    service="https://vcloud.example.com/api/admin/extension/service/45"
    resourceType="application/vnd.vmware.vcloud.vApp+xml"
    rel="down"
    mimeType="application/vnd.vmware.vcloud.backupList+xml"
    linkHref="{baseUri}api/vApp/vapp-{resourceId}/backups"
    href="https://vcloud.example.com/api/admin/extension/service/link/01ca..." />
  <ServiceLinkRecord
    service="https://vcloud.example.com/api/admin/extension/service/45"
    resourceType="application/vnd.vmware.vcloud.org+xml"
    rel="down"
    mimeType="application/vnd.vmware.vcloud.backupList+xml"
    linkHref="{baseUri}org/{resourceId}"
    href="https://vcloud.example.com/api/admin/extension/service/link/a624..." />
  <ServiceLinkRecord
    service="https://vcloud.example.com/api/admin/extension/service/45"
    resourceType="application/vnd.vmware.vcloud.vApp+xml"
    rel="backup"
```

```

    mimeType="application/vnd.vmware.vcloud.createBackupParams+xml"
    linkHref="{baseUri}vApp/vapp-{resourceId}/action/backup"
    href="https://vcloud.example.com/api/admin/extension/service/link/f62e..." />
</QueryResultRecords>

```

NOTE Link id values are truncated in this example.

Using this information, find the `ServiceLinkRecord` that represents the service link you want to delete, and make a DELETE request to that URL.

DELETE `https://vcloud.example.com/api/admin/extension/service/link/f62e...`

Service-Specific Tasks and Events

An extension service can create Task objects in a vCloud Director organization, and can post events to the organization's event stream.

Tasks and events are created in the context of an organization. Each task or event is associated with exactly one user, who must be a system administrator or a member of the organization in which the task or event is created. Tasks and events can also have an owner, which is a reference to the subject of the task or event (for example, an object being created or updated by a task).

Task and Event Workflow

vCloud Director native services typically create Task objects to track the progress of asynchronous events. These objects are returned to clients as Task elements, which can be embedded in a container that represents an object under construction, or simply returned in Task form.

Tasks are also reported in the organization's event stream. A service can also add its own events to this stream, in addition to the ones added as a side-effect of creating a task.

Extension services act as vCloud API clients when creating tasks and events, even if those tasks and events are created to track service-specific objects or operations. After a task is created in vCloud Director, an extension service can use its AMQP connection to vCloud Director to return a Task as the response, or part of the response, to a client request.

Localizing Task and Event Message Content

Message strings included in tasks and events can be localized. See [“Localization Framework for Extension Services,”](#) on page 302.

Create or Update a Service-Specific Task

When a user requests an asynchronous operation from an extension service, the service can create a task object and add it to an organization's tasks list.

Every vCloud Director organization has a tasks list and accepts requests to add a task to the list. When a client requests an asynchronous operation from a service, the service starts to process the request and also POSTs a Task element to the organization's `tasksList` URL. vCloud Director adds information such as an `id` and `startTime` to the Task, places it on the organization's `TasksList`, creates an event in the organization's event stream, and returns the Task to the service. The service can then send the Task, as an AMQP message, to vCloud Director, which sends it as a response to the client that made the original request.

NOTE Because of the diversity of sources from which an extension service can draw references to the `User`, `Owner`, and `Organization` elements of a Task, it may not always be possible for every client to resolve such references. For example, if a service creates an object in an organization of which you are not a member, you will not be able to resolve the reference to the object in the `Owner` element of the Task.

Prerequisites

Verify that you are logged in to the vCloud API as a system administrator.

Procedure

- 1 Retrieve the XML representation of the organization in which you want to create the Task.

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 tasks to the organization's tasks list.

This element has a rel attribute value of task:create and a type attribute value of application/vnd.vmware.vcloud.task+xml, as shown here:

```
<Link
  rel="task:create"
  type="application/vnd.vmware.vcloud.task"
  href="https://vcloud.example.com/api/tasksList/26" />
```

- 3 Create a Task element that specifies the details of the task.
- 4 POST the Task element to the organization's tasksList URL.

The server creates a task object and adds it to the organization's tasks list, and returns the representation of the object to the service. To return the XML representation of the task object to the client that made the original request, the service must create a JSON representation of the Task and return it to vCloud Director AMQP service.

Example: Add a Task to an Organization's Tasks List

Request:

```
POST https://vcloud.example.com/api/tasksList/26
Content-Type: application/vnd.vmware.vcloud.task+xml
...
<?xml version="1.0" encoding="UTF-8"?>
<Task
  xmlns="http://www.vmware.com/vcloud/v1.5"
  status="running"
  serviceNamespace="org.example.vcd.backup"
  type="application/vnd.vmware.vcloud.task+xml"
  name="task">
  <User
    type="application/vnd.vmware.admin.user+xml"
    name="administrator"
    href="https://vcloud.example.com/api/admin/user/85" />
  </Task>
```

Response:

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

```

    operationName="jobCustom"
    operation="External processing" >
    ...
</Task>

```

Create a Service-Specific Event

An extension service can request that vCloud Director add an event message to the event stream of an organization.

The system always creates an event message when a service posts a Task to an organization's tasks list. To create additional event messages, a service can POST an Event element to an organization's events URL.

Prerequisites

Verify that you are logged in to the vCloud API as a system administrator.

Procedure

- 1 Retrieve the XML representation of the organization in which you want to create the event.

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 events to the organization's events stream.

This element has a rel attribute value of event:create and a type attribute value of application/vnd.vmware.vcloud.event+xml, as the following example shows:

```

<Link
  rel="event:create"
  type="application/vnd.vmware.vcloud.event"
  href="https://vcloud.example.com/api/admin/org/26/events" />

```

- 3 Create an Event element that specifies the details of the task.
- 4 POST the Event element to the organization's events URL.

Example: Add an Event to an Organization's Event Stream

Request:

```

POST https://vcloud.example.com/api/admin/org/26/events
Content-Type: application/vnd.vmware.admin.event+xml
...
<?xml version="1.0" encoding="UTF-8"?>
<Event
  success="true"
  serviceNamespace="org.example.vcd.backup"
  type="backupComplete">
  <User
    type="application/vnd.vmware.admin.user+xml"
    name="administrator"
    href="https://vcloud.example.com/api/admin/user/85" />
  </Event>

```

Response:

```
204 No Content
```

Authorization Framework for Extension Service Operations

All requests to extension services must be authenticated through the vCloud API. Extension services can participate in vCloud API REST authorization by controlling access to their objects and operations through new or existing rights and roles.

An extension service that does not enable the use of vCloud Director REST authorization implicitly grants permission for all users to perform all operations that the service uses. A service can use the native vCloud Director REST authorization model by taking the following steps:

- 1 Define resource classes that represent references to service-specific object types.
- 2 Define resource class actions that specify the actions that are implemented for those object types.
- 3 Define ACL rules specifying the rights required to perform an operation on objects of a specific type.

Participation in the Authorization Framework

To participate in the authorization framework, a service must include an `AuthorizationEnabled` element with a value of `true` in its registration request.

```
<vmext:AuthorizationEnabled>true</vmext:AuthorizationEnabled>
```

It must also define at least one resource class, specify at least one action for that class, and define an ACL rule that constrains use of the action on the class.

Resource Classes and Actions

A service uses the following constructs to define the objects, operations, and permissions that constitute its authorization model.

Resource Classes	Set of rules for creating references to service-specific objects. Like other object references in the vCloud API, resource classes are a <code>Link</code> element that specifies the MIME type of the resource and includes an <code>href,URL</code> , that can be used to retrieve the resource. The rules include a MIME type, a URL pattern, and a template for creating an <code>id</code> attribute value in URN form.
Resource Class Actions	Combination of a URL pattern that specifies a resource class and an HTTP method that implements an action on a resource of that class. The action uses the specified method in a request to a URL that matches the specified pattern.
ACL Rules	Specifies the rights that an organization or user have to an operation defined as a resource class action.

Querying for Organization and User Rights

The vCloud API query service implements several queries that return a list of rights that a specified user or organization is granted. A user can make a request that specifies one or more entity references and returns a summary of user rights to the specified entities.

Create an Extension Service Resource Class

To configure your extension service to provide access control for the objects it creates, define a resource class for each of its object types.

A `ResourceClass` element contains the information needed to construct a URL that a client can use to access the resource in a specific context. It must contain the following child elements:

MimeType	The MIME content type of all instances of the resource class.
UrlTemplate	The value of href attribute value for resources of this class. This can be any URI, and can include the variables {baseUri} and {resourceId}. When constructing the href value, vCloud Director replaces {baseUri} with the vCloud Director REST API base URL, and replaces {resourceId} with the UUID portion of the id attribute value of the resource.
Nid	The Namespace Identifier for resources of this type, as specified in http://www.ietf.org/rfc/rfc2141.txt .
UrnPattern	The Namespace Specific String for resources of this type, as specified in http://www.ietf.org/rfc/rfc2141.txt . You can provide a string or a named regular expression, where (?<id>) matches the resource identifier.

Prerequisites

Verify that you are logged in to the vCloud API as a system administrator.

Procedure

- 1 Retrieve the XML representation of the service.

Use a request like this one:

```
GET https://vcloud.example.com/api/admin/extension/service/45
```

- 2 Examine the response to find the Link for adding resource classes.

This Link has the following form:

```
<vcloud:Link
  rel="add"
  type="application/vnd.vmware.admin.resourceClass+xml"
  href="https://vcloud.example.com/api/admin/extension/service/45/resourceclasses" />
```

- 3 Construct a `ResourceClass` element.

See the request portion of “[Example: Create an Extension Service Resource Class](#),” on page 295 for information about the contents of this element.

- 4 POST the `ResourceClass` element to the URL described in [Step 2](#).

Example: Create an Extension Service Resource Class

This request defines a resource class named `Backup`.

- The `MimeType` is specified using the standard form for vnd type names.
- The `UrlTemplate` uses the {baseUri} and {resourceId} variables, and could expand to a URL like `https://vcloud.example.com/backup/27`
- The `Nid` and `UrnPattern` elements provide rules for constructing an URN of the form:

```
urn:vcloud:backup: id
```

as shown in the response.

Request:

POST https://vcloud.example.com/api/admin/extension/service/45/resourceclasses

Content-type: application/vnd.vmware.admin.resourceClass+xml

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

```
<vmext:ResourceClass
  name="Backup"
  xmlns:vmext="http://www.vmware.com/vcloud/extension/v1.5"
  xmlns:vcloud="http://www.vmware.com/vcloud/v1.5">
  <vmext:MimeType>application/vnd.vmware.vcloud.backup+xml</vmext:MimeType>
  <vmext:UrlTemplate>{baseUri}backup/{resourceId}</vmext:UrlTemplate>
  <vmext:Nid>vcloud</vmext:Nid>
  <vmext:UrnPattern>^backup(?&lt;id&gt;[0-9]*)</vmext:UrnPattern>
</vmext:ResourceClass>
```

Response:

201 Created

Content-Type: application/vnd.vmware.admin.resourceClass+xml

...

```
<vmext:ResourceClass
  name="Backup"
  xmlns:vmext="http://www.vmware.com/vcloud/extension/v1.5"
  xmlns:vcloud="http://www.vmware.com/vcloud/v1.5"
  id="urn:vcloud:backup:83"
  type="application/vnd.vmware.admin.resourceClass+xml"
  href="https://vcloud.example.com/api/admin/extension/service/resourceclass/83"
  ...>
  <vcloud:Link
    rel="remove"
    href="https://vcloud.example.com/api/admin/extension/service/resourceclass/83" />
  <vcloud:Link
    rel="up"
    type="application/vnd.vmware.admin.service+xml"
    href="https://vcloud.example.com/api/admin/extension/service/45" />
  <vcloud:Link
    rel="down:resourceClassActions"
    type="application/vnd.vmware.vcloud.query.records+xml"

href="https://vcloud.example.com/api/admin/extension/service/resourceclass/83/resourceclassactions" />
  <vcloud:Link
    rel="down:serviceResources"
    type="application/vnd.vmware.vcloud.query.records+xml"

href="https://vcloud.example.com/api/admin/extension/service/resourceclass/83/serviceResources" />
  <vmext:MimeType>BackupType+xml</vmext:MimeType>
  <vmext:UrlTemplate>{baseUri}backup/{resourceId}</vmext:UrlTemplate>
  <vmext:Nid>nidBackup</vmext:Nid>
  <vmext:UrnPattern>^myNssBackup(?&lt;id&gt;[0-9]*)</vmext:UrnPattern>
</vmext:ResourceClass>
```


Define an Action for a Resource Class

After you define a resource class, you can specify the actions that are permitted on resources of that class.

A `ResourceClassAction` object defines an HTTP method that is allowed on a specific `UrlPattern`.

Prerequisites

Verify that you are logged in to the vCloud API as a system administrator.

Procedure

- 1 Retrieve the XML representation of the service.

Use a request like this one:

```
GET https://vcloud.example.com/api/admin/extension/service/45
```

- 2 Examine the response to find the Link for specifying resource class actions.

This Link has the following form:

```
<vcloud:Link
  rel="add"
  type="application/vnd.vmware.admin.resourceClassAction+xml"
  href="https://vcloud.example.com/api/admin/extension/service/45/resourceclassactions" />
```

- 3 Construct a `ResourceClassAction` element.

See the request portion of [“Example: Define an Action for a Resource Class,”](#) on page 297 for information about the contents of this element.

- 4 POST the `ResourceClassAction` element to the URL described in [Step 2](#).

Example: Define an Action for a Resource Class

This request specifies a GET action for a URL matching the pattern could expand to a URL like `https://vcloud.example.com/backup/27`.

Request:

```
POST https://vcloud.example.com/api/admin/extension/service/45/resourceclassactions
```

```
Content-type: application/vnd.vmware.admin.resourceClassAction+xml
```

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

```
<vmext:ResourceClassAction
```

```
  xmlns:vmext="http://www.vmware.com/vcloud/extension/v1.5"
```

```
  xmlns:vcloud="http://www.vmware.com/vcloud/v1.5"
```

```
  name="Read backups">
```

```
    <vmext:HttpMethod>GET</vmext:HttpMethod>
```

```
    <vmext:UrlPattern>/api/backup/(?&lt;id&gt;[-,a-g,0-9]*)</vmext:UrlPattern>
```

```
</vmext:ResourceClassAction>
```

The response is a `ResourceClassAction` element that includes information derived from the contents you POSTed, along with a set of Link elements that you can use to remove the `ResourceClassAction` or add ACL rules that control access to the resource class via the action.

Response:

```
201 Created
```

```
Content-Type: application/vnd.vmware.admin.resourceClassAction+xml
```

```
...
```

```
<vmext:ResourceClassAction
```

```
  xmlns:vmext="http://www.vmware.com/vcloud/extension/v1.5"
```

```

xmlns:vcloud="http://www.vmware.com/vcloud/v1.5"
name="Read backups"
id="urn:vcloud:resourceClassAction:268"
type="application/vnd.vmware.admin.resourceClassAction+xml"
href="https://vcloud.example.com/api/admin/extension/service/resourceclassaction/268"
... >
<vcloud:Link
  rel="remove"
  href="https://vcloud.example.com/api/admin/extension/service/resourceclassaction/268" />
<vcloud:Link
  rel="up"
  type="application/vnd.vmware.admin.resourceClass+xml"
  href="https://vcloud.example.com/api/admin/extension/service/resourceclass/83" />
<vcloud:Link
  rel="down:aclRules"
  type="application/vnd.vmware.vcloud.query.records+xml"

href="https://vcloud.example.com/api/admin/extension/service/resourceclassaction/268/aclrules" />
<vmext:HttpMethod>GET</vmext:HttpMethod>
<vmext:UriPattern>/api/backup/(?&lt;id&gt;[-,a-g,0-9]*)</vmext:UriPattern>
</vmext:ResourceClassAction>

```

Define an ACL Rule for a Resource Class Action

Permission to execute an extension service operation is controlled by an `ACLRule` contained in the `ResourceClassAction`.

An ACL rule specifies the access controls that apply to a `ResourceClassAction`. Access controls can be defined for any of the following principals:

- an individual user
- a member of a specified organization
- any user whose role includes a specific right
- any resource defined by the service that created the ACL rule

Rights for specific entity types are specified in the following container elements:

ServiceResourceAccess	This specification is optional.
OrganizationAccess	Access for the organizations. This specification is required.
PrincipalAccess	Access control for users, or for any role that includes a specified right. This specification is required.

If the `Access` element in any of these containers has the value `Entity`, the container must also include an `Entity` element that provides a reference to a resource entity, organization, user, or right.

Table 11-2. ACL Rules

Container Element	Access	Comments
ServiceResourceAccess	Shared	The action is authorized for all resources in this resource class
	Entity	The action is authorized for the service resource referenced in the <code>Entity</code> element in this container.
OrganizationAccess	Shared	The action is authorized for all members of the organization that owns the resource.
	Published	The action is authorized for all members of any organization in the cloud.

Table 11-2. ACL Rules (Continued)

Container Element	Access	Comments
	Entity	The action is authorized for members of the organization referenced in the Entity element in this container.
PrincipalAccess	Shared	The action is authorized for all users
	Entity	The action is authorized for the User referenced in the Entity element in this container, or for any role that includes the Right referenced in the Entity element in this container.

A ResourceClassAction can include an arbitrary number of AclRule elements. The action is permitted if the user or resource attempting the action matches any rule.

Prerequisites

Verify that you are logged in to the vCloud API as a system administrator.

Procedure

- 1 Retrieve the XML representation of the resource class action.

This request retrieves the XML representation of the resource class action created in “[Example: Define an Action for a Resource Class](#),” on page 297:

```
GET https://vcloud.example.com/api/admin/extension/service/resourceclassaction/268
```

- 2 Examine the response to find the Link for specifying ACL rules for the resource class action.

This Link has the following form:

```
<vcloud:Link
  rel="add"
  type="application/vnd.vmware.vcloud.query.aclRule+xml"

  href="https://vcloud.example.com/api/admin/extension/service/resourceclassaction/268/aclrules" />
```

- 3 Construct an AclRule element.

See the request portion of “[Example: Define an ACL Rule for a Resource Class Action](#),” on page 299 for information about the contents of this element.

- 4 POST the ResourceClassAction element to the URL described in [Step 2](#).

Example: Define an ACL Rule for a Resource Class Action

This example adds an ACL rule to the resource class action created in “[Example: Define an Action for a Resource Class](#),” on page 297. The rule specifies that all members of a specific organization who have a role that includes a specific right can execute the action.

Request:

```
POST https://vcloud.example.com/api/admin/extension/service/resourceclassaction/268/aclrules
```

```
Content-type: application/vnd.vmware.admin.aclRule+xml
```

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

```
<vmext:AclRule
```

```
  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"
```

```
  name="ACL rule for read backups">
```

```
    <Description>Only users in org/26 who have right/2 can read backups</Description>
```

```
    <vmext:ServiceResourceAccess>
```

```

    <vmext:Access>Shared</vmext:Access>
  </vmext:ServiceResourceAccess>
<vmext:OrganizationAccess>
  <vmext:Access>Entity</vmext:Access>
  <vmext:Entity
    xsi:type="vcloud:ResourceReferenceType"
    type="application/vnd.vmware.admin.org+xml"
    href="https://vcloud.example.com/api/admin/org/26" />
</vmext:OrganizationAccess>
<vmext:PrincipalAccess>
  <vmext:Access>Entity</vmext:Access>
  <vmext:Entity
    xsi:type="vcloud:ResourceReferenceType"
    type="application/vnd.vmware.admin.right+xml"
    href="https://vcloud.example.com/api/admin/right/2" />
</vmext:PrincipalAccess>
</vmext:AclRule>

```

The response contains information supplied in the request, along with several Link elements created by the server.

Response:

201 Created

Content-Type: application/vnd.vmware.admin.aclrule+xml

...

```

<vmext:AclRule
  xmlns:vmext="http://www.vmware.com/vcloud/extension/v1.5"
  xmlns:vcloud="http://www.vmware.com/vcloud/v1.5"
  name="ACL rule for read backups"
  id="urn:vcloud:aclRule:5"
  type="application/vnd.vmware.admin.aclRule+xml"
  href="https://vcloud.example.com/api/admin/extension/service/aclrule/5">
  <Description>Only users in org/26 who have right/2 can read backups</Description><vcloud:Link
    <vcloud:Link
      rel="remove"
      href="https://vcloud.example.com/api/admin/extension/service/resourceclassaction/268" />
    ...
  </vcloud:Link>
</vmext:AclRule>

```

Create a Service-Specific Right

A service can create rights that apply to its operations. You can add these rights to existing roles or new roles.

In the vCloud API, a right is simply a name that a service attaches to a privilege. When a service specifies an ACL rule for a resource class action, the rule can reference a right. A user who is assigned a role that includes the right is authorized to take the specified action.

Prerequisites

Verify that you are logged in to the vCloud API as a system administrator.

Procedure

- 1 Retrieve the XML representation of the service.

Use a request like this one:

```
GET https://vcloud.example.com/api/admin/extension/service/45
```

- 2 Examine the response to find the Link for adding or listing service-specific rights

This Link has the following form:

```
<vcloud:Link
  rel="rights"
  type="application/vnd.vmware.admin.rights+xml"
  href="https://vcloud.example.com/api/admin/extension/service/45/rights" />
```

- 3 Construct a Right element.

For information about the contents of this element, see the request portion of [“Example: Create a Service-Specific Right,”](#) on page 301.

- 4 POST the Right element to the URL described in [Step 2](#).

Example: Create a Service-Specific Right

This request creates a right named DeleteBackup. The name attribute and Category element are required, and can have any string value. Include a BundleKey if any messages associated with the right appear in a localization bundle.

Request:

```
POST https://vcloud.example.com/api/admin/extension/service/45/rights
Content-type: application/vnd.vmware.admin.right+xml
<?xml version="1.0" encoding="UTF-8"?>
<Right
  xmlns="http://www.vmware.com/vcloud/v1.5"
  name="DeleteBackup">
  <Description>Right to remove a backup object</Description>
  <Category>VcdBackup</Category>
  <BundleKey>BackupBundle</BundleKey>
</Right>
```

The response is a Right element that includes information derived from the contents you POSTed. The service namespace name is prepended to the name of the right.

Response:

```
201 Created
Content-Type: application/vnd.vmware.admin.right+xml
...
<Right
  xmlns="http://www.vmware.com/vcloud/v1.5"
  name="{org.example.vcd.backup}:DeleteBackup"
  id="urn:vcloud:right:99"
  type="application/vnd.vmware.admin.right+xml"
  href="https://vcloud.example.com/api/admin/right/99"
  ... >
  <Description>Right to remove a backup object</Description>
  <Category>VcdBackup</Category>
  <BundleKey>BackupBundle</BundleKey>
</Right>
```

Localization Framework for Extension Services

Extension service developers can provide localized service-specific messages by creating and uploading a localization bundle.

An extension service localization bundle is a file in zip format that contains one or more properties files. Each properties file consists of an arbitrary number of *key=value* pairs, where the *key* is the name of a service-specific task operation or event, and the *value* is the localized message.

REST APIs for Extension Services

A simple extension service does not need a REST API. You can define a service-specific REST API endpoint and one or more schema definition files.

An extension service that does not require request or response bodies other than those that the vCloud API defines, *Task*, for example, can simply define the URL patterns that constitute its API filters and the service links that implement its operations.

A service that defines its own request or response bodies must also specify a URL to which clients can direct requests. The service must specify locations of the files, such as XML schema definition (XSD) files, to which its clients require access.

Create an API Filter for an Extension Service

When you register an extension service with vCloud Director, you specify one or more API filters, which are URL patterns or MIME content types that the vCloud Director REST service should treat as extension requests. You cannot update the API filter for a registered service, but you can replace it with a new one.

An API filter can be either a URL pattern, typically in the form of a regular expression, or a content type, typically in the form of a MIME content-type string. Requests whose URL matches the specified `urlPattern` are sent to the service that has registered the filter. An API filter that specifies `responseContentType` is applied only to responses whose `Content-type` attribute has a value that matches the specified `responseContentType`. An extension service that receives such a response must return it, after making any service-specific modifications, to the AMQP service as a JSON message, so that it can be returned to the vCloud Director client that made the request.

Prerequisites

Verify that you are logged in to the vCloud API as a system administrator.

Procedure

- 1 Retrieve the XML representation of the service.

This request retrieves the XML representation of the service created in [“Example: Register an Extension Service,”](#) on page 285:

```
GET https://vcloud.example.com/api/admin/extension/service/45
```

- 2 Examine the response to find the Link for adding API filters

This Link has the following form:

```
<vcloud:Link
  rel="add"
  type="application/vnd.vmware.admin.apiFilter+xml"
  href="https://vcloud.example.com/api/admin/extension/service/45/apifilters" />
```

- 3 Construct an `ApiFilter` element.

This `ApiFilter` overwrites any existing `ApiFilter` defined by the service. See the request portion of [“Example: Create an API Filter for an Extension Service,”](#) on page 303 for information about the contents of this element.

- 4 POST the `ApiFilter` element to the URL described in [Step 2](#).

Example: Create an API Filter for an Extension Service

This request adds a new `UrlPattern` to set of patterns defined in the request portion of [“Example: Register an Extension Service,”](#) on page 285. The request replaces the existing set of filter expressions with a new one that includes the original set and one additional expression.

Request:

```
POST https://vcloud.example.com/api/admin/extension/service/45/apifilters
Content-type:application/vnd.vmware.admin.apifilter+xml
<?xml version="1.0" encoding="UTF-8"?>
<vmext:ApiFilter>
  <vmext:UrlPattern>(/api/org/.*/backups)|(/api/vApp/vapp-.*/backups)|
(/api/vApp/vapp-.*/action/backup)|(/api/backup/.*) |
(/api/vApp/vapp-.*/action/recoverBackup)</vmext:UrlPattern>
</vmext:ApiFilter>
```

Response:

```
201 Created
Content-Type: application/vnd.vmware.admin.resourceClass+xml
...
<vmext:ApiFilter>
  <vmext:UrlPattern>(/api/org/.*/backups)|(/api/vApp/vapp-.*/backups)|
(/api/vApp/vapp-.*/action/backup)|(/api/backup/.*) |
(/api/vApp/vapp-.*/action/recoverBackup)</vmext:UrlPattern>
</vmext:ApiFilter>
```

Create or Update an Extension Service API Definition

You can define an API for an extension service when you register the service. You can also create or update the service API definition later.

An API definition for an extension service includes an API entry point (A URL at which a client can send requests to the API) and a set of file descriptors, each of which consists of a description and a reference to a schema definition file.

An `ApiDefinition` element must contain the following child elements:

EntryPoint	The URL to which a client can send requests to the service API. This can be any URL, and can include the variable <code>{baseUri}</code> , which represents the vCloud Director REST API base URL.
Namespace	The service namespace. See “Register an Extension Service,” on page 284
Files	One or more references to schema definition files. The references must be accessible to vCloud Director.

Prerequisites

Verify that you are logged in to the vCloud API as a system administrator.

Procedure

- 1 Retrieve the XML representation of the service.

Use a request like this one:

```
GET https://vcloud.example.com/api/admin/extension/service/45
```

- 2 Examine the response to find the Link for adding API definitions.

This Link has the following form:

```
<vcloud:Link
  rel="add"
  type="application/vnd.vmware.admin.apiDefinition+xml"
  href="https://vcloud.example.com/api/admin/extension/service/45/apidefinitions" />
```

- 3 Construct an ApiDefinition element.

For information about the contents of this element, see the request portion of [“Example: Create an Extension Service API Definition,”](#) on page 304.

- 4 POST the ApiDefinition element to the URL described in [Step 2](#).

Example: Create an Extension Service API Definition

This request defines an API for a backup service. The definition includes two FileDescriptor elements that reference files available on the vendor's public Website. The endpoint for requests to the service is the vCloud Director API login URL.

Request:

```
POST https://vcloud.example.com/api/admin/extension/service/45/apidefinitions
```

```
Content-type:application/vnd.vmware.admin.apiDefinition+xml
```

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

```
<vmext:ApiDefinition
```

```
  name="Backup service version 5.1">
```

```
  <Description>Backup service API</Description>
```

```
  <vmext:EntryPoint>{baseUri}/login</vmext:EntryPoint>
```

```
  <Namespace>org.example.vcd.backup</Namespace>
```

```
  <vmext:Files>
```

```
    <vmext:FileDescriptor>
```

```
      <vmext:Description>Master schema definition file.</vmext:Description>
```

```
      <vmext:File
```

```
        href="http://example.com/backp/schema/v5.1/master.xsd" />
```

```
    </vmext:FileDescriptor>
```

```
    <vmext:FileDescriptor>
```

```
      <vmext:Description>Schema definition file for backup devices.</vmext:Description>
```

```
      <vmext:File
```



```
        href="http://example.com/vcdBackup/schema/v5.1/devices.xsd" />
    </vmext:FileDescriptor>
</vmext:Files>
</vmext:ApiDefinition>
```

Response:

201 Created

Content-type:application/vnd.vmware.admin.apiDefinition+xml

...

```
<vmext:ApiDefinition
  name="Backup service version 5.1">
  ...
</vmext:ApiDefinition>
```


XML Representations in the vCloud API

12

The vCloud API represents objects in a cloud as XML documents in which object properties are contained in elements and attributes that have 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,”](#) on page 21.

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 309.

Date and Time Values

Values of type `xs:dateTime` are always interpreted as UTC if a timezone has not been explicitly specified.

Length Limits on Element and Attribute String Values

String values for the `name` attribute, `Description` element, and `ComputerName` have length limitations that depend on the object to which they are attached.

Table 12-1. Length Limits on Element and Attribute String Values

Object	Element or Attribute Name	Maximum Length in Characters
Vdc	name	256
Vdc	Description	256
VApp	name	128
VApp	Description	256

Table 12-1. Length Limits on Element and Attribute String Values (Continued)

Object	Element or Attribute Name	Maximum Length in Characters
VAppTemplate	name	128
VAppTemplate	Description	256
Vm	name	128
Vm	ComputerName	15 on Windows, 63 on all other platforms
Catalog	name	128
Catalog	Description	256
Media	name	128
Media	Description	256

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.

NOTE Starting with API version 1.5, requests to all versions of the vCloud API are processed in the `http://www.vmware.com/vcloud/v1.5` XML namespace. Treatment of version-specific elements and attributes in requests is controlled by the value of the `Accept` header. For example, this `Accept` header specifies that the request body is presumed to be valid for vCloud API version 5.1 and a version 5.1 response is expected:

```
Accept: application/*+xml;version=5.1
```

Requests are validated against the elements and attributes defined in the specified version. Responses are filtered to remove elements and attributes that are not defined in the specified version.

The *vCloud API Schema Reference* indicates the deprecation status of elements and attributes, and also indicates when each element or attribute was added to the API. See [“About the Schema Reference,”](#) on page 21.

Extensibility

The vCloud API provides complete programmatic access to the vCloud Director Extension Services facility described at [Chapter 11, “vCloud Director Extension Services,”](#) on page 281.

In addition, there is a more general extensibility mechanism, `VCloudExtension`, that clients are free to use. `VCloudExtensionType` is an abstract type that all complex types defined in the vCloud API namespace extend. It can contain an arbitrary number of elements and attributes, and provides a way for you to add custom attributes and elements to any type.

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 309
- [“Common vCloud API Attributes,”](#) on page 310
- [“Retrieve an Object as an Entity,”](#) on page 312

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 12-2. 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 <code>ovf:Environment</code> element.
xmlns:ovf	http://schemas.dmtf.org/ovf/envelope/1	Required in request bodies that include elements defined in OVF schema http://schemas.dmtf.org/ovf/envelope/1/dsp8023.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 <code>CIM_ResourceAllocationSettingData.xsd</code> .
xmlns:oe	http://schemas.dmtf.org/ovf/environment/1	Required in request bodies that include elements defined in OVF schema <code>dsp8027_1.1.0.xsd</code> .
xmlns:vssd	http://schemas.dmtf.org/wbem/wscim/1/cim-schema/2/CIM_VirtualSystemSettingData	Not required in request bodies.
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 12-3. 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	Catalog
Disk	None
Network	Container (Organization vDC, 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 <code>id</code> attribute uniquely identifies the object, persists for the life of the object, and is never reused. The <code>id</code> attribute value is intended to provide a context-free identifier that can be used with the vCloud API <code>entityResolver</code> (see “Retrieve an Object as an Entity,” on page 312).
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 <code>id</code> attribute value, it uniquely identifies the object, persists for the life of the object, and is never reused. The value of the <code>href</code> 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 <code>href</code> as an opaque string. The rules that govern how the server constructs <code>href</code> 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 12-4. status Attribute

Value	Description	vAppTemplate	vApp	Vm	Media
-1	The object could not be created.	YES	YES	YES	YES
0	The object is unresolved.	YES	YES	YES	YES
1	The object is resolved.	YES	YES	YES	YES
2	The object is deployed.	No	No	No	No
3	The object is suspended.	No	YES	YES	No
4	The object is powered on.	No	YES	YES	No
5	The object is waiting for user input.	No	YES	YES	No
6	The object is in an unknown state.	YES	YES	YES	No
7	The object is in an unrecognized state.	YES	YES	YES	No
8	The object is powered off.	YES	YES	YES	No
9	The object is in an inconsistent state.	No	YES	YES	No
10	Children do not all have the same status.	YES	YES	No	No
11	Upload initiated, OVF descriptor pending.	YES	No	No	No
12	Upload initiated, copying contents.	YES	No	No	No
13	Upload initiated, disk contents pending.	YES	No	No	No
14	Upload has been quarantined.	YES	No	No	No
15	Upload quarantine period has expired.	YES	No	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 to the vCloud API 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 Using the Integrated Identity Provider,”](#) on page 45. 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 312.

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

A

- administrative tasks, about **129**
- administrator, system **132**
- AMQP
 - about **263**
 - certificate and truststore **233**
- AMQP settings
 - to configure **264**
 - to test **266**
- API client, to develop **21**
- attributes
 - custom **105**
 - name **310**
 - status **310**

B

- blocking task, to configure **263**
- blocking task requests, to monitor **276**
- blocking task settings, to configure **273**
- browsing **41**

C

- catalog
 - adding items **73**
 - change owner **79**
 - controlling access to **191**
 - removing items **75**
 - to administer **176**
 - to change owner **129**
 - to create **129, 176**
 - to delete **129**
 - to find **26**
 - to publish **178**
 - to retrieve **27, 41**
- catalog item, to retrieve **28, 41**
- client, REST **21**
- cloud, administrative view of **49**
- console, displaying **37**

D

- datastore
 - to delete **193**
 - to enable or disable **193**
 - to retrieve or update **193**
- DHCP **160, 166**
- dhcp service, in vApp network **112**
- DHCP service, in Edge Gateway **166**
- disk, independent **75**

- download URL **68**

E

- Edge Gateway
 - about **148**
 - interface capacity **170**
 - syslog server settings **175**
 - to create **151**
 - to create or update **129**
- entity, retrieve object as **12**
- entity resolver, about **312**
- event, service-specific **293**
- examples, conventions for **22**
- extensibility **282**
- extension framework **281**
- extension services
 - ACL rules **298**
 - and events **291**
 - and tasks **291**
 - API definition **303**
 - API filters for **302**
 - authorization frameworks for **294**
 - localization of **302**
 - resource class actions **297**
 - resource classes **295**
 - REST APIs for **302**
 - to register **284**
 - URL patterns for **302**
- external network
 - to create **219**
 - to retrieve or update **193**

F

- firewall service
 - and syslog **175**
 - in Edge Gateway **158**
 - in vApp network **112**

G

- group
 - to create, update, or remove **129**
 - to import **183**
 - to import from SAML **185**
- groups, to administer **179**

H

- host
 - to enable or disable **193**
 - to update or repair **193**

I

- id attribute, and entity resolver **312**
- independent disk
 - about **75**
 - list virtual machines using **56**
 - to attach or detach **83, 93**
 - to create **76**
 - to create or update **56**
 - to remove **78**
- instantiation parameters
 - in instantiateVAppTemplate request **85**
 - sections allowed in **97**
- IP address, of virtual machine NIC **115**
- IP addresses, allocated to network **41**

K

- keytab, to upload **233**

L

- LDAP, certificate and keystore **233**
- Link element, rel attribute **12**
- load balancer service **163**
- logging, of firewall actions **158**
- login
 - integrated identity provider **44**
 - SAML identity provider **46**
- login URL, to obtain **41, 43**
- logout **40**

M

- maintenance mode, vApp **97**
- media
 - cataloging **72**
 - copying or moving **71**
 - retrieve owner **79**
 - to copy, move, or delete **56**
 - to insert or remove virtual **83**
 - to upload **55**
 - to upload or download **56**
 - uploading **69**
- metadata
 - about **239**
 - to retrieve or modify **241**
 - to search for **260**
- metadata value, to retrieve or modify **244**

N

- NAT service
 - in Edge Gateway **160**
 - in vApp network **112**
 - network
 - list IP addresses allocated to **41**
 - to create, update, or remove **129**
 - to reset **129**
 - to retrieve **41**
 - network pool
 - and organization vDC network **148**
 - isolation-backed **225**
 - list of **204**
 - portgroup-backed **227**
 - to create **222**
 - to retrieve or update **193**
 - VLAN-backed **223**
 - VxlanPoolType **209, 222, 225**
 - network services
 - in Edge Gateway **156**
 - in vApp network **112**
 - list of **148**
 - networks
 - about **148**
 - DHCP service **166**
 - NAT services **160**
 - to retrieve list from vCenter **203**
 - VPN **165**
 - notifications
 - format of **267**
 - to enable or disable **264**
- O**
- object hierarchy, diagram of **10**
 - object identifiers **12**
 - object references, about **12**
 - organization
 - retrieve or update settings **136**
 - system **132**
 - to administer **132**
 - to create **133**
 - to create, update, or remove **129**
 - to enable or disable **129, 139**
 - to remove **139**
 - to retrieve administrative view of **129**
 - organization vDC network
 - about **148**
 - direct **167**
 - isolated **173**
 - routed **170**
 - to administer **147**
 - to configure static routes **161**
 - to create **166**
 - to create, update, or remove **129**

- organizations
 - federation of **136**
 - to list **41, 48**
- OVF descriptor, to download **68**
- OVF environment **120**
- OVF package
 - manifest file **59, 63**
 - to upload or download **55**
 - uploading **57**
- OVF specification **81**
- OVF upload
 - initiating **59**
 - to monitor progress of **64**
 - using ranged PUTs **65**

P

- ProductSection element, to retrieve or update **83**
- Provider vDC
 - resource pool set **213**
 - to add or remove storage profiles **216**
 - to create **209**
 - to merge **193, 218**
 - to retrieve or update **193**

Q

- queries
 - packaged **252**
 - typed **248**
- query service
 - about **247**
 - and metadata **260**
 - query parameters **257**
 - query types **248**

R

- requests
 - about **18**
 - headers **18, 307**
 - login **24**
- resource pool
 - adding **214**
 - list of **193**
 - removing **214**
 - to enable or disable **193**
 - to migrate virtual machines to or from **193**
- resource pool set, Provider vDC **213**
- resource pools, to retrieve list from vCenter **201**
- responses, about **20**
- right, to create **300**
- role, to create **187, 189**
- roles and rights **187**

S

- SAML
 - identity provider **46, 179**
 - organization settings for **136**
- schema files
 - accessing **21**
 - retrieving **43**
- schema reference **21**
- SectionType element, to retrieve or update **108**
- service
 - DHCP **166**
 - NAT **160**
- service links
 - required and optional elements **288**
 - to add **288**
 - to add or remove **288**
 - to delete **289**
- Session object, to delete **40**
- snapshot, of vApp **83**
- SSL certificate, to upload **233**
- static routing service
 - in Edge Gateway **161**
 - in vApp network **112**
- status attribute
 - of vApp or vApp template **57**
 - values **310**
- storage profiles
 - compliance check for virtual machine **83**
 - to refresh list of **207**
 - to retrieve list from vCenter **207**
- system settings, to retrieve or update **197**

T

- task
 - blocking **277**
 - service-specific **291**
 - to cancel **129**
 - to retrieve **129**
 - update progress **279**
- task list, to retrieve **129**
- task operations **274**
- timezone, in dateTime values **307**
- truststore, to upload **233**

U

- user
 - to create **129, 180**
 - to import **182**
 - to import from SAML **185**
 - to retrieve **129**
 - to update or remove **129**
- users, to administer **179**

V

vApp

- add virtual machines **90**
- capturing **72**
- changing owner **79**
- composing **87**
- configuration links in **99**
- controlling access to **191**
- datacenter operations **81**
- importing **72**
- importing from vCenter **228**
- list of power operations **83**
- maintenance mode **97**
- recompose **90**
- remove virtual machines **90**
- to change name or description **83**
- to change owner **129**
- to compose or recompose **83**
- to configure **97**
- to delete **38**
- to deploy or undeploy **83**
- to enter or exit maintenance mode **193**
- to import from vCenter as template **193**
- to import from vSphere **193**
- to instantiate **85**
- to modify vApp network configuration **109**
- to operate **96**
- to power off **38**
- to retrieve **34**
- to undeploy **38**
- to view or modify lease settings **100**
- to view or modify network settings **100**
- to view or modify startup settings **100**
- Vm snapshots **83**
- vApp network
 - about **148**
 - to modify **109**
 - to retrieve **148**
 - to view or modify configuration **100**
- vApp snapshot **95**
- vApp template
 - cataloging **72**
 - copying or moving **71**
 - retrieve owner **79**
 - to change name or description **56**
 - to copy, move, or delete **56**
 - to create from OVF **56**
 - to download **56**
 - to download as OVF package **66**
 - to enable for download **66**
 - to enable or disable for download **56**
 - to import virtual machine as **229, 230**

- to instantiate **31, 83**
- to relocate virtual machine from **83**
- to retrieve **41**
- to update **97**
- to upload or download **55**
- upload URL **62**
- uploading vmdk files **63**
- vCenter resources, to discover **200**
- vCenter server
 - to attach **198**
 - to register or unregister **193**
 - to update settings **193**
- vCloud API, and RESTful programming style **9**
- vDC
 - allocation models **140**
 - instantiateVAppTemplate action **30**
 - networks in **30**
 - SupportedHardwareVersion elements **140**
 - to add or remove storage profiles **145**
 - to administer **139**
 - to create **140**
 - to create, update, or remove **129**
 - to enable or disable **129, 147**
 - to find **26**
 - to remove **147**
 - to retrieve **41**
- virtual hardware, vmx versions supported by
 - Provider vDC **209**
- virtual machine
 - available for import **205**
 - CPU configuration **117**
 - disks **122**
 - guest customization for **118**
 - hard disk configuration **124**
 - importing from vCenter **228**
 - network cards **122**
 - question from **92**
 - to attach or detach independent disk **83**
 - to consolidate **83**
 - to import into existing vApp **193**
 - to install VMware tools **83**
 - to relocate **83, 231**
 - to relocate to a new instance of a storage
 - profile **126**
 - to update multiple sections **106**
 - to update NetworkConnectionSection **115**
 - to update storage profile **126**
 - to upgrade hardware version **83**
 - to view or modify CPU properties **103**
 - to view or modify guest customization
 - properties **103**
 - to view or modify memory settings **103**
 - to view or modify network cards **103**

- to view or modify network connection **103**
- to view or modify operating system properties **103**
- to view or modify virtual disks **103**
- Vm
 - configuration links in **101**
 - list of power operations **83**
 - pending question **83**
 - to reboot or reset **83**
 - to retrieve **41**
- vmdk file, to download from template **68**
- VMware Tools
 - to install **83**
 - to retrieve installed version **83**
- vSphere
 - operations **52**
 - retrieve object URL **236**
- vSphere client, to generate URL for use by **193**
- vSphere object, retrieve URL to access directly **193**
- vSphere object map **237**
- vSphere platform, to manage **193**

W

- workflow, example of **23**

X

XML

- compressed responses **18**
 - validation of **18**
- XML namespaces **309**
- XML schemas, reference information **307**

