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
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 13
   Client Workflow Overview 18
   Using the vCloud API with vCloud Director 23
   About the vCloud API Examples 25

2 Hello vCloud: A Simplified RESTful Workflow 27
   Logging In 28
   Find a Catalog and a VDC 30
   Retrieve the Contents of a Catalog 31
   Retrieve a Catalog Item 32
   Retrieve Deployment Information From the VDC 34
   Deploy the vApp 35
   Get Information About a vApp 38
   Displaying the Virtual Machine Console 41
   Undeploy, Power Off, and Delete the vApp 42
   Log Out 44

3 Exploring a Cloud 45
   Summary of vCloud API Browsing Requests 45
   Create a Login Session Using the Integrated Identity Provider 47
   Retrieve a List of Organizations Accessible to You 52
   Retrieve an Administrative View of a Cloud 53
   Retrieve a List of vSphere Platform Operations and Objects for a Cloud 55

4 Provisioning an Organization 57
   Summary of vCloud API Provisioning Requests 58
   Upload an OVF Package to Create a vApp Template 60
   Download a vApp or vApp Template as OVF 70
   Upload a Media Image 74
   Download a Media Image 76
   Capturing and Importing vApps 77
   Managing Catalog Items 78
   Creating and Using Independent Disks 82
   View or Change the Owner of an Object 85
   Controlling Access to vApps and Catalogs 86
5 Deploying and Operating vApps 91
- Summary of vCloud API vApp and Virtual Machine Operations Requests 94
- Create a vApp From a Template 96
- Create a vApp From an OVF Package 104
- Compose a vApp From Existing Virtual Machines 107
- Recompose a vApp to Add or Remove Virtual Machines 110
- Clone a vApp 112
- Capture a vApp as a Template 114
- Update vApp Access Controls 116
- Provide User Input Requested by a Virtual Machine 117
- Attach or Detach an Independent Disk 118
- Creating and Using vApp Snapshots 120
- About Virtual Machine Metrics 121
- Operate a vApp 132
- Configuring vApps and Virtual Machines 133

6 Creating and Managing Organizations 169
- Summary of Administrative Requests 169
- Administrator Credentials and Privileges 172
- Organization Administration 173
- VDC Administration 185
- Network Administration 193
- Catalog Administration 221
- User and Group Administration 243
- Working With Roles and Rights 251

7 Managing and Monitoring a Cloud 261
- Summary of System Administration Requests 261
- Retrieve or Update System Settings 265
- Attach a vCenter Server 266
- Finding Available vCenter Resources 268
- Create a Provider VDC 277
- Create an External Network 287
- Create a Network Pool 290
- Import a Virtual Machine from vCenter 296
- Relocate a Virtual Machine to a Different Datastore 299
- System Truststore and Keytab Maintenance 301
- Retrieve the vSphere URL of an Object 304

8 Working With Object Metadata 307
- Retrieve or Update a Metadata Element 309
- Retrieve or Update a Metadata Value 312

9 Using the Query Service 315
- Typed Queries 316
- Packaged Queries 320
- Query Parameters 325
- Add a Metadata Filter to a Query 329
This edition of the vCloud API Programming Guide provides information about version 5.6 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>
<thead>
<tr>
<th>Revision Date</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>07OCT14</td>
<td>API Version 5.6</td>
</tr>
<tr>
<td>19SEP13</td>
<td>API Version 5.5</td>
</tr>
<tr>
<td>10SEP12</td>
<td>API Version 5.1</td>
</tr>
<tr>
<td>01SEP11</td>
<td>API Version 1.5</td>
</tr>
<tr>
<td>30AUG10</td>
<td>API Version 1.0</td>
</tr>
<tr>
<td>14APR10</td>
<td>API Version 0.9</td>
</tr>
</tbody>
</table>

**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 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.

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 publications, go to [http://www.vmware.com/support/pubs](http://www.vmware.com/support/pubs).
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 13
- “Client Workflow Overview,” on page 18
- “Using the vCloud API with vCloud Director,” on page 23
- “About the vCloud API Examples,” on page 25
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 vApp templates and media images. You can configure a catalog in several different ways:

- as a repository for local content that can remain private to the catalog owner or can be shared with other users, groups, or organizations in your cloud
- as a source of published content, to which other clouds can subscribe.
as a local repository for content published by another cloud or any Web site that hosts a VMware Content Subscription Protocol (VCSP) endpoint.

An organization administrator or catalog owner controls catalog sharing. Organization administrators in organizations that have permission to publish catalogs control publication and subscription options for catalogs in their organization. A system administrator can enable background synchronization of catalogs with external sources and set background synchronization schedules to regulate consumption of network bandwidth by this activity.

**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 ISO-format media images are stored in a catalog and represented as catalog item objects. Virtual systems are 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 are tracked by task objects. Running and recently completed tasks initiated by members of an organization are kept on the organization’s tasks list.
Objects, References, and Representations

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

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

**id**

The object identifier, expressed in URN format. The value of the id attribute uniquely identifies the object, persists for the life of the object, and is never reused. The id attribute value is intended to provide a context-free identifier that can be used with the vCloud API entityResolver (see “Retrieve an Object as an Entity,” on page 392).

**type**

The object type, specified as a MIME content type.

**href**

An object reference, expressed in URL format. This reference includes the object identifier portion of the id attribute value, and supplies additional information, including the current location of the object when accessed in a specific view. 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.

Views

The vCloud API defines several contexts, or views, in which you can access objects in a cloud. These views are expressed in the URL returned as the href of an object, and have the following forms, where API-URL is a URL of the form https://vcloud.example.com/api and object-type is a string indicating the type of the object.

**user view**

A URL of the form API-URL/object-type/id indicates that any user can access the object.

**admin view**

A URL of the form API-URL/admin/object-type/id indicates that organization administrators and system administrators can access the object. Organization administrators do not have rights to modify some objects in the admin view.

**extension view**

A URL of the form API-URL/admin/extension/object-type/id indicates that system administrators can access the object.

A given object retrieved in one view may have a different representation and media type from the same object retrieved in a different view. Not all objects are presented in every view.

Example: Object id, type, and href Attributes

These abbreviated request and response examples show the id, type, and href attributes in the user and admin views of an organization.

Request:

GET https://vcloud.example.com/api/org/72fe715c-5f6a-407f-bbb2-bf465915b5f4
Response:

```xml
<Org
  ...
  id="urn:vcloud:org:72fe715c-5f6a-407f-bbb2-bf465915b5f4"
  type="application/vnd.vmware.organization+xml"
  href="https://vcloud.example.com/api/org/72fe715c-5f6a-407f-bbb2-bf465915b5f4"
  ...
  ...
</Org>
```

The id value is the same in both cases, but the type and href attributes have values specific to the view.

Request:

```
GET https://vcloud.example.com/api/admin/org/72fe715c-5f6a-407f-bbb2-bf465915b5f4
```

Response:

```xml
<AdminOrg
  ...
  id="urn:vcloud:org:72fe715c-5f6a-407f-bbb2-bf465915b5f4"
  type="application/vnd.vmware.admin.organization+xml"
  href="https://vcloud.example.com/api/admin/org/72fe715c-5f6a-407f-bbb2-bf465915b5f4"
  ...
  ...
</AdminOrg>
```

The value of the id attribute is a permanent, unique object identifier. The value of the href attribute is an object locator that refers to a specific view of the object in its current location. Unlike the value of the id attribute, object location and view context can change during the life of an object. The example in “Example: Using the entityResolver URL,” on page 393 shows how to retrieve this object as entity.

When a client application must keep a persistent reference to an object, the best practice is to keep a reference to the id and the href (URL) that was most recently used to access the object. When the application needs to access the object in the future, it should first try using the saved href. If that fails, use the id with the entity resolver to obtain a valid reference to the object, then replace the saved href with that valid reference.

**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:

```xml
<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. This reference includes the object identifier portion of the `id` attribute value, and supplies additional information, including the current location of the object when accessed in a specific view. 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

<table>
<thead>
<tr>
<th>rel Attribute Value</th>
<th>Action or Relationship Description</th>
<th>Implied HTTP Verb</th>
</tr>
</thead>
<tbody>
<tr>
<td>abort</td>
<td>Abort this blocking task.</td>
<td>POST</td>
</tr>
<tr>
<td>add</td>
<td>Add an item to this container.</td>
<td>POST</td>
</tr>
<tr>
<td>alternate</td>
<td>References an alternate representation of this object.</td>
<td>GET</td>
</tr>
<tr>
<td>answer</td>
<td>Provide user input requested by a virtual machine.</td>
<td>POST</td>
</tr>
<tr>
<td>authorization:check</td>
<td>Check whether an extension service operation is authorized for an entity.</td>
<td>POST</td>
</tr>
<tr>
<td>blockingTask</td>
<td>A list of pending blocking task requests in this cloud.</td>
<td>GET</td>
</tr>
<tr>
<td>bundle:upload</td>
<td>Upload an extension service localization bundle.</td>
<td>PUT</td>
</tr>
<tr>
<td>bundles:cleanup</td>
<td>Remove unused extension service localization bundles.</td>
<td>POST</td>
</tr>
<tr>
<td>catalogItem</td>
<td>References the CatalogItem object that refers to this object.</td>
<td>GET</td>
</tr>
<tr>
<td>certificate:reset</td>
<td>Removes the SSL certificate used by this service.</td>
<td>POST</td>
</tr>
<tr>
<td>certificate:update</td>
<td>Updates the SSL certificate used by this service.</td>
<td>POST</td>
</tr>
<tr>
<td>checkCompliance</td>
<td>Check that this virtual machine is using a storage profile of the intended type.</td>
<td>POST</td>
</tr>
<tr>
<td>consolidate</td>
<td>Consolidate this virtual machine.</td>
<td>POST</td>
</tr>
<tr>
<td>controlAccess</td>
<td>Apply access controls to this object.</td>
<td>POST</td>
</tr>
<tr>
<td>copy</td>
<td>Reserved</td>
<td>N/A</td>
</tr>
<tr>
<td>customizeAtNextPowerOn</td>
<td>Force guest customization to be applied the next time this virtual machine is powered on.</td>
<td>POST</td>
</tr>
<tr>
<td>deploy</td>
<td>Deploy this vApp.</td>
<td>POST</td>
</tr>
<tr>
<td>disable</td>
<td>Disable this object.</td>
<td>POST</td>
</tr>
<tr>
<td>discardState</td>
<td>Discard the suspended state of this virtual machine.</td>
<td>POST</td>
</tr>
<tr>
<td>disk:attach</td>
<td>Attach an independent disk to this virtual machine.</td>
<td>POST</td>
</tr>
</tbody>
</table>
### Table 1-1. Link Relationships and HTTP Request Types (Continued)

<table>
<thead>
<tr>
<th>rel Attribute Value</th>
<th>Action or Relationship Description</th>
<th>Implied HTTP Verb</th>
</tr>
</thead>
<tbody>
<tr>
<td>disk:detach</td>
<td>Detach an independent disk from this virtual machine.</td>
<td>POST</td>
</tr>
<tr>
<td>down</td>
<td>References an object contained by this object.</td>
<td>GET</td>
</tr>
<tr>
<td>down: ACL Rules</td>
<td>Retrieve the ACL rules for this resource class action.</td>
<td>GET</td>
</tr>
<tr>
<td>down: api definitions</td>
<td>Retrieve the API definitions for this extension service.</td>
<td>GET</td>
</tr>
<tr>
<td>down: apiDefinitions</td>
<td>Retrieve the API definitions for this extension service.</td>
<td>GET</td>
</tr>
<tr>
<td>down: apiFilters</td>
<td>Retrieve the API filters for this extension service.</td>
<td>GET</td>
</tr>
<tr>
<td>down: extensibility</td>
<td>Add an extension service to the system.</td>
<td>POST</td>
</tr>
<tr>
<td>down: file Descriptors</td>
<td>Retrieve file descriptors for extension services APIs</td>
<td>GET</td>
</tr>
<tr>
<td>down: files</td>
<td>Retrieve files for extension services APIs</td>
<td>GET</td>
</tr>
<tr>
<td>down: resourceClassActions</td>
<td>Retrieve the actions defined for this extension service resource class.</td>
<td>GET</td>
</tr>
<tr>
<td>down: resourceClasses</td>
<td>Retrieve the resource classes defined by this extension service.</td>
<td>GET</td>
</tr>
<tr>
<td>down: serviceLinks</td>
<td>Retrieve the service links defined by this extension service.</td>
<td>GET</td>
</tr>
<tr>
<td>down: serviceResources</td>
<td>Retrieve the list of extension service resources of this class.</td>
<td>GET</td>
</tr>
<tr>
<td>down: services</td>
<td>Retrieve the list of registered extension services.</td>
<td>GET</td>
</tr>
<tr>
<td>download: alternate</td>
<td>Reserved</td>
<td>N/A</td>
</tr>
<tr>
<td>download: default</td>
<td>References the default location from which this file can be downloaded.</td>
<td>GET</td>
</tr>
<tr>
<td>download: identity</td>
<td>References the extended OVF descriptor of this vApp template. The extended OVF descriptor contains additional information such as MAC address, BIOS UUID, and NetworkConfigSection</td>
<td>GET</td>
</tr>
<tr>
<td>edgeGateway: configureServices</td>
<td>Update the network services offered by this Edge Gateway.</td>
<td>PUT</td>
</tr>
<tr>
<td>edgeGateway: reapplyServices</td>
<td>Reapply (after an update) the network services offered by this Edge Gateway.</td>
<td>POST</td>
</tr>
<tr>
<td>edgeGateway: redeploy</td>
<td>Redeploy the vShield Edge supporting this Edge Gateway.</td>
<td>POST</td>
</tr>
<tr>
<td>edgeGateway: syncSyslogSettings</td>
<td>Synchronize syslog server addresses used by this Edge Gateway with system defaults.</td>
<td>POST</td>
</tr>
<tr>
<td>edgeGateway: upgrade</td>
<td>Upgrade the backing configuration of this Edge Gateway from compact to full.</td>
<td>POST</td>
</tr>
<tr>
<td>edgeGateways</td>
<td>List the Edge Gateway objects in this organization VDC.</td>
<td>GET</td>
</tr>
<tr>
<td>edit</td>
<td>Modify this object, typically by replacing its current representation with the one in the request body.</td>
<td>PUT</td>
</tr>
<tr>
<td>enable</td>
<td>Enable this object.</td>
<td>POST</td>
</tr>
<tr>
<td>enterMaintenanceMode</td>
<td>Put this virtual machine into maintenance mode.</td>
<td>POST</td>
</tr>
<tr>
<td>entity</td>
<td>Retrieve a representation of the object on which an operation triggered this notification.</td>
<td>GET</td>
</tr>
<tr>
<td>entityResolver</td>
<td>Retrieve an object id as a context-free Entity element.</td>
<td>GET</td>
</tr>
<tr>
<td>event: create</td>
<td>Create an event in an this organization's event stream.</td>
<td>POST</td>
</tr>
</tbody>
</table>
Table 1.1. Link Relationships and HTTP Request Types (Continued)

<table>
<thead>
<tr>
<th>rel Attribute Value</th>
<th>Action or Relationship Description</th>
<th>Implied HTTP Verb</th>
</tr>
</thead>
<tbody>
<tr>
<td>exitMaintenanceMode</td>
<td>Take this virtual machine out of maintenance mode.</td>
<td>POST</td>
</tr>
<tr>
<td>fail</td>
<td>Fail this blocking task.</td>
<td>POST</td>
</tr>
<tr>
<td>firstPage</td>
<td>Reference to the first page of a paginated response.</td>
<td>GET</td>
</tr>
<tr>
<td>installVmwareTools</td>
<td>Install VMware Tools on this virtual machine.</td>
<td>POST</td>
</tr>
<tr>
<td>keystore:reset</td>
<td>Removes the keystore used by this service.</td>
<td>POST</td>
</tr>
<tr>
<td>keystore:update</td>
<td>Updates the keystore used by this service.</td>
<td>POST</td>
</tr>
<tr>
<td>keytab:reset</td>
<td>Removes the keytab used by this service.</td>
<td>POST</td>
</tr>
<tr>
<td>keytab:update</td>
<td>Updates the keytab used by this service.</td>
<td>POST</td>
</tr>
<tr>
<td>lastPage</td>
<td>Reference to the last page of a paginated response.</td>
<td>GET</td>
</tr>
<tr>
<td>media:ejectMedia</td>
<td>Eject virtual media from a virtual device.</td>
<td>POST</td>
</tr>
<tr>
<td>media:insertMedia</td>
<td>Insert virtual media into a virtual device.</td>
<td>POST</td>
</tr>
<tr>
<td>metrics</td>
<td>Retrieve a subset of current or historic metrics from a virtual machine.</td>
<td>POST</td>
</tr>
<tr>
<td>merge</td>
<td>Merge one or more Provider VDCs with this Provider VDC.</td>
<td>POST</td>
</tr>
<tr>
<td>migrateVms</td>
<td>Migrate virtual machines from this resource pool to a different one.</td>
<td>POST</td>
</tr>
<tr>
<td>move</td>
<td>Reserved</td>
<td>N/A</td>
</tr>
<tr>
<td>nextPage</td>
<td>Reference to the next page of a paginated response.</td>
<td>GET</td>
</tr>
<tr>
<td>orgVdcNetworks</td>
<td>List the organization VDC networks supported by this Edge Gateway.</td>
<td>GET</td>
</tr>
<tr>
<td>ova</td>
<td>Reserved</td>
<td>N/A</td>
</tr>
<tr>
<td>ovf</td>
<td>References the OVF descriptor of this vApp template.</td>
<td>GET</td>
</tr>
<tr>
<td>power:powerOff</td>
<td>Power off this vApp or virtual machine.</td>
<td>POST</td>
</tr>
<tr>
<td>power:powerOn</td>
<td>Power on this vApp or virtual machine.</td>
<td>POST</td>
</tr>
<tr>
<td>power:reboot</td>
<td>Reboot this vApp or virtual machine.</td>
<td>POST</td>
</tr>
<tr>
<td>power:reset</td>
<td>Reset this vApp or virtual machine.</td>
<td>POST</td>
</tr>
<tr>
<td>powershutdown</td>
<td>Shut down this vApp or virtual machine.</td>
<td>POST</td>
</tr>
<tr>
<td>powersuspend</td>
<td>Suspend this vApp or virtual machine.</td>
<td>POST</td>
</tr>
<tr>
<td>previousPage</td>
<td>Reference to the previous page of a paginated response.</td>
<td>GET</td>
</tr>
<tr>
<td>publish</td>
<td>Share this catalog.</td>
<td>POST</td>
</tr>
<tr>
<td>publishToExternalOrganizations</td>
<td>Publish this catalog externally</td>
<td>POST</td>
</tr>
<tr>
<td>recompose</td>
<td>Recompose this vApp to add, remove, or reconfigure virtual machines.</td>
<td>POST</td>
</tr>
<tr>
<td>reconfigureVm</td>
<td>Update multiple sections of a virtual machine.</td>
<td>POST</td>
</tr>
<tr>
<td>reconnect</td>
<td>Reconnect this vCenter Server to the system.</td>
<td>POST</td>
</tr>
<tr>
<td>refreshStorageProfiles</td>
<td>Refresh the list of storage profiles that exist on the vCenter service backing this Provider VDC.</td>
<td>POST</td>
</tr>
<tr>
<td>refreshVirtualCenter</td>
<td>Refresh the representation of this vCenter server.</td>
<td>POST</td>
</tr>
<tr>
<td>register</td>
<td>Register a VCenter Server with the system.</td>
<td>POST</td>
</tr>
<tr>
<td>relocate</td>
<td>Relocate this virtual machine.</td>
<td>POST</td>
</tr>
<tr>
<td>rel Attribute Value</td>
<td>Action or Relationship Description</td>
<td>Implied HTTP Verb</td>
</tr>
<tr>
<td>---------------------</td>
<td>------------------------------------</td>
<td>-------------------</td>
</tr>
<tr>
<td>remove</td>
<td>Remove this object.</td>
<td>DELETE</td>
</tr>
<tr>
<td>remove:force</td>
<td>Force removal of this object.</td>
<td>DELETE</td>
</tr>
<tr>
<td>repair</td>
<td>Repair this host or network.</td>
<td>POST</td>
</tr>
<tr>
<td>resourcePoolVmList</td>
<td>List the virtual machines using this resource pool.</td>
<td>GET</td>
</tr>
<tr>
<td>resume</td>
<td>Resume this blocking task.</td>
<td>POST</td>
</tr>
<tr>
<td>rights</td>
<td>List the service-specific rights created by this extension service.</td>
<td>GET</td>
</tr>
<tr>
<td>rights:cleanup</td>
<td>Remove service-specific rights no longer used by any extension service.</td>
<td>POST</td>
</tr>
<tr>
<td>screen:acquireTicket</td>
<td>Retrieve a screen ticket for this virtual machine.</td>
<td>GET</td>
</tr>
<tr>
<td>screen:thumbnail</td>
<td>Retrieve a thumbnail view of the screen of this virtual machine.</td>
<td>GET</td>
</tr>
<tr>
<td>shadowVms</td>
<td>List shadow virtual machines associated with the virtual machines in this vApp template.</td>
<td>GET</td>
</tr>
<tr>
<td>snapshot:create</td>
<td>Create a snapshot of the virtual machines in this vApp.</td>
<td>POST</td>
</tr>
<tr>
<td>snapshot:removeAll</td>
<td>Remove all snapshots created for the virtual machines in this vApp.</td>
<td>POST</td>
</tr>
<tr>
<td>snapshot:revertToCurrent</td>
<td>Revert all virtual machines in this vApp to their current snapshot.</td>
<td>POST</td>
</tr>
<tr>
<td>storageProfile</td>
<td>References the storage profile for this object.</td>
<td>GET</td>
</tr>
<tr>
<td>subscribeToExternalCatalog</td>
<td>Add an external subscription to this catalog.</td>
<td>POST</td>
</tr>
<tr>
<td>sync</td>
<td>Synchronize this catalog or catalog item with its external source.</td>
<td>POST</td>
</tr>
<tr>
<td>syncSyslogSettings</td>
<td>Synchronize syslog server addresses used by this vApp network with system defaults.</td>
<td>POST</td>
</tr>
<tr>
<td>takeOwnership</td>
<td>Take ownership of this user's vApps, media, and catalogs.</td>
<td>POST</td>
</tr>
<tr>
<td>task</td>
<td>Retrieve the blocking task that triggered this notification.</td>
<td>GET</td>
</tr>
<tr>
<td>task:cancel</td>
<td>Cancel this task.</td>
<td>POST</td>
</tr>
<tr>
<td>task:create</td>
<td>Create a task object.</td>
<td>POST</td>
</tr>
<tr>
<td>task:owner</td>
<td>Reference to the owner of a task.</td>
<td>GET</td>
</tr>
<tr>
<td>truststore:reset</td>
<td>Remove the truststore used by this service.</td>
<td>POST</td>
</tr>
<tr>
<td>truststore:update</td>
<td>Update the truststore used by this service.</td>
<td>PUT</td>
</tr>
<tr>
<td>undeploy</td>
<td>Undeploy this vApp.</td>
<td>POST</td>
</tr>
<tr>
<td>unlock</td>
<td>Unlock this user account.</td>
<td>POST</td>
</tr>
<tr>
<td>unregister</td>
<td>Unregister this vCenter Server.</td>
<td>POST</td>
</tr>
<tr>
<td>up</td>
<td>References an object that contains this object.</td>
<td>GET</td>
</tr>
<tr>
<td>update:resourcePools</td>
<td>Update the resource pools of this Provider VDC</td>
<td>POST</td>
</tr>
<tr>
<td>updateProgress</td>
<td>Request an update of this task's progress.</td>
<td>POST</td>
</tr>
<tr>
<td>upgrade</td>
<td>Upgrade this host.</td>
<td>POST</td>
</tr>
<tr>
<td>upload:alternate</td>
<td>Reserved</td>
<td>N/A</td>
</tr>
</tbody>
</table>
### 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 use the Hypertext Transfer Protocol (HTTP) to exchange serialized representations of objects between a client and a server. In the vCloud API, these representations are XML documents.

In a RESTful workflow, representations of objects are passed back and forth between a client and a server 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 application, 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 that information in this Guide, and in the *vCloud API Schema Reference*. See “About the Schema Reference,” on page 25.

### 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.
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 document that can contain any of the following items:

- XML elements and attributes that represent object properties
- Link elements that implement operations on the object or its contents
- If the object is being created or modified, an embedded Task object that tracks the progress of the creation or modification

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 45.

Requests are typically categorized by the type of requested operation: create, retrieve, update, and delete. This sequence of verbs is often abbreviated with the acronym CRUD. Each type of request is characterized by the use of specific HTTP verb to access a URL found in a Link element that has an operation-specific value for its rel (relation) attribute.

**Table 1-2. CRUD Operations Summary**

<table>
<thead>
<tr>
<th>Operation Type</th>
<th>HTTP Verb</th>
<th>Link Relation</th>
<th>Operation Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>Create</td>
<td>POST</td>
<td>add</td>
<td>Creates a new object.</td>
</tr>
<tr>
<td>Retrieve</td>
<td>GET</td>
<td>down</td>
<td>Retrieves the representation of an existing object in its current state.</td>
</tr>
<tr>
<td>Update</td>
<td>PUT</td>
<td>edit</td>
<td>Modifies an existing object.</td>
</tr>
<tr>
<td>Delete</td>
<td>DELETE</td>
<td>remove</td>
<td>Deletes an existing object. If the object is a container, you must remove all of its contents before you can delete it.</td>
</tr>
</tbody>
</table>

For example, this Link element indicates that you can use the URL https://vcloud.example.com/api/admin/org/26 to update the Org object that contains it.

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

The implied HTTP verb is PUT.

**Authentication**

HTTP communications between a vCloud API client and server are secured with SSL. API clients must also complete a login request to receive an authorization token that must be included in all subsequent requests.
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 vCloud API version that the client supports. The following header indicates that the request is from a vCloud API version 5.6 client:

```
Accept: application/*+xml;version=5.6
```

In general, client requests can access objects defined by any version of the vCloud API that is less than or equal to the version specified in the `Accept` header.

**Accept-Encoding**

By default, the system 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.

**Accept-Language**

Message strings in `ErrorType` responses are localized. To specify the language desired in responses, use the `Accept-Language` request header. To request a response with message strings localized to French, use the following header:

```
Accept-Language: fr
```

**Authorization**

All requests from authenticated clients must include an `Authorization` header. See “Create a Login Session Using the Integrated Identity Provider,” on page 47 and “Create a Login Session Using a SAML Identity Provider,” on page 49.

**Content-Type**

Requests that include a body must include an appropriate HTTP `Content-Type` header. Content types for all elements are listed 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"
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.

**X-VMWARE-VCLOUD-CLIENT-REQUEST-ID**

The value of this header is used to build a request ID returned in the value of the X-VMWARE-VCLOUD-REQUEST-ID header (see “Response Headers,” on page 22). The value of this header cannot contain more than 128 characters drawn from the set of letters, numbers, and the hyphen (-). Values with invalid characters are ignored. Values with more than 128 characters are truncated.

**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 389.
- 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.

**Request Limits**

To guard against denial-of-service attacks, vCloud Director imposes the following limits on vCloud API requests:

- Requests cannot exceed 512 KB.
- Requests cannot contain more than 4096 XML elements.
- Requests cannot have a depth greater than 100.

**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

<table>
<thead>
<tr>
<th>Status Code</th>
<th>Status Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>200 OK</td>
<td>The request is valid and was completed. The response includes a document body.</td>
</tr>
<tr>
<td>201 Created</td>
<td>The request is valid. The requested object was created and can be found at the URL specified in the Location header.</td>
</tr>
<tr>
<td>202 Accepted</td>
<td>The request is valid and a task was created to handle it. This response is usually accompanied by a Task element.</td>
</tr>
<tr>
<td>204 No Content</td>
<td>The request is valid and was completed. The response does not include a body.</td>
</tr>
<tr>
<td>400 Bad Request</td>
<td>The request body is malformed, incomplete, or otherwise invalid.</td>
</tr>
<tr>
<td>401 Unauthorized</td>
<td>The user’s session has expired.</td>
</tr>
<tr>
<td>403 Forbidden</td>
<td>Any of:</td>
</tr>
<tr>
<td></td>
<td>- One or more objects specified in the request could not be found in the specified container.</td>
</tr>
<tr>
<td></td>
<td>- The user is not authenticated or does not have adequate privileges to access one or more objects specified in the request.</td>
</tr>
<tr>
<td></td>
<td>- The user’s session has expired.</td>
</tr>
<tr>
<td>404 Not Found</td>
<td>Usually indicates a malformed request URL or request body.</td>
</tr>
<tr>
<td>405 Method Not Allowed</td>
<td>The HTTP method specified in the request is not supported for this object.</td>
</tr>
<tr>
<td>406 Not Acceptable</td>
<td>The resource identified by the request is not capable of generating a response of the type specified in the request’s Accept header.</td>
</tr>
<tr>
<td>409 Conflict</td>
<td>The object state is not compatible with the requested operation.</td>
</tr>
<tr>
<td>415 Unsupported Media Type</td>
<td>The resource identified by the request does not support a request of the specified Content-Type and HTTP method.</td>
</tr>
<tr>
<td>500 Internal Server Error</td>
<td>The request was received but could not be completed because of an internal error at the server.</td>
</tr>
<tr>
<td>503 Service Unavailable</td>
<td>The server is currently unable to handle the request due to a temporary condition such as resource exhaustion or server maintenance.</td>
</tr>
<tr>
<td>504 Gateway Timeout</td>
<td>The server, while acting as a gateway or proxy, did not receive a timely response from the upstream server specified by the request URL.</td>
</tr>
</tbody>
</table>

**Response Headers**

The following HTTP headers can appear in responses to vCloud API requests:

- **x-vcloud-authorization** An authentication token in the response to a login request. Use this token to authenticate subsequent requests. See “Logging In,” on page 28.
- **X-VMWARE-VCLOUD-REQUEST-ID** If a request supplied an X-VMWARE-VCLOUD-CLIENT-REQUEST-ID header, the response contains an X-VMWARE-VCLOUD-REQUEST-ID header whose value combines the value in the X-VMWARE-VCLOUD-CLIENT-REQUEST-ID with a unique ID. This value is added to every vCloud Director, vCenter, and ESXi log message related to processing the
request, and provides a way to correlate the processing of a request across all participating systems. If a request did not supply a X-VMWARE-VCLOUD-CLIENT-REQUEST-ID header, the response contains an X-VMWARE-VCLOUD-REQUEST-ID header with a generated value that cannot be used for log correlation.

**X-VMWARE-VCLOUD-REQUEST-EXECUTION-TIME**

The execution time, in milliseconds, of the request that generated this response.

### Using the vCloud API with vCloud Director

VMware vCloud Director supports several versions 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 25. For information about HTTP client programs to use with vCloud Director, see “REST Client Programs,” on page 25.

**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) Retrieve the list of supported API versions from the server.

   After the vCloud Director REST API base URL has been configured, any HTTP client can request a document that lists all the API versions that the server supports. See “Example: Retrieve the Login URL and List of Supported API Versions,” on page 23.

3. (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.

**Example: Retrieve the Login URL and List of Supported API Versions**

The api/versions request can be made by any client, whether or not the client is authenticated by vCloud Director. The response, a small subset of which is shown here, includes a VersionInfo element for each API version that this vCloud Director installation 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 28.
- 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"
xmllns: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>5.1</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>
  </VersionInfo>
  ...<VersionInfo>
    <Version>5.5</Version>
    <LoginUrl>https://vcloud.example.com/api/sessions</LoginUrl>
    <MediaTypeMapping>
      ...</MediaTypeMapping>
  </VersionInfo>
  ...<VersionInfo>
    <Version>5.6</Version>
    <LoginUrl>https://vcloud.example.com/api/sessions</LoginUrl>
    <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. This type of request can be made by any client, whether or not the client is authenticated by vCloud Director. For example, this request retrieves the schema file master.xsd:

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

All vCloud API requests are processed in the http://www.vmware.com/vcloud/v1.5 XML namespace. Schema files for all API versions can be retrieved from a .../api/v1.5/... URL.

**What to do next**

Decide on an HTTP client program to use. See “REST Client Programs,” on page 25.
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 the vCloud API Programming Guide 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 21. 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 5.6 of the vCloud API, is omitted from most examples.
  
  Accept: application/*+xml;version=5.6

- 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-c518-4f53-968c-950db9e3a1fd.
Hello vCloud: A Simplified RESTful Workflow

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 28
   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 30
   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 31
   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 32
   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 34
   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 35
   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 38
   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.
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.

After you undeploy a vApp and power it off, you can use an HTTP DELETE request to delete the vApp object.

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 “Example: Retrieve the Login URL and List of Supported API Versions,” on page 23. 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 29.

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 have the login credentials of a user with the vApp Author role.
- 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 169.
- 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 57.

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 29.
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 47. 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 29. 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 47.

**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 172. 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=5.6
```

Response:

```
200 OK
x-vcloud-authorization: cn9uYmdugN8E2j96+5Lqrc3YBvFsEgDHXzyf3jJ/6BM=
Content-Type: application/vnd.vmware.vcloud.session+xml;version=5.6
...
<Session
   xmlns="http://www.vmware.com/vcloud/v1.5"
   user="HelloUser"
   org="ExampleOrg"
   ...>
   <Link
      rel="down"
      type="application/vnd.vmware.vcloud.orgList+xml"
      href="https://vcloud.example.com/api/org"/>
   <Link
      rel="down"
      type="application/vnd.vmware.vcloud.query.queryList+xml"
      href="https://vcloud.example.com/api/query"/>
   <Link
      rel="entityResolver"
      type="application/vnd.vmware.vcloud.entity+xml"
      href="https://vcloud.example.com/api/entity/"/>
</Session>
```
The response code indicates whether the request succeeded, or how it failed.

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

  x-vcloud-authorization: token

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

- If the authentication header is missing, the server returns HTTP response code 403.

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

Find a Catalog and a VDC

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

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

Prerequisites

Verify that you are logged in to the vCloud API as a user with the vApp Author role.

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 30.

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 30.

Example: Retrieve the Contents of an Organization

This example retrieves the ExampleOrg organization listed in the orgList element shown in Step 1.
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 user with the vApp Author role.

**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:

```xml
<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 32.

**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 30.

Request:

```plaintext
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 33, 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 user with the vApp Author role.

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 33.

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 32.

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
```
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 34 shows this subset of VDC contents.

Prerequisites

Verify that you are logged in to the vCloud API as a user with the vApp Author role.

Procedure

1. Retrieve the representation of your organization.
   
   See the request portion of "Example: Retrieve the Contents of an Organization," on page 30.

2. Examine the `org` response to find the links to the organization's VDCs.
   
   Links to VDCs have the form:
   
   ```xml
   <Link rel="down" type="application/vnd.vmware.vcloud.vdc+xml"
   href="https://vcloud.example.com/api/vdc/id"
   name="VDC_name" />
   ```

3. 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 34.

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">
```
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 user with the vApp Author role.

**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. You can also use the query service to return a list of references to vApp templates that you can access.

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 36 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 the vApp 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 36.

**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 97. 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 32 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:

    ```xml
    <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 34.

  - 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 193.

The target of the request is the instantiateVAppTemplate URL of this VDC. See “Example: Deployment Information in a VDC,” on page 34. 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"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  name="Linux FTP server"
  deploy="true"
  powerOn="true"
  xsi:schemaLocation="http://www.vmware.com/vcloud/v1.5"
>`
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

```xml
<vApp
 xmlns:ovf="http://schemas.dmtf.org/ovf/envelope/1"
><Description>Example FTP Server</Description>
<NetworkConfigSection>
  <ovf:Info>Configuration parameters for logical networks</ovf:Info>
  <NetworkConfig>
    <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>
  </NetworkConfig>
</NetworkConfigSection>
</InstantiationParams>
</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

```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
```
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 user with the vApp Author role.

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 38.

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 391.
- 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 36. 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 91.

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: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"
<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="undelete"
      href="https://vcloud.example.com/api/vApp/vm-4/action/undelete" />
    <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"/>
Displaying the Virtual Machine Console

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

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

NOTE A Vm element might also contain a link of the form:

```xml
<Link
  rel="screen:acquireMksTicket"
  type="application/vnd.vmware.vcloud.mksTicket+xml"
```

This link acquires a special kind ticket to be used with the WebMKS Javascript API, which is described in the VMware Remote Console SDK documentation.

**Prerequisites**

- Verify that you are logged in to the vCloud API as a user with the vApp Author role.
- 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 Remote Console SDK* documentation.

**Undeploy, Power Off, and Delete the vApp**

After you undeploy a vApp and power it off, you can 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 the vApp and 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 the vApp.

**Prerequisites**

Verify that you are logged in to the vCloud API as a user with the vApp Author role.

**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 was returned when you created the vApp from the template. See “Get Information About a vApp,” on page 38.
2. Undeploy the vApp, and specify that it should also be powered off.

Make a POST request to the vApp 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 43.

3. Retrieve the XML representation of the vApp again.

After it is 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 43.

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**

You can use the undeploy request body, an UndeployVAppParams element, to specify an UndeployPowerAction element. This example specifies an UndeployPowerAction of powerOff.

**NOTE** powerOff is the default UndeployPowerAction. It appears 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 is 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)"
...
</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 44.

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 45
- “Create a Login Session Using the Integrated Identity Provider,” on page 47
- “Retrieve a List of Organizations Accessible to You,” on page 52
- “Retrieve an Administrative View of a Cloud,” on page 53
- “Retrieve a List of vSphere Platform Operations and Objects for a Cloud,” on page 55

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>
<thead>
<tr>
<th>Operation</th>
<th>Request</th>
<th>Request Body</th>
<th>Response</th>
</tr>
</thead>
<tbody>
<tr>
<td>Show login URL and list of supported API versions</td>
<td>GET API-URL/versions</td>
<td>None</td>
<td>SupportedVersions</td>
</tr>
<tr>
<td>Log in</td>
<td>POST API-URL/sessions</td>
<td>None</td>
<td>Session</td>
</tr>
<tr>
<td>Log out</td>
<td>DELETE API-URL/session</td>
<td>None</td>
<td>200 OK</td>
</tr>
<tr>
<td>Log in [DEPRECATED]</td>
<td>POST API-URL/login</td>
<td>None</td>
<td>OrgList</td>
</tr>
<tr>
<td>Log out [DEPRECATED]</td>
<td>POST API-URL/login</td>
<td>None</td>
<td>200 OK</td>
</tr>
<tr>
<td>Retrieve a list of entry points for browsing operations</td>
<td>GET API-URL/session</td>
<td>None</td>
<td>200 OK</td>
</tr>
<tr>
<td>Retrieve a list of organizations to which you have access</td>
<td>GET API-URL/org/</td>
<td>None</td>
<td>OrgList</td>
</tr>
<tr>
<td>Retrieve the contents of an organization</td>
<td>GET API-URL/org/id</td>
<td>None</td>
<td>Org</td>
</tr>
<tr>
<td>Retrieve properties of a network</td>
<td>GET API-URL/network/id</td>
<td>None</td>
<td>OrgNetwork</td>
</tr>
<tr>
<td>Retrieve the contents of a catalog</td>
<td>GET API-URL/catalog/id</td>
<td>None</td>
<td>Catalog</td>
</tr>
<tr>
<td>Retrieve properties of a catalog item</td>
<td>GET API-URL/catalogItem/id</td>
<td>None</td>
<td>CatalogItem</td>
</tr>
<tr>
<td>Retrieve the contents of a VDC</td>
<td>GET API-URL/vdc/id</td>
<td>None</td>
<td>Vdc</td>
</tr>
<tr>
<td>Retrieve properties of a media image</td>
<td>GET API-URL/media/id</td>
<td>None</td>
<td>Media</td>
</tr>
<tr>
<td>Retrieve a vApp template</td>
<td>GET API-URL/vAppTemplate/vappTemplate-id</td>
<td>None</td>
<td>VAppTemplate</td>
</tr>
<tr>
<td>Retrieve properties of a vApp</td>
<td>GET API-URL/vApp/vapp-id</td>
<td>None</td>
<td>VApp</td>
</tr>
<tr>
<td>Retrieve properties of a virtual machine</td>
<td>GET API-URL/vApp/vm-id</td>
<td>None</td>
<td>Vm</td>
</tr>
<tr>
<td>Retrieve metadata for an object such as an organization, network, vApp, virtual machine, media image, or independent disk.</td>
<td>GET API-URL/object/id/metadata</td>
<td>None</td>
<td>Metadata</td>
</tr>
<tr>
<td>Retrieve a list of IP addresses allocated to a network.</td>
<td>GET API-URL/network/id/allocatedAddresses</td>
<td>None</td>
<td>AllocatedIpAddresses</td>
</tr>
</tbody>
</table>
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 “Example: Retrieve the Login URL and List of Supported API Versions,” on page 23.
- 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 47.

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 265.

   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=5.6

Response:

200 OK
x-vcloud-authorization: cn9uYmdugN8E2j96+5Lqrc3YBvFsEyDxHyfJrJ/6bM=
Content-Type: application/vnd.vmware.vcloud.session+xml;version=5.6

...<Session
xmlns="http://www.vmware.com/vcloud/v1.5"
userUrn="urn:vcloud:user:2a60c0c3-...."
user="sysadmin"
org="System"
...
</Session>
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.

The vCloud API login mechanism supports Security Assertion Markup Language (SAML) authentication using two types security assertions:

- Bearer assertions, which can make no guarantees about message integrity and claimed client identity.
- Holder-of-key assertions, which guarantee subject identity by including a signature generated with the subject’s private key.

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 “Example: Retrieve the Login URL and List of Supported API Versions,” on page 23
- 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. SAML Authorization Header Attributes and Values

<table>
<thead>
<tr>
<th>Attribute Name</th>
<th>Attribute Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>token</td>
<td>The compressed, encoded identity assertion from your SAML identity provider.</td>
</tr>
<tr>
<td>signature</td>
<td>Base64 encoded signature of the token XML (the uncompressed identity assertion from your SAML identity provider) generated using client's private key. Required when using holder-of-key subject confirmation.</td>
</tr>
<tr>
<td>signature_alg</td>
<td>The algorithm used to generate the signature, expressed as one of the values listed in <a href="http://docs.oracle.com/javase/7/docs/technotes/guides/security/StandardNames.html#Signature">http://docs.oracle.com/javase/7/docs/technotes/guides/security/StandardNames.html#Signature</a> Required if signature is present.</td>
</tr>
<tr>
<td>org</td>
<td>The name of your vCloud Director organization. Defaults to org=&quot;system&quot; if not specified.</td>
</tr>
</tbody>
</table>

See “Example: Create a Login Session Using a SAML Identity Provider,” on page 50.

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 265.

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`. This example shows two varieties of the request.

Request (bearer token):

```
POST https://vcloud.example.com/api/sessions
Authorization: SIGN token="compressed-encoded-credentials",
org="Finance"
Accept: application/*+xml;version=5.6
```

When using a SAML assertion that provides holder-of-key (HOK) subject confirmation, the request header must include `signature` and `signature_alg` attributes, as shown in this example, which assumes a signature created with a SHA encoding and RSA encryption algorithms:
Request (holder-of-key token):

```plaintext
POST https://vcloud.example.com/api/sessions
Authorization: SIGN token="compressed-encoded-credentials",
org="Finance",
signature="encoded-signature"
signature_alg="SHA1withRSA"
Accept: application/4+xml;version=5.6
```

The response is the same in both cases.

Response:

```plaintext
200 OK
x-vcloud-authorization: cn9uYmdugN8E2j96+5Lqrc3YBvFsEgDHXzyfJrJl/6dM=
Content-Type: application/vnd.vmware.vcloud.session+xml;version=5.6
...
<Session
  xmlns="http://www.vmware.com/vcloud/v1.5"
  userUrn="urn:vcloud:user:fe50b0b5-...
  user="bob"
  org="Finance"
  ...
><Link
  rel="down"
  type="application/vnd.vmware.vcloud.org+xml"
  name="System"
  href="https://vcloud.example.com/api/org/5" />
<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>
```

Because this user has limited rights, the response includes only a few link types:

- **org**
  - A link to the user's organization. See “Retrieve a List of Organizations Accessible to You,” on page 52.

- **queryList**
  - A link to the set of typed queries you can run. See Chapter 9, “Using the Query Service,” on page 315.

- **entity**
  - A link to the entity resolver. See “Retrieve an Object as an Entity,” on page 392.

An administrator would see a more extensive set of links in the returned Session object. See “Example: Create a Login Session Using the Integrated Identity Provider,” on page 47.
Retrieve a List of Organizations Accessible to You

A successful login request returns a Session element, which contains Link elements that reference the 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. Unless you are the system administrator, this list includes just the organization to which you logged in. For a 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 47.

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 52.

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"
   names="ExampleOrg"
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 47.

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 53

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
Example Corporation's vCloud

<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"/>
</OrganizationReferences>

<ProviderVdcReferences>
  <ProviderVdcReference type="application/vnd.vmware.admin.providervdc+xml"
    name="Main Provider"
    href="https://vcloud.example.com/api/admin/providervdc/2"/>
</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"/>
</RightReferences>

<RoleReferences>
  <RoleReference type="application/vnd.vmware.admin.role+xml"
    name="Organization Administrator"
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 servers and hosts, vShield Manager, portgroups, virtual switches, and so on. The VMWExtension element provides access to a cloud-wide namespace of vSphere platform objects that are registered for use by the system, and links that allow you to add vSphere servers and related resources to your cloud. Objects in the admin/extension XML namespace 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 47.

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 56.
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.vmmProviderVdcReferences+xml"
 href="https://vcloud.example.com/api/admin/extension/providerVdcReferences" />
 <vcloud:Link
 rel="down"
 type="application/vnd.vmware.admin.vmmExternalNetworkReferences+xml"
 href="https://vcloud.example.com/api/admin/extension/externalNetworkReferences" />
 <vcloud:Link
 rel="down"
 type="application/vnd.vmware.admin.vmmNetworkPoolReferences+xml"
 href="https://vcloud.example.com/api/admin/extension/networkPoolReferences" />
 <vcloud:Link
 rel="down"
 type="application/vnd.vmware.admin.vmwwimServerReferences+xml"
 href="https://vcloud.example.com/api/admin/extension/vimServerReferences" />
 <vcloud:Link
 rel="down"
 type="application/vnd.vmware.admin.vmwwimHostReferences+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 a vCloud Director organization.

The vCloud API allows you to upload and download OVF packages and ISO-format media images. Operations are characterized as uploads when they transfer content from a vCloud API client system to a target catalog in a vCloud Director organization, and as downloads when a vCloud API client requests the transfer of content from vCloud Director. 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. Uploaded vApp templates and media images are made available as catalog items in the target catalog.

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 `cloneVApp` operation creates a copy of a vApp in a specified VDC. You can specify whether to delete the source vApp after the operation completes. Deleting the source vApp after cloning it moves or renames it.

- **Capturing**
  The vCloud API `captureVApp` operation creates a vApp template from a vApp and places the template in a specified catalog.

- **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.

- **Subscribing**
  Organizations that have the appropriate permissions can create catalogs with external subscriptions. These contents of these catalogs are downloaded from a catalog hosted on another instance of vCloud Director, or any Web site that implements the VMware Content Subscription Protocol. See “Create a Catalog With an External Subscription,” on page 227.

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 58
- “Upload an OVF Package to Create a vApp Template,” on page 60
- “Download a vApp or vApp Template as OVF,” on page 70
- “Upload a Media Image,” on page 74
- “Download a Media Image,” on page 76
Summary of vCloud API Provisioning Requests

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

<table>
<thead>
<tr>
<th>Operation</th>
<th>Request</th>
<th>Request Body</th>
<th>Response</th>
</tr>
</thead>
<tbody>
<tr>
<td>Upload OVF to a catalog create a vApp template.</td>
<td>POST API-URL/catalog/id/action/upload</td>
<td>UploadVAppTemplateParams</td>
<td>CatalogItem</td>
</tr>
<tr>
<td>Enable a vApp for download.</td>
<td>POST API-URL/vApp/id/action/enableDownload</td>
<td>None</td>
<td>Task</td>
</tr>
<tr>
<td>Disable a vApp for download.</td>
<td>POST API-URL/vApp/id/action/disableDownload</td>
<td>None</td>
<td>Task</td>
</tr>
<tr>
<td>Enable a vApp template for download.</td>
<td>POST API-URL/vAppTemplate/id/action/enableDownload</td>
<td>None</td>
<td>Task</td>
</tr>
<tr>
<td>Disable a vApp template for download.</td>
<td>POST API-URL/vAppTemplate/id/action/disableDownload</td>
<td>None</td>
<td>204 No Content</td>
</tr>
<tr>
<td>Download a vApp or vApp template as an OVF package.</td>
<td>GET download-URL</td>
<td>None</td>
<td>Depends on file type.</td>
</tr>
<tr>
<td>Upload a media image to a catalog.</td>
<td>POST API-URL/catalog/id/action/upload</td>
<td>Media</td>
<td>Media</td>
</tr>
<tr>
<td>Move a catalog item.</td>
<td>POST API-URL/catalogItem/id/action/move</td>
<td>CopyOrMoveCatalogItemParams</td>
<td>Task</td>
</tr>
<tr>
<td>Copy a catalog item.</td>
<td>POST API-URL/catalogItem/id/action/copy</td>
<td>CopyOrMoveCatalogItemParams</td>
<td>Task</td>
</tr>
</tbody>
</table>
Table 4-1. Summary of Provisioning Requests (Continued)

<table>
<thead>
<tr>
<th>Operation</th>
<th>Request</th>
<th>Request Body</th>
<th>Response</th>
</tr>
</thead>
<tbody>
<tr>
<td>Synchronize a catalog with its remote source.</td>
<td>POST API-URL/catalog/id/action/sync</td>
<td>None</td>
<td>Task</td>
</tr>
<tr>
<td>Synchronize a catalog item with its remote source.</td>
<td>POST API-URL/catalogItem/id/action/sync</td>
<td>None</td>
<td>Task</td>
</tr>
<tr>
<td>Change the name or description of a vApp template.</td>
<td>PUT API-URL/vAppTemplate/vAppTemplate-id</td>
<td>VAppTemplate</td>
<td>Task</td>
</tr>
<tr>
<td>Change the name or description of a media image.</td>
<td>PUT API-URL/media/id</td>
<td>Media</td>
<td>Task</td>
</tr>
<tr>
<td>Delete a vApp template, vApp, or media image.</td>
<td>DELETE object-URL</td>
<td>None</td>
<td>Task</td>
</tr>
<tr>
<td>Synchronize a catalog with its external source.</td>
<td>POST API-URL/catalog/id/action/sync</td>
<td>None</td>
<td>Task</td>
</tr>
<tr>
<td>Synchronize a catalog item with its external source.</td>
<td>POST API-URL/catalogItem/id/action/sync</td>
<td>None</td>
<td>Task</td>
</tr>
<tr>
<td>Remove an item from a catalog.</td>
<td>DELETE API-URL/catalog/id/catalogItem/id</td>
<td>None</td>
<td>204 No content</td>
</tr>
<tr>
<td>Control access to catalogs.</td>
<td>POST API-URL/catalog/id/action/controlAccess</td>
<td>ControlAccessParams</td>
<td>ControlAccessParams</td>
</tr>
<tr>
<td>Retrieve the owner of a media object.</td>
<td>GET API-URL/media/id/owner</td>
<td>None</td>
<td>Owner</td>
</tr>
<tr>
<td>Retrieve the owner of a vApp template.</td>
<td>GET API-URL/vAppTemplate/vAppTemplate-id/owner</td>
<td>None</td>
<td>Owner</td>
</tr>
<tr>
<td>Retrieve the owner of a vApp.</td>
<td>GET API-URL/vApp/id/owner</td>
<td>None</td>
<td>Owner</td>
</tr>
<tr>
<td>Update the owner of a vApp.</td>
<td>PUT API-URL/vApp/id/owner</td>
<td>Owner</td>
<td>204 No Content</td>
</tr>
<tr>
<td>Create an independent disk in a VDC.</td>
<td>POST API-URL/vdc/id/disk</td>
<td>DiskCreateParams</td>
<td>Disk</td>
</tr>
<tr>
<td>Retrieve properties of an independent disk.</td>
<td>GET API-URL/disk/id</td>
<td>None</td>
<td>Disk</td>
</tr>
<tr>
<td>Update an independent disk in a VDC.</td>
<td>POST API-URL/disk/id</td>
<td>Disk</td>
<td>Disk</td>
</tr>
<tr>
<td>Retrieve a list of all virtual machines attached to an independent disk.</td>
<td>GET API-URL/disk/id/attachedVms</td>
<td>None</td>
<td>Vms</td>
</tr>
<tr>
<td>Delete an independent disk.</td>
<td>DELETE API-URL/disk/id</td>
<td>None</td>
<td>Task</td>
</tr>
</tbody>
</table>
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 91.

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 makes a POST request to the catalog chosen to hold the template derived from the uploaded OVF. The request body specifies a name, description, and other parameters used when creating the template.

2. The server returns a `CatalogItem` that references a vApp template.

3. The client makes a GET request to the entity reference in the `CatalogItem` to retrieve the `VAppTemplate`, which includes an upload URL for the OVF descriptor. Because the template is not yet complete, the `VAppTemplate` element has `status="0"`.

4. The client makes a PUT request to the upload URL and supplies the OVF descriptor in the request body.

5. The server processes 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.

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

7. 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 391.

**Note** If you have an OVF package that you want to deploy immediately as a vApp, without creating a vApp template and corresponding catalog item, make an instantiateOvf request. See "Create a vApp From an OVF Package," on page 104.

### 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 61
   
   To initiate the OVF upload, a client makes a POST request to an action/upload link in the target catalog. The type of this link is application/vnd.vmware.vcloud.uploadVAppTemplateParams+xml. The request body is an UploadVAppTemplateParams element.

2. **Retrieving the Upload URL for the OVF Descriptor** on page 63
   
   After the vApp template and corresponding catalog item have been created, you must retrieve the template to get the upload URL for the OVF descriptor.

3. **Uploading the OVF Descriptor** on page 65
   
   You upload the OVF descriptor by making a PUT request to the upload URL created for it in the VAppTemplate. The request body is the descriptor's Envelope element. If the request is valid, the server responds with a 200 OK status.

4. **Retrieving Additional Upload URLs** on page 66
   
   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.

5. **Uploading Referenced Files** on page 67
   
   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 an action/upload link in the target catalog. The type of this link is application/vnd.vmware.vcloud.uploadVAppTemplateParams+xml. The request body is an UploadVAppTemplateParams element.

The first step in uploading an OVF package is to request vCloud Director to create a catalog item in the target catalog and a corresponding vAppTemplate object to represent the template that will be constructed from the upload.
Prerequisites

Verify that the following are true:

- You have 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 catalog that will receive the upload. Retrieve the XML representation of your organization to see a list of the catalogs that it contains.

Procedure

1. Find the action/upload link for vApp templates in the target catalog.

   Retrieve the XML representation of the catalog using a request like the one shown in the request portion of “Example: Deployment Information in a VDC,” on page 34. The response contains an action/upload link, which has the following form:

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

2. Create an UploadVAppTemplateParams element that specifies the parameters for the vApp template that this request creates.

   See the request portion of “Example: Initiating the Upload,” on page 62.

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 upload 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 62.

5. Examine the response.

   The response, a CatalogITem element, contains an Entity element that contains a reference to the vApp template that will be constructed from the uploaded OVF. See the response portion of “Example: Initiating the Upload,” on page 62.

   The server creates a VAppTemplate object and a corresponding CatalogItem in the target catalog, and returns an XML representation of the CatalogItem. See the response portion of “Example: Initiating the Upload,” on page 62.

Example: Initiating the Upload

This example assumes an OVF package that has no manifest.

Request:

```plaintext
POST https://vcloud.example.com/api/catalog/32/action/upload
Content-Type: application/vnd.vmware.vcloud.uploadVAppTemplateParams+xml
...
<?xml version="1.0" encoding="UTF-8"?>
<UploadVAppTemplateParams
    name="Ubuntu Template"
    ...
```
Retrieving the Upload URL for the OVF Descriptor

After the vApp template and corresponding catalog item have been created, you must retrieve the template to get the upload URL for the OVF descriptor.

**Procedure**

1. Examine the CatalogItem returned by the upload request to find the reference to the new vApp template.

   The reference is the value of the href attribute of the Entity element, as shown here.

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

2. Retrieve the VAppTemplate.

   See the request portion of “Example: OVF Descriptor Upload URL in a vAppTemplate,” on page 64.
3 Examine the template to find the upload URL for the OVF descriptor. These URLs are contained in Link elements where rel="upload:default".

**Example: OVF Descriptor Upload URL in a vAppTemplate**

This request uses the vApp template URL referenced in the Entity element shown in the response portion of “Example: Initiating the Upload,” on page 62.

Request:

```
GET https://vcloud.example.com/api/vAppTemplate/vappTemplate-111
```

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:111"
    href="https://vcloud.example.com/api/vAppTemplate/vappTemplate-111"
    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-111" />
    <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 `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`.

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 the upload URL created for it in the VAppTemplate. The request body is the descriptor’s `Envelope` element. If the request is valid, the server responds with a **200 OK** status.

**Prerequisites**

Verify that you have an upload URL for the OVF descriptor. See “Retrieving the Upload URL for the OVF Descriptor,” on page 63.

**Procedure**

1. **Upload the OVF descriptor.**

   Make a PUT request to the upload URL in the VAppTemplate. The upload URL for the OVF descriptor is in a `Link` element with the following form:

   ```xml
   <Link
       rel="upload:default"
       href="https://vcloud.example.com/transfer/.../descriptor.ovf"/>
   ```

   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:

   ```xml
   <Envelope
   ```

**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
```
Retrieving Additional 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.
   
   See the request portion of “Example: Upload URLs in a vAppTemplate,” on page 66.

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 “Retrieving the Upload URL for the OVF Descriptor,” on page 63.

Request:

GET https://vcloud.example.com/api/vAppTemplate/vappTemplate-111

Response:

200 OK

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

...
In this example, which omits most of the additional elements shown in “Example: Initiating the Upload,” on page 62, the ovfDescriptorUploaded attribute has a value of true and the status attribute has a value of 0. If the descriptor fails validation, status is set to -1, and the template contains a Task element whose Error element indicates the reason for the failure.

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

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

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

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

**NOTE** Upload URLs remain valid while a transfer session is in progress, and for a maximum of 60 minutes of transfer session idle time. A system administrator can change this default value. See “Retrieve or Update System Settings,” on page 265.

### 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 65.
- Retrieve the upload URLs for all files in the package. See “Retrieving Additional Upload URLs,” on page 66.

**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 68.

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...
```

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 checking 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 68.

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 68.

**Example: Monitoring the Progress of an Upload**

Request:

```
GET https://vcloud.example.com/api/vAppTemplate/vappTemplate-111
```

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:111"
  href="https://vcloud.example.com/api/vAppTemplate/vappTemplate-111"
  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 69.

**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"
...>
Download a vApp or vApp Template as OVF

You can download a vApp or vApp template object as an OVF package. After you enable the object for download, you can download the OVF descriptor, then download all of the files referenced by the descriptor. A vApp must powered off and undeployed before you can enable it for download.

When you enable a vApp or 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 or vApp template. The server excludes any deployment-specific information that the object contains, and populates the descriptor’s References element with references 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 descriptor references.
3. The server updates the corresponding VApp or VAppTemplate element with a link that contains a URL from which you can retrieve the OVF descriptor.
4. After you retrieve the descriptor, you can examine it to discover the download URLs for the files it references.

You can download any file referenced in the descriptor by making a GET request to its download URL.

1. Enable a vApp or vApp Template for Download on page 71
   Before you can download a vApp or vApp template, an administrator or privileged user must explicitly enable the object for download.
2. Download the OVF Descriptor of a vApp or vApp Template on page 72
   To download the OVF descriptor, make a GET request to the download:default or download:identity URL in the download-enabled VApp or VAppTemplate element.
3. Download a Referenced File on page 73
   After you download the OVF descriptor of a vApp or vApp template, you can examine the contents of the descriptor to discover download URLs for .vmdk and other files in the package.
Enable a vApp or vApp Template for Download

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

Prerequisites

- Verify that you are logged in as an administrator or other user who has privileges to enable a vApp or vApp template for download.
- Verify that any vApp you plan to enable for download is powered off and undeployed. See “Undeploy, Power Off, and Delete the vApp,” on page 42.

Procedure

1. Retrieve the XML representation of the VApp or VAppTemplate object.
2. Examine the representation to find its action/enableDownload link.
   
   Every VAppTemplate element includes a link of the following form, where \textit{id} is the id of the template:

   \[
   \text{<Link rel="enable" href="https://vcloud.example.com/api/vAppTemplate/vappTemplate-}\text{id/}\text{action/enableDownload"/>}
   \]
   
   Every vApp element includes a similar link:

   \[
   \text{<Link rel="enable" href="https://vcloud.example.com/api/vApp/}\text{id/}\text{action/enableDownload"/>}
   \]

3. Enable the object for download.
   
   Make a POST request to the action/enableDownload URL, which you retrieved in Step 2. The response is a Task that tracks the completion of the enablement operation.

4. When the task completes, retrieve the representation of the object, which now contains download URLs for the OVF descriptor.
   
   Download 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 265.

Example: vApp Template with Download URLs for OVF Descriptor

Request:

\[
\text{GET https://vcloud.example.com/api/vAppTemplate/vappTemplate-111}
\]

Response:

\[
200 \text{ OK}
\]

\[
\text{Content-Type: application/vnd.vmware.vcloud.vAppTemplate+xml}
\]

\[
\text{...}
\]

\[
\text{<VAppTemplate ovfDescriptorUploaded="true" status="8" name="Ubuntu Template"...>}
\]

\[
\text{...}
\]

\[
\text{<Link type="text/xml" rel="download:default" href="https://vcloud.example.com/transfer/.../descriptor.ovf"/>}
\]
Download the OVF Descriptor of a vApp or vApp Template

To download the OVF descriptor, make a GET request to the download:default or download:identity URL in the download-enabled VApp or VAppTemplate element.

You can download an OVF descriptor with or without identity information such as lease settings and network connection details. Identity information is typically not portable to another deployment environment.

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 or vApp template that is enabled for download. See “Enable a vApp or vApp Template for Download,” on page 71.

Procedure

1. Retrieve the XML representation of the VApp or VAppTemplate object.
2. Examine the representation to find the download URLs for the OVF descriptor.

   The download URLs are contained in Link elements, each with a different value for the rel attribute.

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Retrieve a descriptor that does not include identity information</td>
<td>Use the download:default URL.</td>
</tr>
<tr>
<td>Retrieve a descriptor that includes identity information</td>
<td>Use the download:identity URL.</td>
</tr>
</tbody>
</table>

3. Make a GET request to the URL that retrieves the descriptor you want.

   See “Example: Downloading the OVF Descriptor,” on page 72.

Example: Downloading the OVF Descriptor

This example downloads the OVF descriptor from the download:default URL shown in the href value of the Link shown in “Example: vApp Template with Download URLs for OVF Descriptor,” on page 71. 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 or vApp template, you can examine the contents of the descriptor to discover download URLs for .vmdk and other files in the package.

The OVF descriptor includes an href value for each file that the descriptor references. To retrieve one of these files, you must create a download URL for it by combining this href value with a URL derived from the download URL that you used to retrieve the descriptor. You must retrieve all of the referenced 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 or vApp template that has been 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 73.

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 72 with the file name shown in this File element:

```xml
<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

Uploading an ISO-format media image to a catalog creates a Media object and a corresponding CatalogItem object.

vCloud Director supports using the vCloud API to upload media images to a catalog.

**Note** Media images in formats other than ISO can be uploaded, but are given an imageType of other in the catalog.

The workflow for uploading media images is similar to the one shown in “Upload an OVF Package to Create a vApp Template,” on page 60.

**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 catalog that will receive the upload. Retrieve the XML representation of your organization to see a list of the catalogs that it contains.

**Procedure**

1. Find the add link for media in the target catalog
   
   This link has the following form:
   
   ```xml
   <Link
       rel="add"
       type="application/vnd.vmware.vcloud.media+xml"
       href="https://vcloud.example.com/api/catalog/32/action/upload" />
   ```

2. POST an action/upload request to the URL shown in Step 1
   
   The request body is a Media element. See the request portion of “Example: Upload a Media Image,” on page 74.
   
   The server uses this information to create a CatalogItem and corresponding Media object, then returns the CatalogItem in its response. See the response portion of “Example: Upload a Media Image,” on page 74.

3. Use the URL in the Entity element of the CatalogItem to retrieve the Media object.
   
   The Media element includes a File element that contains an upload:default URL.

4. 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 67.

**Example: Upload a Media Image**

There are two steps to uploading a media file. The first step is to make an action/upload request to the catalog. The request body is a Media element that specifies the size of the ISO file and the name that you want to apply to the created Media object. The imageType attribute is optional, and must be set to a value of iso if you supply it.
Request:

POST https://vcloud.example.com/api/catalog/32/action/upload
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>

Response:

201 Created
Content-Type: application/vnd.vmware.vcloud.catalogItem+xml
...
<CatalogItem
  xmlns="http://www.vmware.com/vcloud/v1.5"
  name="database.iso"
  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
    type="application/vnd.vmware.vcloud.media+xml"
    name="database.iso"
    href="https://vcloud.example.com/api/media/254" />
</CatalogItem>

Examine the response to the action/upload request, then make a GET request to the URL in the Entity element of the CatalogItem to retrieve the Media object.

GET https://vcloud.example.com/api/media/254

The Media object includes an upload URL for the media file itself.

<Media ... >
... 
<Files>
  <File
    name="database.iso"
    bytesTransferred="0"
<Link
  rel="upload:default"
  href="https://vcloud.example.com/transfer/.../database.iso" />
</File>
</Files>

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 67.

The upload 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 265.

Download a Media Image

The vCloud API supports downloading media images from a catalog.

Prerequisites

Verify that the following conditions are met:

- You are logged in as a user who has permission to download media images.
- You know the URL of the catalog item that references the media image.

Procedure

1. Retrieve the XML representation of the catalog and examine the catalog items that it contains.
2. Retrieve the catalog item that represents the media image.
3. Use the URL in the Entity element of the CatalogItem to retrieve the Media object.
   - The Media element includes a Link element of the following form, where id is the id of the media image:
     <Link
       rel="enable"
       href="https://vcloud.example.com/api/media/id/action/enableDownload"/>
4. Enable the media image for download.
   - Make a POST request to the action/enableDownload URL shown in Step 2. The response is a Task element.
5. When the task completes, retrieve the media item again.
   - The Media object now includes a download URL for the media file.
6. Make a GET request to the download:default URL.
   - The media file is downloaded to the current working directory.

Example: Download a Media Image

When you download a media file, you first enable the file for download.

Request:

POST https://vcloud.example.com/api/media/254/action/enableDownload
Response:

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

...<Task
...
operation="Enabling download of Media database.iso (254)" ... >
...
</Task>

The Task in the response tracks the creation of the downloadable image. When the task completes, retrieve the Media element again.

GET https://vcloud.example.com/api/media/254

The Media object now includes a download URL for the media file.

<Media ...
...
</Files
...

The download 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 265.

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.

As an administrator, catalog author, or vApp author, you can capture an undeployed vApp to create a vApp template in a catalog. Instantiating this template recreates the vApp from which the template was captured. Capturing a vApp in this way provides a way to save the results of composing, recomposing, or modifying a vApp after the vApp is undeployed. In addition, capturing a vApp preserves all vApp reconfiguration in template form. Although most elements of a vApp template are read-only, you can instantiate a template, modify the resulting vApp, and capture it to create a modified version of the template. See “Capture a vApp as a Template,” on page 114

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 296.
Managing Catalog Items

Catalog items are references to vApp templates and media files. If you have the appropriate rights, you can copy, move, rename, or delete catalog items in your organization's catalogs. You cannot modify catalog items in catalogs that have an external subscription.

After you add vApp templates or media files to a catalog, you might need to modify the `CatalogItem` objects that represent them. Your rights to manipulate catalog items depend on the source from which the catalog items were created.

- If you are a catalog author or an administrator, you can copy, move, delete, or rename catalog items that were uploaded, imported, or captured to a catalog that your organization owns, whether or not the catalog is published externally. Changes you make to an externally published catalog are replicated to all of the catalog's subscribers when those subscribers synchronize their copy of the catalog.
- You cannot make changes to catalog items in catalogs that have an external subscription.

Changes to a catalog item increment the version number of the item and its containing catalog. See “Version Numbers,” on page 222.

In addition to providing storage for locally created vApp templates and media files, catalogs provide a flexible publication mechanism that supports distribution of content to other organizations and clouds. If your organization allows it, you can publish a catalog to external consumers. You can also subscribe to catalogs that external sources publish, although catalog items in such catalogs cannot be managed by subscribers. See “Catalog Administration,” on page 221.

Copy or Move a Catalog Item

A `Catalog` object includes links that implement copy and move operations for the catalog items it contains.

To copy or move a catalog item from a source catalog to a target catalog, POST a `CopyOrMoveCatalogItemParams` element that contains a reference to the catalog item to move to the copy or move link of the target catalog.

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.
- Verify that the target catalog does not have an external subscription.

Procedure

1. Retrieve the XML representation of the source catalog.
   
   Use a request like this one:

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

2. Examine the `Catalog` element to find the `CatalogItem` elements it contains.
   
   Each `CatalogItem` in the `CatalogItems` container has `name`, `type`, and `href` attributes. If you need more information about a catalog item, you can retrieve it with a GET request to the URL in its `href` attribute.

3. Retrieve the XML representation of the target catalog.
4 Examine the Catalog element to find the copy and move links it contains.

These links have the following form:

```xml
<Link
    rel="copy"
    type="application/vnd.vmware.vcloud.copyOrMoveCatalogItemParams+xml"
    href="https://vcloud.example.com/api/catalog/44/action/copy" />
</Link>
```

```xml
<Link
    rel="move"
    type="application/vnd.vmware.vcloud.copyOrMoveCatalogItemParams+xml"
    href="https://vcloud.example.com/api/catalog/44/action/move" />
</Link>
```

5 Create a CopyOrMoveCatalogItemParams element that specifies the catalog item in the Source element.

See “Example: Copy a Catalog Item,” on page 79.

6 POST the CopyOrMoveCatalogItemParams to the appropriate link from the target catalog.

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Copy the Catalog Item</td>
<td>POST the CopyOrMoveCatalogItemParams to the rel=&quot;copy&quot; link.</td>
</tr>
<tr>
<td>Move the Catalog Item</td>
<td>POST the CopyOrMoveCatalogItemParams to the rel=&quot;move&quot; link.</td>
</tr>
</tbody>
</table>

**Example: Copy a Catalog Item**

This request copies the catalog item shown in “Example: Retrieve a Catalog Item,” on page 33 to another catalog. The response is a Task.

Request:

```plaintext
POST https://vcloud.example.com/api/catalog/44/action/copy
Content-Type: application/vnd.vmware.vcloud.copyOrMoveCatalogItemParams+xml

<?xml version="1.0" encoding="UTF-8"?>
<CopyOrMoveCatalogItemParams
    xmlns="http://www.vmware.com/vcloud/v1.5"
    name="Ubuntu 10.04 Template">
    <Description>Reference copy of Ubuntu FTP Server</Description>
    <Source
        href="https://vcloud.example.com/api/catalogItem/221" />
</CopyOrMoveCatalogItemParams>
```

Response:

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

<Task ... operation="Copying Virtual Application Template Ubuntu 10.04 Template" ...>
...

</Task>
```

**Change the Name or Description of a Catalog Item**

Every CatalogItem object includes a rel="edit" link that you can use to modify the name or description of the catalog item.

**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.
Verify that the target catalog does not have an external subscription.

**Procedure**

1. Retrieve the catalog item from the catalog.
2. Locate the `rel="edit"` link in the `CatalogItem` element.
3. Modify the retrieved `CatalogItem` element to change its name, description, or both.
   
   See "Example: Change the Name and Description of a Catalog Item,” on page 80.
4. Make a PUT request to the `href` value of the `rel="edit"` link in the `CatalogItem`, supplying the modified `CatalogItem` in the request body.

**Example: Change the Name and Description of a Catalog Item**

This request changes the name and the description of the catalog item shown in "Example: Retrieve a Catalog Item,” on page 33. The request body excludes components such as `Link` elements and `id` attributes that were present in the retrieved `CatalogItem`. These components are ignored if you include them in a request.

**Request:**

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

```
Content-Type: application/vnd.vmware.vcloud.catalogItem+xml
...
<?xml version="1.0" encoding="UTF-8"?>
<CatalogItem
    xmlns="http://www.vmware.com/vcloud/v1.5"
    name="DEPRECATED Ubuntu Template">
    <Description>Deprecated. Use https://vcloud.example.com/api/vAppTemplate/vappTemplate-230 instead</Description>
    <Entity
        href="https://vcloud.example.com/api/vAppTemplate/vappTemplate-111"
        type="application/vnd.vmware.vcloud.vAppTemplate+xml"
        name="Ubuntu Template with vsftpd" />
</CatalogItem>
```

The response shows the modified `CatalogItem`.

**Response:**

```
200 OK
Content-Type: application/vnd.vmware.vcloud.catalogItem+xml
...
<CatalogItem
    xmlns="http://www.vmware.com/vcloud/v1.5"
    name="DEPRECATED Ubuntu Template">
    <Description>Deprecated. Use https://vcloud.example.com/api/vAppTemplate/vappTemplate-230 instead</Description>
    <Entity
        href="https://vcloud.example.com/api/vAppTemplate/vappTemplate-111"
        type="application/vnd.vmware.vcloud.vAppTemplate+xml"
        name="Ubuntu vApp Template" />
</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.

Removing a CatalogItem also removes the referenced vApp template or media image from the catalog's storage.

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.
- Verify that the target catalog does not have an external subscription.

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.

Example: Remove an Item from a Catalog

This request removes the source catalog item that was copied in “Example: Copy a Catalog Item,” on page 79.

Request:
DELETE https://vcloud.example.com/api/catalogItem/221

Response:
204 No Content

Synchronize a Catalog or Catalog Item

Catalogs that have external subscriptions are synchronized with their external sources by a background process that the system administrator controls. You can also force synchronization of individual catalog items or entire catalogs at any time.

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 XML representation of a catalog that has an external subscription.
   Use a request like this one:
   GET https://vcloud.example.com/api/catalog/32
2. Examine the Catalog element to find the CatalogItem elements that it contains.
3. Examine the Catalog and CatalogItem element to find the sync links that they contain.
   In catalogs, these links have the following form:
   
   <Link
       rel="sync"
       href="https://vcloud.example.com/api/catalog/id/action/sync" />
In catalog items, these links have the following form:

```xml
<Link
  rel="sync"
  href="https://vcloud.example.com/api/catalogItem/id/action/sync" />
```

4 Synchronize the catalog or catalog item.

Make a POST request to the appropriate `action/sync` link.

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Synchronize a Catalog</td>
<td>Make a POST request to the <code>action/sync</code> link in the Catalog element.</td>
</tr>
<tr>
<td>Synchronize a Catalog Item</td>
<td>Make a POST request to the <code>action/sync</code> link in the CatalogItem element.</td>
</tr>
</tbody>
</table>

**Example: Synchronize a Catalog Item**

This request synchronizes a single catalog item. The response is a task that tracks the progress of the synchronization.

Request:

```plaintext
POST https://vcloud.example.com/api/catalogItem/102/action/sync
```

Response:

```
202 Accepted
Content-Type: application/vnd.vmware.vcloud.task+xml
...
<Task ... operation="Synchronizing Catalog Item DB.iso (102)" ...>
...
</Task>
```

**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 85.

**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 83.

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 83.

**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 185. Because optional attributes `busType` and `busSubType` are omitted, a SCSI disk is created.

Request:

```
POST https://vcloud.example.com/api/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" />
</Disk>
```

---

VMware, Inc.  83
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.

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 Vm and Disk objects that you own.
- **AdminvmDiskRelation**: Lists this information for all Vm and Disk 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 252.

Prerequisites
To change the owner of a Disk, VApp, or 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 86.

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

Controlling Access to vApps and Catalogs

Upon creation, catalogs and vApps grant full access to their owners and no access to other users. The vCloud API access control mechanism enables object owners to retrieve or update these access controls as needed.

To retrieve or update the access controls on a vApp or catalog, use controlAccess links. The controlAccess links for catalogs are included when you retrieve the containing AdminOrg. The controlAccess links for a vApp are included in the VApp element itself.

vCloud Director defines three levels of access:

- **ReadOnly**
  The ReadOnly access level grants rights to read or use the object.

- **Change**
  The Change access level includes all rights granted by ReadOnly access and grants additional rights to modify the object and its properties.

- **FullControl**
  The FullControl access level includes all rights granted by Change access and grants additional rights to change the owner of the object, share it, or delete it.

See “Access Rights to vCloud Director Objects,” on page 88 for detailed information about the rights granted by each access level.

Access Control for vApps

An administrator or vApp owner can control access to a vApp. Each VApp element includes two types of access control links:

- Links where rel="down".

  ```xml
  <Link rel="down"
    type="application/vnd.vmware.vcloud.controlAccess+xml"
    href="https://vcloud.example.com/api/vApp/vapp-id/controlAccess/"/>
  ```

  Use this kind of link to retrieve the access control settings for the vApp identified in the href value.
Links where rel="controlAccess".

Use this kind of link to specify new access control settings for the vApp 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. See “Update vApp Access Controls,” on page 116 for an example.

Access Control for Catalogs

A system administrator or organization administrator can control access to a catalog. Each Org element includes two types of access control links for the catalogs owned by the organization it represents:

Links where rel="down".

Use this kind of link to retrieve the access control settings for the catalog identified in the href value.

Links where rel="controlAccess".

Use this kind of link to specify new access control settings for the catalog 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.

Note The controlAccess links for catalogs are not returned in the AdminOrg response when you retrieve the organization in the admin view.

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 specific users, 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.

```xml
<ControlAccessParams
 xmlns="http://www.vmware.com/vcloud/v1.5">
 <IsSharedToEveryone>false</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>
```

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 85.

Access Rights to vCloud Director Objects

Each access level supported by vCloud Director grants one or more users a specific set of rights to an object. vCloud Director access levels are similar to roles in that they give a name to a set of rights. When you apply an access control to an object, you grant one or more users in your organization a set of rights to the object. Access rights are additive. You can make an object more accessible to users who have limited rights, but you cannot restrict the rights that a user may already have. For example, an organization administrator retains full control of an object even if you apply `ReadOnly` access rights to it for all organization members.

<table>
<thead>
<tr>
<th>Catalog: Add vApp from My Cloud</th>
<th>FullControl</th>
<th>Change</th>
<th>ReadOnly</th>
</tr>
</thead>
<tbody>
<tr>
<td>Catalog: Change Owner</td>
<td>X</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Catalog: VCSP Publish Subscribe</td>
<td>X</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Catalog: Edit Properties</td>
<td>X</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Access Levels and the Rights They Grant (Continued)</td>
<td>FullControl</td>
<td>Change</td>
<td>ReadOnly</td>
</tr>
<tr>
<td>---------------------------------------------------</td>
<td>-------------</td>
<td>--------</td>
<td>---------</td>
</tr>
<tr>
<td>Catalog: Publish</td>
<td>X</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Catalog: View Private and Shared Catalogs</td>
<td>X</td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>Catalog: View Published Catalogs</td>
<td>X</td>
<td></td>
<td>X</td>
</tr>
<tr>
<td>vApp Template or Media: Copy</td>
<td>X</td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>vApp Template or Media: Create or Upload</td>
<td>X</td>
<td></td>
<td></td>
</tr>
<tr>
<td>vApp Template or Media: Edit</td>
<td>X</td>
<td></td>
<td></td>
</tr>
<tr>
<td>vApp Template or Media: View</td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>vApp Template: Checkout (Add to My Cloud)</td>
<td>X</td>
<td></td>
<td></td>
</tr>
<tr>
<td>vApp Template: Download</td>
<td>X</td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>vApp: Change Owner</td>
<td>X</td>
<td></td>
<td></td>
</tr>
<tr>
<td>vApp: Copy</td>
<td>X</td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>vApp: Create or Reconfigure</td>
<td>X</td>
<td></td>
<td></td>
</tr>
<tr>
<td>vApp: Delete</td>
<td>X</td>
<td></td>
<td></td>
</tr>
<tr>
<td>vApp: Edit Properties</td>
<td>X</td>
<td></td>
<td></td>
</tr>
<tr>
<td>vApp: Edit VM CPU</td>
<td>X</td>
<td></td>
<td></td>
</tr>
<tr>
<td>vApp: Edit VM Hard Disk</td>
<td>X</td>
<td></td>
<td></td>
</tr>
<tr>
<td>vApp: Edit VM Memory</td>
<td>X</td>
<td></td>
<td></td>
</tr>
<tr>
<td>vApp: Edit VM Network</td>
<td>X</td>
<td></td>
<td></td>
</tr>
<tr>
<td>vApp: Edit VM Properties</td>
<td>X</td>
<td></td>
<td></td>
</tr>
<tr>
<td>vApp: Manage VM Password Settings</td>
<td>X</td>
<td></td>
<td></td>
</tr>
<tr>
<td>vApp: Power Operations</td>
<td>X</td>
<td></td>
<td></td>
</tr>
<tr>
<td>vApp: Sharing</td>
<td>X</td>
<td></td>
<td></td>
</tr>
<tr>
<td>vApp: Use Console</td>
<td>X</td>
<td>X</td>
<td></td>
</tr>
</tbody>
</table>
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.
About DMTF, CIM, and RASD

Virtual hardware in OVF packages elements is defined using an open standard framework established by the Distributed Management Task Force (DMTF). This framework, called the Common Information Model (CIM), defines virtual hardware resources using the ResourceAllocationSettingData (RASD) schema. In this schema, each class of virtual hardware is represented as an Item element with a specific ResourceType. Many vCloud API operations that deploy and configure vApps and virtual machines require you to understand and sometimes modify RASD Item elements.

You can download the RASD schema files and related information from http://www.dmtf.org/standards/cim/cim_schema_v2191

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 catalog.
- 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.
This chapter includes the following topics:

- “Create a vApp From a Template,” on page 96
- “Create a vApp From an OVF Package,” on page 104
- “Compose a vApp From Existing Virtual Machines,” on page 107
- “Recompose a vApp to Add or Remove Virtual Machines,” on page 110
- “Clone a vApp,” on page 112
- “Capture a vApp as a Template,” on page 114
- “Update vApp Access Controls,” on page 116
- “Provide User Input Requested by a Virtual Machine,” on page 117
- “Attach or Detach an Independent Disk,” on page 118
- “Creating and Using vApp Snapshots,” on page 120
- “About Virtual Machine Metrics,” on page 121
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>
<thead>
<tr>
<th>Operation</th>
<th>Request</th>
<th>Request Body</th>
<th>Response</th>
</tr>
</thead>
<tbody>
<tr>
<td>Create a vApp from an OVF package</td>
<td>POST <code>API-URL/vdc/id/action/instantiateOvf</code></td>
<td>InstantiateOvfParams</td>
<td>VApp</td>
</tr>
<tr>
<td>Instantiate a vApp template to create a vApp</td>
<td>POST <code>API-URL/vdc/id/action/instantiateVAppTemplate</code></td>
<td>InstantiateVAppTemplateParams</td>
<td>VApp</td>
</tr>
<tr>
<td>Change the name or description of a vApp.</td>
<td>PUT <code>API-URL/vApp/vapp-id</code></td>
<td>VApp</td>
<td>Task</td>
</tr>
<tr>
<td>Compose a vApp</td>
<td>POST <code>API-URL/vdc/id/action/composeVApp</code></td>
<td>ComposeVAppParams</td>
<td>VApp</td>
</tr>
<tr>
<td>Capture a vApp as a Template</td>
<td>POST <code>API-URL/catalog/id/action/captureVApp</code></td>
<td>CaptureVAppParams</td>
<td>VAppTemplate</td>
</tr>
<tr>
<td>Clone a vApp</td>
<td>POST <code>API-URL/vdc/id/action/cloneVApp</code></td>
<td>CloneVAppParams</td>
<td>VApp</td>
</tr>
<tr>
<td>Recompose a vApp to add or remove virtual machines</td>
<td>POST <code>API-URL/vApp/vapp-id/action/recomposeVApp</code></td>
<td>RecomposeVAppParams</td>
<td>Task</td>
</tr>
<tr>
<td>Deploy a vApp or Virtual Machine</td>
<td>POST <code>VApp-or-Vm-URL/action/deploy</code></td>
<td>DeployVAppParams</td>
<td>Task</td>
</tr>
<tr>
<td>Undeploy a vApp or Virtual Machine</td>
<td>POST <code>VApp-or-Vm-URL/action/undeploy</code></td>
<td>UndeployVAppParams</td>
<td>Task</td>
</tr>
<tr>
<td>Power On a vApp or Virtual Machine</td>
<td>POST <code>VApp-or-Vm-URL/power/action/powerOn</code></td>
<td>None</td>
<td>Task</td>
</tr>
<tr>
<td>Operation</td>
<td>Request</td>
<td>Request Body</td>
<td>Response</td>
</tr>
<tr>
<td>-------------------------------------------------------</td>
<td>----------------------------------------------</td>
<td>-----------------------</td>
<td>--------------</td>
</tr>
<tr>
<td>Power Off a vApp or Virtual Machine</td>
<td>POST VApp-or-Vm-URL/power/action/powerOff</td>
<td>None</td>
<td>Task</td>
</tr>
<tr>
<td>Reset a vApp or Virtual Machine</td>
<td>POST VApp-or-Vm-URL/action/reset</td>
<td>None</td>
<td>Task</td>
</tr>
<tr>
<td>Suspend a vApp or Virtual Machine</td>
<td>POST VApp-or-Vm-URL/action/suspend</td>
<td>None</td>
<td>Task</td>
</tr>
<tr>
<td>Discard the Suspended State of a vApp or Virtual Machine</td>
<td>POST VApp-or-Vm-URL/action/discardSuspendedState</td>
<td>None</td>
<td>Task</td>
</tr>
<tr>
<td>Shut Down a vApp or Virtual Machine</td>
<td>POST VApp-or-Vm-URL/action/shutdown</td>
<td>None</td>
<td>Task</td>
</tr>
<tr>
<td>Reboot a vApp or Virtual Machine</td>
<td>POST VApp-or-Vm-URL/action/reboot</td>
<td>None</td>
<td>Task</td>
</tr>
<tr>
<td>Retrieve product sections of a vApp or virtual machine</td>
<td>GET VApp-or-Vm-URL/productSections</td>
<td>None</td>
<td>ProductSectionList</td>
</tr>
<tr>
<td>Update product sections of a vApp or virtual machine</td>
<td>PUT VApp-or-Vm-URL/productSections</td>
<td>ProductSectionList</td>
<td>Task</td>
</tr>
<tr>
<td>Retrieve product sections of a vApp template</td>
<td>GET API-URL/vAppTemplate/vappTemplate-id/productSections</td>
<td>None</td>
<td>ProductSectionList</td>
</tr>
<tr>
<td>Retrieve version of VMware Tools installed on a virtual machine</td>
<td>GET Vm-URL/runtimeInfoSection</td>
<td>None</td>
<td>RuntimeInfoSection</td>
</tr>
<tr>
<td>Install VMware Tools on a virtual machine</td>
<td>POST Vm-URL/action/installVMwareTools</td>
<td>None</td>
<td>Task</td>
</tr>
<tr>
<td>Consolidate a virtual machine</td>
<td>POST Vm-URL/action/consolidate</td>
<td>None</td>
<td>Task</td>
</tr>
<tr>
<td>Upgrade the hardware version of a virtual machine</td>
<td>POST Vm-URL/action/upgradeHardwareVersion</td>
<td>None</td>
<td>Task</td>
</tr>
<tr>
<td>Insert Media Into a Virtual Machine</td>
<td>POST Vm-URL/action/insertMedia</td>
<td>MediaInsertOrEjectParams</td>
<td>Task</td>
</tr>
<tr>
<td>Eject Media from a Virtual Machine</td>
<td>POST Vm-URL/action/ejectMedia</td>
<td>MediaInsertOrEjectParams</td>
<td>Task</td>
</tr>
<tr>
<td>List Media Devices of a Virtual Machine</td>
<td>GET Vm-URL/virtualHardwareSection/media</td>
<td>None</td>
<td>RasdItemsList</td>
</tr>
<tr>
<td>Get a Request for User Input</td>
<td>GET Vm-URL/question</td>
<td>None</td>
<td>VmPendingQuestion</td>
</tr>
<tr>
<td>Provide Requested User Input</td>
<td>POST Vm-URL/question/action/answer</td>
<td>VmQuestionAnswer</td>
<td>204 No Content</td>
</tr>
<tr>
<td>Get a Screen Thumbnail for a Virtual Machine</td>
<td>GET Vm-URL/screen</td>
<td>None</td>
<td>Returns a screen thumbnail (Content-type: image/png) if one is available. Otherwise returns null (Content-Length: 0).</td>
</tr>
</tbody>
</table>
### Table 5-1. Summary of vApp and Virtual Machine Operations Requests (Continued)

<table>
<thead>
<tr>
<th>Operation</th>
<th>Request</th>
<th>Request Body</th>
<th>Response</th>
</tr>
</thead>
<tbody>
<tr>
<td>Get a Screen Ticket for a Virtual Machine</td>
<td>POST Vm-URL/screen/action/acquire Ticket</td>
<td>None</td>
<td>ScreenTicket</td>
</tr>
<tr>
<td>Attach an independent disk to a virtual machine.</td>
<td>POST Vm-URL/disk/action/attach</td>
<td>DiskAttachOrDetachParams</td>
<td>Task</td>
</tr>
<tr>
<td>Detach an independent disk from a virtual machine.</td>
<td>POST Vm-URL/disk/action/detach</td>
<td>DiskAttachOrDetachParams</td>
<td>Task</td>
</tr>
<tr>
<td>Create snapshots of all virtual machines in a vApp.</td>
<td>POST API-URL/vApp/vapp-id/action/createSnapshot</td>
<td>CreateSnapshotParams</td>
<td>Task</td>
</tr>
<tr>
<td>Remove snapshots of all virtual machines in a vApp.</td>
<td>POST API-URL/vApp/vapp-id/action/removeAllSnapshots</td>
<td>None</td>
<td>Task</td>
</tr>
<tr>
<td>Revert all virtual machines in a vApp to their current snapshots.</td>
<td>POST API-URL/vApp/vapp-id/action/revertToCurrentSnapshot</td>
<td>None</td>
<td>Task</td>
</tr>
<tr>
<td>Retrieve information about vApp snapshots.</td>
<td>GET API-URL/vApp/vapp-id/snapshotSection</td>
<td>None</td>
<td>SnapshotSection</td>
</tr>
<tr>
<td>Validate the storage profile compliance of a virtual machine.</td>
<td>POST API-URL/vApp/vm-id/action/checkCompliance</td>
<td>None</td>
<td>Task</td>
</tr>
<tr>
<td>Retrieve a virtual machine storage profile validation report.</td>
<td>GET API-URL/vApp/vm-id/complianceResult</td>
<td>None</td>
<td>ComplianceResult</td>
</tr>
<tr>
<td>Retrieve all current metrics for a virtual machine. [NEW]</td>
<td>GET Vm-URL/metrics/current</td>
<td>None</td>
<td>CurrentUsage</td>
</tr>
<tr>
<td>Retrieve all historic metrics for a virtual machine. [NEW]</td>
<td>GET Vm-URL/metrics/historic</td>
<td>None</td>
<td>HistoricUsage</td>
</tr>
<tr>
<td>Retrieve a subset of current metrics for a virtual machine. [NEW]</td>
<td>POST Vm-URL/metrics/current</td>
<td>CurrentUsageSpec</td>
<td>CurrentUsage</td>
</tr>
<tr>
<td>Retrieve a subset of historic metrics for a virtual machine. [NEW]</td>
<td>POST Vm-URL/metrics/historic</td>
<td>HistoricUsageSpec</td>
<td>HistoricUsage</td>
</tr>
</tbody>
</table>

### Create a vApp From a Template

An instantiateVAppTemplate request creates a vApp from a vApp template.

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 35. 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 NetworkConfigSection that defines the vApp networks to which virtual machines in this vApp can connect.
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 an administrator, Catalog Author, or vApp Author.
- Review the current configuration of the vApp and its virtual machines. Configuration parameters in the VAppTemplate such as its NetworkConfigSection and LeaseSettingsSection affect all virtual machines in the vApp. Configuration parameters for individual virtual machines are defined in Vm elements in the VAppTemplate. A virtual machine’s network connections are defined in its NetworkConnectionSection, and its hardware configuration is defined in its VirtualHardwareSection. To view the configuration of a virtual machine in a vApp template, you can retrieve the template and examine the VirtualHardwareSection of that Vm, or you can download the OVF descriptor of the vApp template, as shown in “Download the OVF Descriptor of a vApp or vApp Template,” on page 72.

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. You can also use the query service to return a list of references to vApp templates that you can access.
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 97 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 the vApp 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 36 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 133.

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 36.

Modify Virtual Machine Hardware During vApp Template Instantiation

An InstantiateVAppTemplateParams request body can include a SourcedVmInstantiationParams element for each virtual machine in the vApp. Values you provide in this element modify a subset of virtual machine hardware configuration parameters during instantiation.

An InstantiateVAppTemplateParams request body that incorporates one or more SourcedVmInstantiationParams elements supports the following configuration changes:

- Specify a storage profile for the virtual machine.
- Increase the capacity of the virtual machine’s SATA or SCSI disks.
Increase or decrease the size of the virtual machine's memory.
Increase or decrease the number of CPU cores per virtual socket.
Add or remove CPUs.

You can also modify any of these configuration settings after the vApp is deployed. See “Configuring vApps and Virtual Machines,” on page 133

Prerequisites
- Verify that you are logged in to the vCloud API as an administrator, Catalog Author, or vApp Author.
- Review the current configuration of the vApp and its virtual machines. Configuration parameters in the VAppTemplate such as its NetworkConfigSection and LeaseSettingsSection affect all virtual machines in the vApp. Configuration parameters for individual virtual machines are defined in Vm elements in the VAppTemplate. A virtual machine's network connections are defined in its NetworkConnectionSection, and its hardware configuration is defined in its VirtualHardwareSection. To view the configuration of a virtual machine in a vApp template, you can retrieve the template and examine the VirtualHardwareSection of that Vm, or you can download the OVF descriptor of the vApp template, as shown in “Download the OVF Descriptor of a vApp or vApp Template,” on page 72.

Procedure
1. Examine the OVF descriptor of the template to determine the values that you can include in a SourcedVmInstantiationParams element.
2. Include the SourcedVmInstantiationParams element in an InstantiateVAppTemplateParams element.
3. Make a POST request to the action/instantiateVAppTemplate URL of the VDC.
   Supply the InstantiateVAppTemplateParams element as the request body.

Example: Modify Virtual Machine Hardware During vApp Template Instantiation

IMPORTANT This example uses the SourcedVmInstantiationParams element, which is deprecated as of API version 5.6 and might be removed from future releases. Use the SourcedItem element, as shown in “Example: Modify Virtual Machine Name, Description, and Hardware During vApp Template Instantiation,” on page 102, instead.

This InstantiateVAppTemplateParams request extends the one shown in “Example: Instantiate a vApp Template,” on page 97 to include a SourcedVmInstantiationParams element that makes several configuration changes in the virtual machine referenced at https://vcloud.example.com/api/vAppTemplate/vm-4.

- Specifies a storage profile for the virtual machine.
- Adds a virtual CPU and changes the value of CoresPerSocket to 2. If you include a CoresPerSocket element, its value must be an integer multiple of the value of the existing rasd:VirtualQuantity of CPU items, or of the value you supply in NumberOfCpus. “Example: Modify the CPU Configuration of a Virtual Machine,” on page 153 shows the original CPU configuration, and how to make this change by reconfiguring the virtual machine in a deployed vApp.
- Increases the capacity of the hard disk from 1GB to 10GB by including a Disk element that specifies a Size of 10240 for the disk that has a rasd:InstanceID value of 2000. The value you supply for Size is interpreted as megabytes. You can see the original disk configuration in “Example: Retrieve the Hard Disks and Controllers in a Virtual Machine,” on page 159. “Example: Modify the Hard Disk Configuration of a Virtual Machine,” on page 161 shows how to make the same change by reconfiguring the virtual machine in a deployed vApp. If you include a Disk element, the value of its
instanceId attribute must match the value in the rasd:InstanceID element of an existing Item that defines a virtual disk (RASD resource type 17). You can increase disk capacity, but not reduce it, for disks on SATA and SCSI controllers. You cannot change the capacity of other disk types. Item elements that represent SATA disks have a vcloud:busType attribute with the value 20. Those that represent SCSI disks have a vcloud:busType attribute with the value 6.

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</networkName>
        <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>2013-04-11T08:08:16.438-07:00</StorageLeaseExpiration>
    </LeaseSettingsSection>
  </InstantiationParams>
  <Source href="https://vcloud.example.com/api/vAppTemplate/vappTemplate-111" />
  <SourcedVmInstantiationParams>
    <Source href="https://vcloud.example.com/api/vAppTemplate/vm-4" />
    <StorageProfile href="https://vcloud.example.com/api/vdcStorageProfile/33" />
    <HardwareCustomization>
      <NumberOfCpus>2</NumberOfCpus>
      <CoresPerSocket>2</CoresPerSocket>
      <Disk instanceId="2000">
        <Size>10240</Size>
      </Disk>
    </HardwareCustomization>
  </SourcedVmInstantiationParams>
</InstantiateVAppTemplateParams>
Modify Virtual Machine Name, Description, Hardware, and Other Properties During vApp Template Instantiation

To allow greater flexibility when modifying template properties during instantiation, use SourcedItem instead of SourcedVmInstantiationParams when specifying instantiation details in InstantiateVAppTemplateParams.

An InstantiateVAppTemplateParams request body that incorporates one or more SourcedItem elements supports the following configuration changes:

- Change the name, Description, and NeedsCustomization properties of the virtual machine.
- Specify a storage profile for the virtual machine.
- Specify a storage profile for any of the virtual machine's hard disks, overriding the virtual machine's default storage profile.
- Increase the capacity of the virtual machine's SATA or SCSI disks.
- Increase or decrease the size of the virtual machine's memory.
- Increase or decrease the number of CPU cores per virtual socket.
- Add or remove CPUs.

You can also modify any of these configuration settings after the vApp is deployed. See “Configuring vApps and Virtual Machines,” on page 133.

Prerequisites

- Verify that you are logged in to the vCloud API as an administrator, Catalog Author, or vApp Author.
- Review the current configuration of the vApp and its virtual machines. Configuration parameters in the VAppTemplate such as its NetworkConfigSection and LeaseSettingsSection affect all virtual machines in the vApp. Configuration parameters for individual virtual machines are defined in Vm elements in the VAppTemplate. A virtual machine's network connections are defined in its NetworkConnectionSection, and its hardware configuration is defined in its VirtualHardwareSection. To view the configuration of a virtual machine in a vApp template, you can retrieve the template and examine the VirtualHardwareSection of that Vm, or you can download the OVF descriptor of the vApp template, as shown in “Download the OVF Descriptor of a vApp or vApp Template,” on page 72.

Procedure

1. To change the name, Description, or NeedsCustomization properties of the virtual machine, add a VmGeneralParams element to the SourcedItem.
2. Examine the OVF descriptor of the template to determine the values that you can include in the VirtualHardwareSection of the SourcedItem element.
3. Include the SourcedItem element in an InstantiateVAppTemplateParams element.
4. Make a POST request to the action/instantiateVAppTemplate URL of the VDC.
   Supply the InstantiateVAppTemplateParams element as the request body.
Example: Modify Virtual Machine Name, Description, and Hardware During vApp Template Instantiation

This InstantiateVAppTemplateParams request extends the request shown in “Example: Instantiate a vApp Template,” on page 97 to include a SourcedItem element that makes several configuration changes in the virtual machine referenced at https://vcloud.example.com/api/vAppTemplate/vm-4.

- Changes the name and Description of the virtual machine, so that they are not the same as the name and Description in the template.
- Sets the value of NeedsCustomization to true.
- Adds a virtual CPU and changes the value of CoresPerSocket to 2. If you include a CoresPerSocket element, its value must be an integer multiple of the value of the existing rasd:VirtualQuantity of CPU items, or of the value you supply in NumberOfCpus. “Example: Modify the CPU Configuration of a Virtual Machine,” on page 153 shows the original CPU configuration, and how to make this change by reconfiguring the virtual machine in a deployed vApp.
- Increases the capacity of the hard disk from 1GB to 10GB by including a Disk element that specifies a Size of 10240 for the disk that has a rasd:InstanceID value of 2000. The value you supply for Size is interpreted as megabytes. You can see the original disk configuration in “Example: Retrieve the Hard Disks and Controllers in a Virtual Machine,” on page 159. “Example: Modify the Hard Disk Configuration of a Virtual Machine,” on page 161 shows how to make the same change by reconfiguring the virtual machine in a deployed vApp. If you include a Disk element, the value of its instanceId attribute must match the value in the rasd:InstanceID element of an existing Item that defines a virtual disk (RAID resource type 17). Disk capacity can be raised, but not lowered, for disks on SATA and SCSI controllers. The capacity of other disk types cannot be changed. Item elements that represent SATA disks have a vcloud:busType attribute with the value 20. Those that represent SCSI disks have a vcloud:busType attribute with the value 6.

With the exception of the changes specified in VmGeneralParams (name, Description, and NeedsCustomization), this example has the same effect as the example in “Example: Modify Virtual Machine Hardware During vApp Template Instantiation,” on page 99.

**IMPORTANT** You cannot mix SourcedVmInstantiationParams and SourcedItem elements in a single InstantiateVAppTemplateParams request body.

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>
<source href="https://vcloud.example.com/api/vAppTemplate/vappTemplate-111"/>
</Source>
</SourcedItem>
<Source href="https://vcloud.example.com/api/vAppTemplate/vm-4"/>
</VmGeneralParams>
<Name>ftp1</Name>
<Description>Primary FTP Server Instance</Description>
<NeedsCustomization>true</NeedsCustomization>
```
<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>2013-11-01T08:00:22.001-07:00</StorageLeaseExpiration>
</LeaseSettingsSection>

<ovf:VirtualHardwareSection xmlns:ovf="http://schemas.dmtf.org/ovf/envelope/1"
  xmlns:rasd="http://schemas.dmtf.org/wbem/wscim/1/cim-schema/2/CIM_ResourceAllocationSettingData"
  xmlns:vmw="http://www.vmware.com/schema/ovf"
  xmlns:vcloud="http://www.vmware.com/vcloud/v1.5"
  xmlns:vssd="http://schemas.dmtf.org/wbem/wscim/1/cim-schema/2/CIM_VirtualSystemSettingData"
  vcloud:href="https://vcloud.example.com/api/vApp/vm-4/virtualHardwareSection/"
  vcloud:type="application/vnd.vmware.vcloud.virtualHardwareSection+xml">
  <ovf:Info>Virtual hardware requirements</ovf:Info>
  <ovf: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="lsilogicsas"
      vcloud:busType="6">
      <rasd:InstanceID>2000</rasd:InstanceID>
      <rasd:ResourceType>17</rasd:ResourceType>
    </rasd:HostResource>
    <rasd:InstanceID>41</rasd:InstanceID>
    <rasd:Reservation>0</rasd:Reservation>
    <rasd:ResourceType>3</rasd:ResourceType>
    <rasd:VirtualQuantity>2</rasd:VirtualQuantity>
  </ovf:Item>
  </ovf:VirtualHardwareSection>
</InstantiationParams>
Create a vApp From an OVF Package

An instantiateOvf request uploads an OVF package and then creates a vApp from it. By default, this operation also deploys the vApp and powers it on.

If you want to deploy an OVF package as a vApp without creating a vApp template and corresponding catalog item, make an instantiateOvf request. This request initiates a workflow similar to the one shown in “Upload an OVF Package to Create a vApp Template,” on page 60, but with a few important differences.

- The target of the upload is a VDC, not a catalog.
- The request body is an InstantiateOvfParams element, not an UploadVAppTemplateParams element.
- The response is a VApp element that includes an upload URL for the OVF descriptor.

After you retrieve the VApp element created by the instantiateOvf request, you can upload the descriptor and the files it references.

Prerequisites

Verify that the following are true:

- You have an OVF package to upload.
- You are logged in as a user who has permission to upload OVF packages and create vApps.
- 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.

Review the contents of the Envelope element in the descriptor file of your OVF package. Several properties in this file have implications for the InstantiateOvfParams request body you must construct to initiate the upload.

Procedure

1. Retrieve the XML representation of the target VDC.
2. Examine the response to locate the Link element that contains the URL for creating a vApp from an OVF package.
   - This element has a rel attribute value of add and a type attribute value of application/vnd.vmware.vcloud.instantiateOvfParams+xml, as shown here:
     ```xml
     <link
       rel="add"
       type="application/vnd.vmware.vcloud.instantiateOvfParams+xml"
       href="https://vcloud.example.com/api/vdc/44/action/instantiateOvf" />
     ```
3. Create an InstantiateOvfParams request body.
   - See the request portion of “Example: Create a vApp From an OVF Package,” on page 105.
4. POST the InstantiateOvfParams to the instantiateOvf URL you retrieved in Step 2
   - See the request portion of “Example: Create a vApp From an OVF Package,” on page 105.
5. Examine the response.

The response, a VApp element, contains a File element that specifies an upload URL for the OVF descriptor. See the response portion of “Example: Create a vApp From an OVF Package,” on page 105.

6. Upload the OVF descriptor.

Make a PUT request to the upload URL in the VApp. The upload URL for the OVF descriptor is in a Link element with the following form:

```
<Link rel='upload:default' href='https://vcloud.example.com/transfer/.../descriptor.ovf' />
```

Supply the OVF descriptor as the request body. The OVF descriptor contains a single Envelope element.

7. Retrieve the remaining upload URLs

a. Retrieve the VApp.

b. Verify that the value of the ovfDescriptorUploaded attribute is true.

c. Examine the VApp to find the upload URLs for the files referenced in the OVF descriptor.

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

8. Upload the referenced files.

You can follow the progress of the upload by retrieving the vApp and noting the progress of the embedded Task.

After all of the referenced files are uploaded, the VApp element no longer includes an embedded Task. The vApp is placed in the power and deployment state that the values of the powerOn and deploy attributes specify in the request body.

Example: Create a vApp From an OVF Package

This request includes a NetworkMapping element that maps a network name found in the uploaded OVF descriptor to the name of a network available in the target VDC.

Request:

```
POST https://vcloud.example.com/api/vdc/5/action/instantiateOvf
Content-Type: application/vnd.vmware.vcloud.instantiateOvfParams+xml

<?xml version="1.0" encoding="UTF-8"?>
<InstantiateOvfParams
  xmlns:ovf="http://schemas.dmtf.org/ovf/envelope/1"
  xmlns="http://www.vmware.com/vcloud/v1.5"
  name="W2K8">
  <Description>Example vApp</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>
</InstantiateOvfParams>
```
The response is a sparsely populated VApp element that includes an upload URL for the OVF descriptor. See "Uploading Referenced Files," on page 67 for file upload procedures.

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="W2K8"
type="application/vnd.vmware.vcloud.vApp+xml"
href="https://vcloud.example.com/api/vApp/vapp-23">
<Link
rel="up"
type="application/vnd.vmware.vcloud.vApp+xml"
href="https://vcloud.example.com/api/vdc/5"/>
<Description>Example vApp</Description>
<Link
rel="remove"
href="https://vcloud.example.com/api/vApp/vapp-23" />
<Description>Example vApp</Description>
<Tasks>
<Task
status="running"
operation="Creating Virtual Application W2K8(23)"
... />
...
</Task>
</Tasks>
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 133. 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 134.
  - 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:

   ```xml
   <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 108.

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 315.

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>
                <ParentNetwork
                    href="https://vcloud.example.com/api/network/54"/>
                <Configuration>
                    <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:Info />
    <PrimaryNetworkConnectionIndex>0</PrimaryNetworkConnectionIndex>
    <NetworkConnection
        network="CRMApplianceNetwork">
        <NetworkConnectionIndex>0</NetworkConnectionIndex>
        <IsConnected>true</IsConnected>
        <IpAddressAllocationMode>DHCP</IpAddressAllocationMode>
    </NetworkConnection>
</NetworkConnectionSection>
</InstantiationParams>
</SourcedItem>
</SourcedItem>
</SourcedItem>
<AllEULAsAccepted>true</AllEULAsAccepted>
</ComposeVAppParams>

The response is a sparsely populated VApp element in the target VDC. When the Task embedded in the response is complete, the vApp has been composed.

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 and CreateItem 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 110.

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.

```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"/>
  ...
  <Children>
    <Vm
      status="8"
      name="CRM-DB"
      href="https://vcloud.example.com/api/vApp/vm-7" ...>
      ...
    </Vm>
    <Vm
      status="8"
```
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:

```plaintext
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>
```
Clone a vApp

You can make a copy of a vApp by cloning it. If the vApp is deployed when you clone it, the clone procedure also clones the memory state of the virtual machines in the vApp.

The cloneVApp request makes a copy of the vApp referenced in the Source element of the CloneVappParams 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 or renames the vApp.

If the vApp is deployed when you clone it and the target VDC is backed by the same provider VDC as the source VDC, the clone is created with the following properties:

- Memory state of all virtual machines in the source vApp is preserved in the clone.
- The clone is suspended and connected to an isolated network.

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 VDC in which you want to deploy the cloned vApp.

   Use a request like this one, where id is the identifier of the VDC:

   GET https://vcloud.example.com/admin/vdc/id

   You can create the clone in the same VDC that holds the source vApp, or in a different VDC.

2. Examine the response to locate the Link element that contains the URL for cloning a vApp.

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

   <Link
      rel="add"
      type="application/vnd.vmware.vcloud.captureVAppParams+xml"
      href="https://vcloud.example.com/api/vdc/12/action/cloneVApp"/>
3 Create a CloneVappParams element that references the vApp to clone and specifies details of the clone operation.

See “Example: Clone a vApp,” on page 113.

4 POST the CloneVappParams element to the action/cloneVApp link of the VDC in which the clone should be created.

Example: Clone a vApp

This request creates a copy of the vApp created in “Example: Instantiate a vApp Template,” on page 97 in another VDC. Because the ParentNetwork in the Source vApp is not available in the VDC specified in the action/cloneVApp request, the CloneVappParams request body must include InstantiationParams that specify a new ParentNetwork for the vApp network.

---

**N**ote: If the vApp is deployed when you clone it, any network configuration you specify in the CloneVappParams is ignored and the clone is created with a connection to an isolated network.

---

Request:

```
POST https://vcloud.example.com/api/vdc/12/action/cloneVApp
Content-Type: application/vnd.vmware.vcloud.cloneVAppParams+xml

<?xml version="1.0" encoding="UTF-8"?>
<CloneVappParams
  xmlns="http://www.vmware.com/vcloud/v1.5"
  xmlns:ovf="http://schemas.dmtf.org/ovf/envelope/1"
  name="cloned"
  deploy="false"
  powerOn="false">
  <Description>Cloned vApp Example</Description>
  <InstantiationParams>
    <NetworkConfigSection>
      <ovf:Info>Configuration parameters for logical networks</ovf:Info>
      <NetworkConfig
        networkName="vAppNetwork">
        <Configuration>
          <ParentNetwork
            href="https://vcloud.example.com/api/network/12" />
          <FenceMode>bridged</FenceMode>
        </Configuration>
      </NetworkConfig>
    </NetworkConfigSection>
  </InstantiationParams>
  <Source href="https://vcloud.example.com/api/vApp/vapp-7" />
  <IsSourceDelete>false</IsSourceDelete>
</CloneVappParams>
```

The response is a sparsely populated VApp element in the target VDC. When the Task embedded in the response is complete, the vApp has been cloned.

Response:

```
201 Created
Content-Type: application/vnd.vmware.vcloud.vApp+xml

< VApp
```
Capture a vApp as a Template

You can capture a vApp to create a vApp template in a catalog. Instantiating the resulting template recreates the vApp from which it was captured.

The captureVApp request creates a template based on the vApp referenced in the Source element of the CaptureVAppParams request body. The request specifies a new name and, optionally, a new description and storage profile for the template. If you want the new template to overwrite an existing template in the catalog, you can specify a TargetCatalogItem in the request. Otherwise, the new template is stored in a new catalog item.

If the vApp is deployed when you capture it, the template is created with the following properties.

- Memory state of all virtual machines in the source vApp is preserved in the template.
- Instantiating the template always creates a suspended vApp connected to an isolated network.

**Note** If the template is instantiated in a VDC that is not backed by the provider VDC that backed the VDC in which the vApp was captured, memory state in the template is discarded on instantiation, and the vApp is created with the network connections defined in the template or instantiation parameters.

**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.
- Verify that the target catalog does not have an external subscription.
Procedure

1. Retrieve the XML representation of the catalog to which to add the vApp template that the capture creates.

   Use a request like this one, where id is the identifier of the catalog:

   GET https://vcloud.example.com/api/catalog/id

2. Examine the response to locate the Link element that contains the URL for capturing a vApp.

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

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

3. Create a CaptureVAppParams element that references the vApp to capture.

   See “Example: Capture a vApp,” on page 115.

4. POST the CaptureVAppParams element to the action/captureVApp link shown in Step 2.

Example: Capture a vApp

This request captures the vApp created in “Example: Compose a vApp,” on page 108. Because the request does not specify a TargetCatalogItem, a new catalog item is created for the new template.

Request:

POST https://vcloud.example.com/api/catalog/5/action/captureVApp
Content-Type: application/vnd.vmware.vcloud.captureVAppParams+xml

...<xml version="1.0" encoding="UTF-8"?>
   <CaptureVAppParams
      xmlns="http://www.vmware.com/vcloud/v1.5"
      name="Example Corp's CRM Appliance">
      <Description>Captured CRM Appliance</Description>
      <Source href="https://vcloud.example.com/api/vApp/vapp-33" />
   </CaptureVAppParams>

The response is a sparsely populated VApp element in the target VDC. It contains a Link to the catalog specified in the request. When the Task embedded in the response is complete, the vApp has been captured and a vApp template created in the target catalog.

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"
   href="https://vcloud.example.com/api/vAppTemplate/vappTemplate-232">
   ...
   <Link rel="up" type="application/vnd.vmware.vcloud.catalog+xml"
      href="https://vcloud.example.com/api/catalog/5"/>

<ovf:Envelope
   xmlns:ovf="http://schemas.dmtf.org/ovf/envelope/1"
   xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
   xmlns:xsd="http://www.w3.org/2001/XMLSchema">
   ...
   <ovf:References>
      ...
   </ovf:References>
</ovf:Envelope>
Update vApp Access Controls

A system or organization administrator can use the controlAccess links in a VApp element to grant or restrict access to the VApp.

A VApp initially grants full access to its owner and no access to other users. The vCloud API access control mechanism enables an administrator to retrieve or update VApp access controls to add or remove rights for all users, or for individual users. For a general discussion of access controls in vCloud Director, see “Controlling Access to vApps and Catalogs,” on page 86.

Procedure

1. Retrieve the XML representation of the vApp.
   
   Use a request like this one:
   
   GET https://vcloud.example.com/api/vApp/vapp-7

2. Examine the VApp element to find the controlAccess links that it contains.

3. Create a ControlAccessParams element request body that specifies the details of the update.

4. POST the ControlAccessParams element to the action/controlAccess link that you retrieved in Step 1.

Example: Update vApp Access Controls

This request updates the access controls of a VApp to grant full control to one user and read-only access to another user. The request body, a ControlAccessParams element, specifies a value of false for the IsSharedToEveryone element, and contains an AccessSetting element for each user whose access rights are being modified. Each user is identified by a reference to a User object. See “User and Group Administration,” on page 243. The response, a subset of which appears in this example, echoes the request.

Request:

POST https://vcloud.example.com/api/vApp/vapp-7/action/controlAccess

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

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

<ControlAccessParams
    xmlns="http://www.vmware.com/vcloud/v1.5">
    <IsSharedToEveryone>false</IsSharedToEveryone>
    <AccessSettings>
        <AccessSetting>
            <Subject
                type="application/vnd.vmware.admin.user+xml"
                href="https://vcloud.example.com/api/admin/user/40"/>
            <AccessLevel>FullControl</AccessLevel>
        </AccessSetting>
    </AccessSettings>
</ControlAccessParams>
Provision 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.

Procedure

1. Find the question link in the target vApp.

   This link has the following form:

   ```xml
   <Link
       rel="down"
       type="application/vnd.vmware.vcloud.vmPendingQuestion+xml"
   ```

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.


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"
   <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"

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 82.

**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 82 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/$7"
 ... status="running"
```
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.
- Verify that you are logged in as a user whose role includes the `vApp: Snapshot Operations` right.

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 120.

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 their 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:

```plaintext
POST https://vcloud.example.com/api/vApp/vapp-33/action/createSnapshot
Content-Type: application/vnd.vmware.vcloud.createSnapshotParams+xml
```

... 

```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`. 

VMware, Inc.
Response:

```xml
<?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.

### About Virtual Machine Metrics

Virtual machine metrics provide information about a virtual machine's performance and resource consumption in several categories.

vCloud Director collects metrics for all running virtual machines. You can retrieve virtual machine metrics either as an instantaneous reading taken when the request is made, or as a collection of historic readings taken at regular intervals. Historic metrics are retained for 14 days, and can be retrieved whether or not the virtual machine is powered on.

The following categories of metrics are collected:

- **cpu**: Virtual CPU usage metrics. CPU metrics are an aggregate of all the virtual machine's cores and sockets.
- **disk**: Virtual hard disk capacity and performance metrics. When a virtual machine has multiple disks, metrics are reported as an aggregate for all disks.
- **mem**: Virtual machine memory usage.

For historic and current metrics, you can make a GET request to retrieve all categories of metrics from a virtual machine. A GET request for historic metrics returns only the most recent 24 hours of metric data samples. You can also make a POST request with a body that constrains the response to a subset of current or historic metrics in any or all categories. A POST request for historic metrics can specify a time period of up to 14 days.

**IMPORTANT** To support retrieval of historic metrics, vCloud Director must be configured with additional database software dedicated to historic metrics collection. See the vCloud Director Installation and Upgrade Guide.

### Metrics Links in a Vm Element

When a virtual machine is powered on, the Vm element that represents it includes four links where the value of the type attribute has the form application/vnd.vmware.vcloud.meters.*UsageSpec.xml.

```xml
<Vm ... >
    ...
    <Link
        rel="down"
        href="https://vcloud.example.com/api/vApp/vm-4/metrics/current"
        type="application/vnd.vmware.vcloud.meters.currentUsageSpec+xml"/>
</Vm ... >
```

VMware, Inc.  121
Use the links where rel="down" with a GET request to retrieve current or historic metrics in all categories.

Use the links where rel="metrics" with a POST request to retrieve a subset of current or historic metrics.

When a virtual machine is powered off, you cannot retrieve current metrics from it, so the .../metrics/current links are not returned in the Vm element.

**Negative Values for Current Metrics**

Current metrics can be returned with negative values if the specific metric being requested does not have a valid current value. This condition can exist if you make a metrics request before the first collection interval for that metric elapses. For example, a request for CPU usage metrics made shortly after a virtual machine is powered on can report negative values.

**Retrieving All Current or Historic Metrics**

You can make a GET request to either of the metrics links in a Vm object to retrieve all current or historic metrics.

Current metrics are instantaneous values collected at the time the request is processed by the server. A GET request for current metrics retrieves a list of current metric values, one for each category, and returns it as a single XML document. You can retrieve a subset of current metrics by making a POST request with a CurrentUsageSpec body.

Historic metrics are collected and stored for 14 days. A GET request for historic metrics retrieves the past 24 hours of metric history and returns it as a single XML document. In the typical case, this document contains thousands of lines. You can retrieve a subset of historic metrics by making a POST request with a HistoricUsageSpec body.

**Prerequisites**

Verify that you are logged in to the vCloud API as an administrator, Catalog Author, or vApp Author.

**Procedure**

1. Retrieve the XML representation of the virtual machine.

   Use a request like this one:

   ```
   GET https://vcloud.example.com/api/vApp/vm-4
   ```
2 Examine the Vm element to find the metrics links that it contains.

These links have the following form, where id is the virtual machine’s unique identifier:

```
<Link
   rel="down"
   type="application/vnd.vmware.vcloud.metrics.currentUsageSpec+xml"
   href="https://vcloud.example.com/api/vApp/vm-id/metrics/current"/>
```

```
<Link
   rel="down"
   type="application/vnd.vmware.vcloud.metrics.historicUsageSpec+xml"
   href="https://vcloud.example.com/api/vApp/vm-id/metrics/historic"/>
```

3 Make a GET request to the appropriate link.

<table>
<thead>
<tr>
<th>Option</th>
<th>Request</th>
</tr>
</thead>
<tbody>
<tr>
<td>Retrieve all current metrics</td>
<td>GET <a href="https://vcloud.example.com/api/vApp/vm-id/metrics/current">https://vcloud.example.com/api/vApp/vm-id/metrics/current</a></td>
</tr>
<tr>
<td>Retrieve all historic metrics for the past 24 hours</td>
<td>GET <a href="https://vcloud.example.com/api/vApp/vm-id/metrics/historic">https://vcloud.example.com/api/vApp/vm-id/metrics/historic</a></td>
</tr>
</tbody>
</table>

See “Example: Retrieving All Current or Historic Metrics,” on page 123

**Example: Retrieving All Current or Historic Metrics**

This request retrieves all current metrics from a virtual machine.

**Request**

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

**Response**

200 OK

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

```
<CurrentUsage
   xmlns="http://www.vmware.com/vcloud/v1.5"
   ...>
   <Link
      rel="up"
      href="https://vcloud.example.com/api/vApp/vm-4"
      type="application/vnd.vmware.vcloud.vm+xml"/>
   <Metric
      name="cpu.usage.average"
      units="PERCENT"
      value="33.7"/>
   <Metric
      name="cpu.usage.maximum"
      units="PERCENT"
      value="89.0"/>
   <Metric
      name="cpu.usagemhz.average"
      units="MEGAHERTZ"
      value="262.0"/>
   <Metric
      name="mem.usage.average"
      units="PERCENT"
      value="44.0"/>
```

"Example: Retrieving All Current or Historic Metrics,” on page 123
This request retrieves the past 24 hours of metrics from a virtual machine. The typical response contains thousands of lines, so only a subset appears here.

Request

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

Response

200 OK
Content-Type: application/vnd.vmware.vcloud.metrics.historicusage+xml

```xml
<?xml version="1.0" encoding="UTF-8"?>
<HistoricUsage xmlns="http://www.vmware.com/vcloud/v1.5">
  <Link rel="up" href="https://vcloud.example.com/api/vApp/vm-4" type="application/vnd.vmware.vcloud.vm+xml"/>
  <MetricSeries expectedInterval="1800"
    name="disk.provisioned.latest"
    unit="KILOBYTE">
    <Sample timestamp="2013-12-02T20:00:00.000Z"
      value="875295.0"/>
    <Sample timestamp="2013-12-02T20:30:00.000Z"
      value="741388.0"/>
  </MetricSeries>
  <MetricSeries expectedInterval="300"
    name="cpu.usagemhz.average"
    unit="MEGAHERTZ">
    <Sample timestamp="2013-12-02T20:03:20.000Z"
      value="505.0"/>
  </MetricSeries>
</HistoricUsage>
```

VMware, Inc.
<MetricSeries
    expectedInterval="300"
    name="disk.read.average"
    unit="KILOBYTES_PER_SECOND">
    <Sample
        timestamp="2013-12-02T20:03:20.000Z"
        value="208.7"/>
    <Sample
        timestamp="2013-12-02T20:06:40.000Z"
        value="0.0"/>
</MetricSeries>

<MetricSeries
    expectedInterval="300"
    name="mem.usage.average"
    unit="PERCENT">
    <Sample
        timestamp="2013-12-02T20:03:00.000Z"
        value="55.26272895119407"/>
    <Sample
        timestamp="2013-12-02T20:06:40.000Z"
        value="47.19066823323568"/>
</MetricSeries>

<MetricSeries
    expectedInterval="300"
    name="cpu.usage.average"
    unit="PERCENT">
    <Sample
        timestamp="2013-12-02T20:03:20.000Z"
        value="24.07100028991699"/>
    <Sample
        timestamp="2013-12-02T20:06:40.000Z"
        value="5.87066666285197"/>
</MetricSeries>

<MetricSeries
    expectedInterval="1800"
    name="disk.used.latest"
    unit="KILOBYTE">
    <Sample
        timestamp="2013-12-02T20:00:00.000Z"
        value="262154.0"/>
    <Sample
        timestamp="2013-12-02T20:30:00.000Z"
        value="373779.0"/>
</MetricSeries>

<MetricSeries
    expectedInterval="300"
    name="disk.write.average"
    unit="KILOBYTES_PER_SECOND"/>
Retrieving Metrics Subsets

To retrieve a subset of current or historic metrics, make a POST request to a current or historic metrics URL. The request body specifies the metrics you want.

The vCloud API defines two request bodies, CurrentUsageSpec and HistoricUsageSpec, that you can use to constrain a metrics request to a subset of available metrics. With these request bodies, you can use metrics by name, or supply a pattern that uses a wildcard to match a set of names. You can also specify a collection interval for historic metrics. See "Metric Names and Patterns," on page 130.

Prerequisites

Verify that you are logged in to the vCloud API as an administrator, Catalog Author, or vApp Author.

Procedure

1. Retrieve the XML representation of the virtual machine.
   Use a request like this one:
   
   GET https://vcloud.example.com/api/vApp/vm-4

2. Examine the Vm element to find the metrics links where rel="metrics".
   These links have the following form, where id is the virtual machine's unique identifier:
   
   <Link rel="metrics" type="application/vnd.vmware.vcloud.metrics.currentUsageSpec+xml" href="https://vcloud.example.com/api/vApp/vm-id/metrics/current"/>

   <Link rel="metrics" type="application/vnd.vmware.vcloud.metrics.historicUsageSpec+xml" href="https://vcloud.example.com/api/vApp/vm-id/metrics/historic"/>
3 Create the appropriate request body.

The request body specifies the metrics you want.

<table>
<thead>
<tr>
<th>Option</th>
<th>Request</th>
</tr>
</thead>
<tbody>
<tr>
<td>Retrieve a subset of current metrics</td>
<td>Create a CurrentUsageSpec element to use as the request body.</td>
</tr>
<tr>
<td>Retrieve a subset of historic metrics</td>
<td>Create a HistoricUsageSpec element to use as the request body.</td>
</tr>
</tbody>
</table>

4 POST the request body to the appropriate metrics link.

See “Example: Retrieving a Subset of Current or Historic Metrics,” on page 127

**NOTE** If you do not include a request body, the request is processed as though you had made a GET request.

**Example: Retrieving a Subset of Current or Historic Metrics**

This example uses a POST request with a CurrentUsageSpec body to request a subset of the current metrics retrieved in “Example: Retrieving All Current or Historic Metrics,” on page 123. This example also shows the use of wildcards in a MetricPattern element. See “Metric Patterns,” on page 131.

Request

```plaintext
POST https://vcloud.example.com/api/vApp/vm-4/metrics/current
Content-type: application/vnd.vmware.vcloud.metrics.currentUsageSpec+xml
...
<?xml version="1.0" encoding="UTF-8"?>
<CurrentUsageSpec xmlns="http://www.vmware.com/vcloud/v1.5">
  <MetricPattern>*.average</MetricPattern>
</CurrentUsageSpec>
```

Response

```plaintext
200 OK
Content-Type: application/vnd.vmware.vcloud.metrics.currentusage+xml
...
<?xml version="1.0" encoding="UTF-8"?>
<CurrentUsage
  xmlns="http://www.vmware.com/vcloud/v1.5">
  <Link
    rel="up"
    href="https://vcloud.example.com/api/vApp/vm-4f"
    type="application/vnd.vmware.vcloud.vm+xml"/>
  <Metric
    name="cpu.usage.average"
    unit="PERCENT"
    value="5.36"/>
  <Metric
    name="cpu.usagemhz.average"
    unit="MEGAHERTZ"
    value="112.0"/>
  <Metric
    name="mem.usage.average"
    unit="PERCENT"
    value="2.99"/>
  <Metric
    name="disk.write.average"
    unit="KILOBYTES_PER_SECOND"```
This example uses a POST request with a HistoricUsageSpec body to request a subset of the historic metrics retrieved in “Example: Retrieving All Current or Historic Metrics,” on page 123. This example also shows the use of wildcards and a time specification for the past 8 hours.

Request

POST https://vcloud.example.com/api/vApp/vm-4/metrics/historic
Content-type: application/vnd.vmware.vcloud.metrics.historicUsageSpec+xml

<?xml version="1.0" encoding="UTF-8"?>
<HistoricUsageSpec xmlns="http://www.vmware.com/vcloud/v1.5">
  <RelativeStartTime
    interval="8"
    unit="hour"/>
  <RelativeEndTime
    interval="0"
    unit="hour"/>
  <MetricPattern>cpu.*</MetricPattern>
  <MetricPattern>disk.*</MetricPattern>
</HistoricUsageSpec>

The full response contains several thousand lines, so only a subset appears here.

Response

200 OK
Content-Type: application/vnd.vmware.vcloud.metrics.historicusage+xml

<?xml version="1.0" encoding="UTF-8"?>
<HistoricUsage xmlns="http://www.vmware.com/vcloud/v1.5">
  <Link
    rel="up"
    href="https://vcloud.example.com/api/vApp/vm-4" type="application/vnd.vmware.vcloud.vm+xml"/>
  <MetricSeries
    expectedInterval="1800"
    name="disk.provisioned.latest"
    unit="KILOBYTE">
    <Sample
      timestamp="2013-12-02T20:00:00.000Z"
      value="875295.0"/>
    <Sample
      timestamp="2013-12-02T20:30:00.000Z"
      value="741388.0"/>
  </MetricSeries>
  <MetricSeries
    expectedInterval="300"
    name="cpu.usagemhz.average"
    unit="MEGAHERTZ">
    <Sample
      timestamp="2013-12-02T20:00:00.000Z"
      value="875295.0"/>
    <Sample
      timestamp="2013-12-02T20:30:00.000Z"
      value="741388.0"/>
  </MetricSeries>
</HistoricUsage>
Chapter 5 Deploying and Operating vApps
Metric Names and Patterns

When you retrieve a subset of current or historic metrics, you can specify the metrics by name, or by using a pattern that includes a wildcard character.

Metric Names

Metric names are dot-separated strings.

Table 5-2. Metric Names

<table>
<thead>
<tr>
<th>Metric Name</th>
<th>Type</th>
<th>Unit</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>cpu.usage.average</td>
<td>rate</td>
<td>percent</td>
<td>Host view of this virtual machine's average actively used CPU as a percentage of total available. Includes all cores in all sockets.</td>
</tr>
<tr>
<td>cpu.usagemhz.average</td>
<td>rate</td>
<td>megahertz</td>
<td>Host view of this virtual machine's average actively used CPU as a raw measurement. Includes all cores in all sockets.</td>
</tr>
<tr>
<td>cpu.usage.maximum</td>
<td>rate</td>
<td>percent</td>
<td>Host view of this virtual machine's maximum actively used CPU as a percentage of total available. Includes all cores in all sockets.</td>
</tr>
<tr>
<td>mem.usage.average</td>
<td>absolute</td>
<td>percent</td>
<td>Memory used by this virtual machine as a percentage of total configured memory.</td>
</tr>
</tbody>
</table>
Table 5.2. Metric Names (Continued)

<table>
<thead>
<tr>
<th>Metric Name</th>
<th>Type</th>
<th>Unit</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>disk.provisioned.latest</td>
<td>absolute</td>
<td>kilobytes</td>
<td>Storage space allocated to this virtual hard disk in the containing organization virtual data center.</td>
</tr>
<tr>
<td>disk.used.latest</td>
<td>absolute</td>
<td>kilobytes</td>
<td>Storage used by all virtual hard disks.</td>
</tr>
<tr>
<td>disk.read.average</td>
<td>rate</td>
<td>kilobytes per second</td>
<td>Average read rate for all virtual hard disks.</td>
</tr>
<tr>
<td>disk.write.average</td>
<td>rate</td>
<td>kilobytes per second</td>
<td>Average write rate for all virtual hard disks.</td>
</tr>
</tbody>
</table>

**Note**: When a virtual machine has multiple disks, metrics are reported as an aggregate for all disks. CPU metrics are an aggregate of all cores and sockets.

**Metric Patterns**

A CurrentUsageSpec or HistoricUsageSpec can include MetricPattern elements that specify multiple metric names using a wildcard character. To form a metric pattern, replace any component of a metric name with an asterisk, which is a wildcard that matches all values of the metric name component it replaces. For example, this MetricPattern matches all metric names that begin with disk.

```xml
<MetricPattern>disk.*</MetricPattern>
```

The response would include these metric names: disk.provisioned.latest, disk.used.latest, disk.read.average, and disk.write.average.

A different MetricPattern matches all metric names that begin with disk and end with average.

```xml
<MetricPattern>disk.*,average</MetricPattern>
```

The response would include the metric names disk.read.average and disk.write.average.

**Metric Series Expected Intervals and Timestamps**

A HistoricUsage element includes zero or more MetricSeries elements, each of which includes a set of Sample elements. Each MetricSeries has an expectedInterval attribute that specifies the interval, in milliseconds, at which the samples in the series are reported. Each Sample in the MetricSeries has a timestamp attribute noting the absolute time at which the sample was taken. You can use the timestamp and expectedInterval values to aggregate sample data, and to determine when metrics became unavailable because the virtual machine was powered off or unreachable.

**Specifying Collection Start and End Times**

A HistoricUsageSpec can include a time specification that constrains the result set to metrics collected between a start time and an end time. This time specification can be relative or absolute.

In RelativeStartTime and RelativeEndTime elements, start and end times are specified as an interval and a unit, which are interpreted as interval units ago. For example, this HistoricUsageSpec requests metrics collected during the past 8 hours.

```xml
<HistoricUsageSpec xmlns="http://www.vmware.com/vcloud/v1.5">
  <RelativeStartTime
    interval="8"
    unit="hour"/>
  <RelativeEndTime
```
interval="0"
unit="hour"/>
...
</HistoricUsageSpec>

You can also write this specification with no RelativeEndTime element, rather than a RelativeEndTime with an interval attribute value of 0. Both constructions specify an end time of now.

Alternatively, you can use AbsoluteStartTime and AbsoluteEndTime elements to specify absolute start and end times in a HistoricUsageSpec, as shown in this example, which returns metrics recorded during a one hour period:

```xml
<HistoricUsageSpec xmlns="http://www.vmware.com/vcloud/v1.5">
  <AbsoluteStartTime time="2013-11-13T10:00:00.000Z" />
  <AbsoluteEndTime time="2013-11-13T11:00:00.000Z" />
  ...
</HistoricUsageSpec>
```

**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.


**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:
   
   ```plaintext
   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 “Capture a vApp as a Template,” on page 114.

Updating a vApp Template

You can update the following sections of a vApp template.

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

- **Owner**
  You can use a PUT request to update the value of the template's Owner element. See “View or Change the Owner of an Object,” on page 85.

- **ProductSection elements**
  You can retrieve or update the template's ProductSection elements, which provide a way to pass runtime information to the virtual machines defined in the template. See “Retrieve or Modify ProductSection Elements,” on page 156.

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 you supply 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 InstantiationParams element of an instantiateVAppTemplate request.

- **StartupSection**
  Defines the order in which the virtual machines in the vApp start up and shut down. If this section is omitted, the startup and shutdown order of virtual machines in the vApp is indeterminate.
Configuring a Virtual Machine

An instantiateVAppTemplate request can modify certain properties of individual virtual machines in a vApp template by including a SourcedItem in the InstantiateVAppTemplateParams request body. You can include a SourcedItem for each virtual machine in the vApp. You can also reconfigure a virtual machine after the vApp has been deployed by updating any of the following sections of one of its Vm elements:

- **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 134 to update individual sections or groups of related sections.

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 System Administration Requests,” on page 261.
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 the Link where rel="edit" to update the section with your modifications.

Prerequisites

Verify that you are logged in to the vCloud API as an administrator, Catalog Author, or vApp Author.

Procedure

1. Retrieve the XML representation of the vApp.

   Use a GET request as shown in “Example: Configuration Links in a vApp,” on page 135.

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

   The response portion of “Example: Configuration Links in a vApp,” on page 135 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 137.

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>
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>
<thead>
<tr>
<th>Operation</th>
<th>Request</th>
<th>Request Body</th>
<th>Response</th>
</tr>
</thead>
<tbody>
<tr>
<td>Retrieve vApp LeaseSettingsSection</td>
<td>GET API-URL/vApp/vapp-id/leaseSettingsSection/</td>
<td>None</td>
<td>LeaseSettingsSection</td>
</tr>
<tr>
<td>Update vApp LeaseSettingsSection</td>
<td>PUT API-URL/vApp/vapp-id/leaseSettingsSection/</td>
<td>LeaseSettingsSection</td>
<td>Task</td>
</tr>
<tr>
<td>Retrieve vApp StartupSection</td>
<td>GET API-URL/vApp/vapp-id/startupSection/</td>
<td>None</td>
<td>StartupSection</td>
</tr>
<tr>
<td>Update vApp StartupSection</td>
<td>PUT API-URL/vApp/vapp-id/startupSection/</td>
<td>StartupSection</td>
<td>Task</td>
</tr>
<tr>
<td>Retrieve vApp NetworkConfigSection</td>
<td>GET API-URL/vApp/vapp-id/networkConfigSection/</td>
<td>None</td>
<td>NetworkConfigSection</td>
</tr>
<tr>
<td>Update vApp NetworkConfigSection</td>
<td>PUT API-URL/vApp/vapp-id/networkConfigSection/</td>
<td>NetworkConfigSection</td>
<td>Task</td>
</tr>
</tbody>
</table>

vCloud API Programming Guide

VMware, Inc.
Retrieve the Configuration Links for a Virtual Machine

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

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

Prerequisites

Verify that you are logged in to the vCloud API as an administrator, Catalog Author, or vApp Author.

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 135.
2. In the \texttt{VApp} element's \texttt{Children} element, find the \texttt{Vm} element that represents the virtual machine and retrieve it.
3. Examine the response to find the \texttt{reconfigureVm} link and \texttt{edit} links to modifiable sections.
   The response portion of “Example: Configuration Links in a Vm Element,” on page 137 shows the \texttt{reconfigureVm} link and links for each of the modifiable sections of the \texttt{Vm}. You cannot modify sections that do not contain a \texttt{Link} where \texttt{rel="edit"}.

Example: Configuration Links in a Vm Element

This example retrieves a \texttt{Vm} element shown in “Example: Configuration Links in a vApp,” on page 135. It expands that element to show its configuration links. It also shows the entire \texttt{NetworkConnectionSection} of that \texttt{Vm}, and additional information that is referenced in other examples. You cannot modify sections that do not have a \texttt{Link} where \texttt{rel="edit"}, so they do not appear in this example. Modifiable sections of the parent \texttt{vApp} are shown in “Example: Configuration Links in a vApp,” on page 135.

Request:

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

Response:

200 OK
Content-Type: application/vnd.vmware.vcloud.vm+xml

```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
```
<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>

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>
<thead>
<tr>
<th>Operation</th>
<th>Request</th>
<th>Request Body</th>
<th>Response</th>
</tr>
</thead>
<tbody>
<tr>
<td>Retrieve the NetworkConnectionSection of a virtual machine</td>
<td>GET API-URL/vApp/vm-id/networkConnectionSection</td>
<td>None</td>
<td>NetworkConnectionSection</td>
</tr>
<tr>
<td>Update the NetworkConnectionSection of a virtual machine</td>
<td>PUT API-URL/vApp/vm-id/networkConnectionSection</td>
<td>NetworkConnectionSection</td>
<td>Task</td>
</tr>
<tr>
<td>Retrieve the GuestCustomizationSection of a virtual machine</td>
<td>GET API-URL/vApp/vm-id/guestCustomizationSection</td>
<td>None</td>
<td>GuestCustomizationSection</td>
</tr>
<tr>
<td>Update the GuestCustomizationSection of a virtual machine</td>
<td>PUT API-URL/vApp/vm-id/guestCustomizationSection</td>
<td>GuestCustomizationSection</td>
<td>Task</td>
</tr>
<tr>
<td>Retrieve the OperatingSystemSection of a virtual machine</td>
<td>GET API-URL/vApp/vm-id/operatingSystemSection</td>
<td>None</td>
<td>OperatingSystemSection</td>
</tr>
<tr>
<td>Operation</td>
<td>Request</td>
<td>Request Body</td>
<td>Response</td>
</tr>
<tr>
<td>-----------</td>
<td>---------</td>
<td>--------------</td>
<td>----------</td>
</tr>
<tr>
<td>Update the OperatingSystemSection of a virtual machine</td>
<td>PUT API-URL/vApp/vm-id/operatingSystemSection</td>
<td>OperatingSystemSection</td>
<td>Task</td>
</tr>
<tr>
<td>Retrieve the VirtualHardwareSection of a virtual machine</td>
<td>GET API-URL/vApp/vm-id/virtualHardwareSection</td>
<td>None</td>
<td>VirtualHardwareSection</td>
</tr>
<tr>
<td>Update the VirtualHardwareSection of a virtual machine</td>
<td>PUT API-URL/vApp/vm-id/virtualHardwareSection</td>
<td>VirtualHardwareSection</td>
<td>Task</td>
</tr>
<tr>
<td>Retrieve the CPU configuration of a virtual machine</td>
<td>GET API-URL/vApp/vm-id/virtualHardwareSection/cpu</td>
<td>None</td>
<td>ovf:Item</td>
</tr>
<tr>
<td>Update the CPU configuration of a virtual machine</td>
<td>PUT API-URL/vApp/vm-id/virtualHardwareSection/cpu</td>
<td>ovf:Item</td>
<td>Task</td>
</tr>
<tr>
<td>Retrieve the memory item from the VirtualHardwareSection of a virtual machine</td>
<td>GET API-URL/vApp/vm-id/virtualHardwareSection/memory</td>
<td>None</td>
<td>ovf:Item</td>
</tr>
<tr>
<td>Update the memory item in the VirtualHardwareSection of a virtual machine</td>
<td>PUT API-URL/vApp/vm-id/virtualHardwareSection/memory</td>
<td>ovf:Item</td>
<td>Task</td>
</tr>
<tr>
<td>Retrieve virtual disk items from the VirtualHardwareSection of a virtual machine</td>
<td>GET API-URL/vApp/vm-id/virtualHardwareSection/disks</td>
<td>None</td>
<td>RasdItemsList</td>
</tr>
<tr>
<td>Update virtual disk items in the VirtualHardwareSection of a virtual machine</td>
<td>PUT API-URL/vApp/vm-id/virtualHardwareSection/disks</td>
<td>RasdItemsList</td>
<td>Task</td>
</tr>
<tr>
<td>Retrieve network card items from the VirtualHardwareSection of a virtual machine</td>
<td>GET API-URL/vApp/vm-id/virtualHardwareSection/networkCards</td>
<td>None</td>
<td>RasdItemsList</td>
</tr>
<tr>
<td>Update network card items in the VirtualHardwareSection of a virtual machine</td>
<td>PUT API-URL/vApp/vm-id/virtualHardwareSection/networkCards</td>
<td>RasdItemsList</td>
<td>Task</td>
</tr>
<tr>
<td>Update the name, Description, and any or all of VirtualHardwareSection, OperatingSystemSection, NetworkConnectionSection, GuestCustomizationSection of a virtual machine.</td>
<td>POST API-URL/vApp/vm-id/action/reconfigureVm</td>
<td>Vm</td>
<td>Task</td>
</tr>
</tbody>
</table>
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 aVm.

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

<table>
<thead>
<tr>
<th>Element Name</th>
<th>RASD Resource Type</th>
<th>Attribute Name</th>
<th>Attribute Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>rasd:Connection</td>
<td>10 (Network adapters)</td>
<td>ipAddressingMode</td>
<td>xs:string</td>
<td>IP addressing mode to use for this connection. One of NONE, MANUAL, DHCP, POOL.</td>
</tr>
<tr>
<td>rasd:Connection</td>
<td>10 (Network adapters)</td>
<td>ipAddress</td>
<td>xs:string</td>
<td>If ipAddressingMode=&quot;MANUAL&quot;, set the IP address here</td>
</tr>
<tr>
<td>rasd:Connection</td>
<td>10 (Network adapters)</td>
<td>primaryNetworkConnection</td>
<td>xs:boolean</td>
<td>True if this is the primary network connection of the virtual machine</td>
</tr>
<tr>
<td>rasd:HostResource</td>
<td>17 (Hard disks)</td>
<td>capacity</td>
<td>xs:string</td>
<td>Hard disk capacity in megabytes. See “Retrieve or Modify the Hard Disk Configuration of a Virtual Machine,” on page 160</td>
</tr>
<tr>
<td>rasd:HostResource</td>
<td>17 (Hard disks)</td>
<td>busType</td>
<td>xs:string</td>
<td>Bus type. One of:</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>■ 5 (IDE)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>■ 6 (SCSI)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>■ 20 (SATA)</td>
</tr>
<tr>
<td>rasd:HostResource</td>
<td>17 (Hard disks)</td>
<td>busSubType</td>
<td>xs:string</td>
<td>Hard disk controller type. One of:</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>■ buslogic</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>■ lsilogic</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>■ lsilogicsas</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>■ VirtualSCSI</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>■ vmware.sata.ahci</td>
</tr>
<tr>
<td></td>
<td></td>
<td>storageProfileOverrid eVmDefault</td>
<td>xs:boolean</td>
<td>See “Retrieve or Modify the Hard Disk Configuration of a Virtual Machine,” on page 160 for valid combinations of busType and busSubType.</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>If true, the storage profile specified in storageProfileHref is always used for this disk regardless of the value specified in the virtual machine’s StorageProfile.</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>If false, any storage profile is specified in storageProfileHref is ignored and the disk is migrated to the storage specified in the virtual machine’s StorageProfile.</td>
</tr>
</tbody>
</table>
### Table 5-5. vCloud API Custom Attributes for OVF and RASD Elements (Continued)

<table>
<thead>
<tr>
<th>Element Name</th>
<th>RASD Resource Type</th>
<th>Attribute Name</th>
<th>Attribute Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>rasd:HostResource</td>
<td>17 (Hard disks)</td>
<td>storageProfileHref</td>
<td>xs:string</td>
<td>If storageProfileOverrideVmDefault is true, the value of storageProfileHref specifies the storage profile to use for this hard disk. The storage profile must be available in the VDC where this virtual machine is deployed.</td>
</tr>
<tr>
<td>ovf:OperatingSystemSection</td>
<td>N/A</td>
<td>osType</td>
<td>xs:string</td>
<td>Internal VMware identifier for the guest operating system. See Enum - VirtualMachineGuestOsIdentifier in the VMware vSphere API Reference documentation.</td>
</tr>
</tbody>
</table>
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>
</NetworkConnectionsSection>

<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)"
... />

...
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 an administrator, Catalog Author, or vApp Author.

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 135.

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">

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 an administrator, Catalog Author, or vApp Author.

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 144. 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 137.

For more information about these and other network services in vApp networks, see “Network Services in vApp Networks,” on page 148.

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>
    </IpScope>
   </IpScopes>
   <ParentNetwork
    type="application/vnd.vmware.vcloud.network+xml"
    name="Internet"
<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>manual</MappingMode>
        <VAppScopedVmId>3963994b-5a0a-48fe-b9ae-7f9a2d8e5b</VAppScopedVmId>
        <VmNicId>0</VmNicId>
      </OneToOneVmRule>
    </NatRule>
  </NatService>
</Features>
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 137, specifies an IP address that will not be available after the vApp network is reconfigured as shown here. “Example: Update a NetworkConnectionSection,” on page 152 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 152. 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 152 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 193. For more information about network services for organization VDC networks, see “Configure Edge Gateway Services,” on page 201

DHCP Service

A DhcpService element defines an IP address range and lease policies for a DHCP service that can be used by 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.

```xml
<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 146

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 137. 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.

```xml
<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>
  </DestinationVm>
</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 NAT strategy for a single Vm.

```xml
<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 146.

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.

```xml
<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 206, 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>
  ...
</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>
  ...
</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 an administrator, Catalog Author, or vApp Author.

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 section with your modifications.

   a In the retrieved section, find the Link element where rel="edit".
   b 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.

Example: Update a NetworkConnectionSection

This example modifies the NetworkConnectionSection shown in “Example: Configuration Links in a Vm
Element,” on page 137 so that this network connection is compatible with the reconfigured vApp network to
which it must connect. See “Example: Update a NetworkConfigSection,” on page 146. 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>
   </NetworkConnection>
   </NetworkConnectionSection>
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 an administrator, Catalog Author, or vApp Author.

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.
   a. In the retrieved section, find the Link element where rel="edit".
   b. 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

The initial configuration for the virtual machine used in this example shows a single CPU and
CoresPerSocket value of 1.
This request modifies the CPU section to add a second CPU to the VM by changing the \texttt{rasd:VirtualQuantity} value of the Item to 2. It also raises the value of \texttt{CoresPerSocket} to 2.

Request:

\texttt{PUT https://vcloud.example.com/api/vApp/vm-4/virtualHardwareSection/cpu}

\texttt{Content-type: application/vnd.vmware.vcloud.rasdItem+xml}

\begin{verbatim}
<?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>
    <vmw:CoresPerSocket ovf:required="false">2</vmw:CoresPerSocket>
</Item>
\end{verbatim}

Response:

202 Accepted

\texttt{Content-Type: application/vnd.vmware.vcloud.task+xml}

\begin{verbatim}
<Task ... operation="Updating Virtual Application Linux FTP server (7)" ...>...
</Task>
\end{verbatim}

\textbf{Retrieve or Modify the GuestCustomizationSection of a Virtual Machine}

The \texttt{GuestCustomizationSection} element includes a customization script and other parameters that are applied when you customize a virtual machine.

The \texttt{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 \texttt{OrgGuestPersonalizationSettings} of the organization that owns the virtual machine. See “Retrieve or Update Organization Settings,” on page 177.

The vCloud API also supports use of the \texttt{ovf:ProductSection} to pass an arbitrary set of \texttt{key=value} pairs to a vApp or virtual machine through the \texttt{ovf:Environment} element. See “Retrieve or Modify ProductSection Elements,” on page 156.
Prerequisites
Verify that you are logged in to the vCloud API as an administrator, Catalog Author, or vApp Author.

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.
   a In the retrieved section, find the Link element where rel="edit".
   b 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 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>
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 vApp can use VMware Tools to read these values from its ovf:Environment. A vApp 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 and modify the ProductSectionList to update the individual ProductSection elements it contains.

Prerequisites

Verify that you are logged in to the vCloud API as an administrator, Catalog Author, or vApp Author.

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 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:

```plaintext
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 137, 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 an administrator, Catalog Author, or vApp Author.

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 159.

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 137 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:
200 OK
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

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 an administrator, Catalog Author, or vApp Author.
**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 159.

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.

   You cannot modify the values of the `busType` and `busSubType` attributes after you create a new disk. When creating a new disk, be sure to set the values of `busType` and `busSubType` to a valid combination.

   **Table 5-6. Valid Combinations of busType and busSubType**

<table>
<thead>
<tr>
<th>busType</th>
<th>busSubType</th>
<th>Controller</th>
</tr>
</thead>
<tbody>
<tr>
<td>5</td>
<td>null</td>
<td>IDE controller</td>
</tr>
<tr>
<td>6</td>
<td>buslogic</td>
<td>BusLogic Parallel SCSI controller</td>
</tr>
<tr>
<td>6</td>
<td>lsiLogic</td>
<td>LSI Logic Parallel SCSI controller</td>
</tr>
<tr>
<td>6</td>
<td>lsiLogicSas</td>
<td>LSI Logic SAS SCSI controller</td>
</tr>
<tr>
<td>6</td>
<td>VirtualSCSI</td>
<td>Paravirtual SCSI controller</td>
</tr>
<tr>
<td>20</td>
<td>vmware.sata.ahci</td>
<td>SATA controller (hardware version 10 and later)</td>
</tr>
</tbody>
</table>

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

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

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

**Example: Modify the Hard Disk Configuration of a Virtual Machine**

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

Request:

```
PUT https://vcloud.example.com/api/vApp/vm-4/virtualHardwareSection/disks
Content-Type: application/vnd.vmware.vcloud.rasditemslist+xml

...<xml version="1.0" encoding="UTF-8"?>
```
<RasdItemsList
    xmlns="http://www.vmware.com/vcloud/v1.5"
    xmlns:rasd="http://schemas.dmtf.org/wbem/wscim/1/cim-
    schema/2/CIM_ResourceAllocationSettingData"
    type="application/vnd.vmware.vcloud.rasdItemsList+xml">

    <Item>
        <rasd:Address>0</rasd:Address>
        <rasd:Description>SCSI Controller</rasd:Description>
        <rasd:ElementName>SCSI Controller 0</rasd:ElementName>
        <rasd:InstanceID>2</rasd:InstanceID>
        <rasd:ResourceSubType>lsilogic</rasd:ResourceSubType>
        <rasd:ResourceType>6</rasd:ResourceType>
    </Item>

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

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

    <Item>
        <rasd:Address>0</rasd:Address>
        <rasd:Description>IDE Controller</rasd:Description>
        <rasd:ElementName>IDE Controller 0</rasd:ElementName>
        <rasd:InstanceID>3</rasd:InstanceID>
        <rasd:ResourceType>5</rasd:ResourceType>
    </Item>

</RasdItemsList>
The response is a task.

Response:

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

...<Task ... operation="Updating Virtual Application Linux FTP server (7)" ...>
...
</Task>

Update the Storage Profile for a Virtual Machine

You can update aVm 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.

EveryVm element includes a StorageProfile element whose href attribute value specifies the default storage profile consumed by the virtual machine. This default is used for all hard disk Items in the VirtualHardwareSection that do not specify storageProfileOverrideVmDefault.

If you do not specify a StorageProfile during an instantiate, compose, or recompose operation, it is inherited from the organization VDC in which the virtual machine is deployed. To change the value of an existing StorageProfile, you must update the entireVm element that contains it.

IMPORTANT When the system administrator changes the datastore that stores a virtual machine, you must update theVm element as shown in “Example: Update the Storage Profile for a Virtual Machine,” on page 164.

Prerequisites

Verify that you are logged in to the vCloud API as an administrator, Catalog Author, or vApp Author.

Procedure

1 Retrieve theVm element.

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

2 Modify the retrievedVm 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 theVm with your modifications.

a Find the Link element in theVm where rel="edit".

b Make a PUT request to the URL in that link’s href attribute value, and supply the modifiedVm 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’s StorageProfile has been updated and the virtual machine has been relocated to the new storage profile.
**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
<?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

```xml
<Task ... operation="Updating Virtual Application Linux FTP server (7)" ...>
  ...
</Task>
```

**Override the Default Storage Profile for a Hard Disk**

By default, all hard disks defined in the VirtualHardwareSection of a Vm element use the storage profile specified for the Vm. You can override this default for any of these disks when you instantiate a vApp template, compose or recompose a vApp, or reconfigure a virtual machine.

Every Vm element includes a StorageProfile element. The storage profile referenced in this element normally provides storage for all the hard disk Items in the virtual machine's VirtualHardwareSection. You can override this default by updating the virtual machine's VirtualHardwareSection to add storageProfileOverrideVmDefault and storageProfileHref attributes to the Item that defines the hard disk. You can update a VirtualHardwareSection when you are instantiating a vApp template, composing or recomposing a vApp, or reconfiguring a virtual machine.

**IMPORTANT** You cannot override the default storage profile for any hard disk of a virtual machine that is deployed in a VDC where fast provisioning is enabled.

**Prerequisites**

Verify that you are logged in to the vCloud API as an administrator, Catalog Author, or vApp Author.

**Procedure**

1. Retrieve the hard disk configuration from the virtual machine.

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

   ```bash
   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 159.

2 In the VirtualHardwareSection of the retrieved Vm, modify the rasd:HostResource element of the Item that defines the disk for which you want to override the default storage profile.

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.

a In the VirtualHardwareSection of the Vm, find the Item that represents the hard disk for which you want to override the default storage profile.

b Add a storageProfileHref to the rasd:HostResource element of the Item and set its value to the href of the storage profile you want to use for this disk. The storage profile you specify must be available in the VDC where this virtual machine is deployed.

c Add a storageProfileOverrideVmDefault attribute to the rasd:HostResource element of the Item. The value of this attribute controls whether changes to the virtual machine's StorageProfile affect the storage profile that this disk uses.

Table 5-7. How storageProfileOverrideVmDefault Values Affect Hard Disk Storage Profile Assignment

<table>
<thead>
<tr>
<th>Value</th>
<th>Result</th>
</tr>
</thead>
<tbody>
<tr>
<td>true (default)</td>
<td>The storage profile specified in storageProfileHref is always used for this disk regardless of the value specified in the virtual machine's StorageProfile</td>
</tr>
<tr>
<td>false</td>
<td>Any storage profile that is specified in storageProfileHref is ignored and the disk is migrated to the storage specified in the virtual machine's StorageProfile.</td>
</tr>
</tbody>
</table>

If you omit the storageProfileOverrideVmDefault attribute, the storageProfileHref is ignored.

3 Update the section with your modifications.

a In the retrieved section, find the Link element where rel="edit".

b 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.

Example: Override the Default Storage Profile for a Hard Disk

This example builds on the ones shown in “Example: Update the Storage Profile for a Virtual Machine,” on page 164 and “Example: Modify the Hard Disk Configuration of a Virtual Machine,” on page 161. This virtual machine has the storage profile specified in “Example: Update the Storage Profile for a Virtual Machine,” on page 164:

```xml
<StorageProfile
type="application/vnd.vmware.vcloud.vdcStorageProfile+xml"
name="Gold"
href="https://vcloud.example.com/api/vdcStorageProfile/3" />
```

To specify a new storage profile for the disk whose capacity was increased in “Example: Modify the Hard Disk Configuration of a Virtual Machine,” on page 161, you must provide the storageProfileHref for the new storage profile and also set the storageProfileOverrideVmDefault attribute to true.
Request:

PUT https://vcloud.example.com/api/vApp/vm-4/virtualHardwareSection/disks
Content-Type: application/vnd.vmware.vcloud.rasdItemsList+xml

<?xml version="1.0" encoding="UTF-8"?>
<RasdItemsList
    xmlns="http://www.vmware.com/vcloud/v1.5"
    xmlns:rasd="http://schemas.dmtf.org/wbem/wscim/1/cim-schema/2/CIM_ResourceAllocationSettingData"
    type="application/vnd.vmware.vcloud.rasdItemsList+xml">
    <Item>
        <rasd:Address>0</rasd:Address>
        <rasd:Description>SCSI Controller</rasd:Description>
        <rasd:ElementName>SCSI Controller 0</rasd:ElementName>
        <rasd:InstanceID>2</rasd:InstanceID>
        <rasd:ResourceSubType>lsilogic</rasd:ResourceSubType>
        <rasd:ResourceType>6</rasd:ResourceType>
    </Item>
    <Item>
        <rasd:AddressOnParent>0</rasd:AddressOnParent>
        <rasd:Description>Hard disk</rasd:Description>
        <rasd:ElementName>Hard disk 1</rasd:ElementName>
        <rasd:HostResource
            xmlns:vcloud="http://www.vmware.com/vcloud/v1.5"
            vcloud:capacity="10240"
            vcloud:busSubType="lsilogic"
            vcloud:busType="6"
            vcloud:storageProfileOverrideVmDefault="true"
            vcloud:storageProfileHref="https://vcloud.example.com/api/vdcStorageProfile/5">
        </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: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>
    </Item>
</RasdItemsList>
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 53.

This chapter includes the following topics:

- “Summary of Administrative Requests,” on page 169
- “Administrator Credentials and Privileges,” on page 172
- “Organization Administration,” on page 173
- “VDC Administration,” on page 185
- “Network Administration,” on page 193
- “Catalog Administration,” on page 221
- “User and Group Administration,” on page 243
- “Working With Roles and Rights,” on page 251

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.
<table>
<thead>
<tr>
<th>Operation</th>
<th>Request</th>
<th>Request Body</th>
<th>Response</th>
</tr>
</thead>
<tbody>
<tr>
<td>Retrieve an administrative view of an organization.</td>
<td>GET API-URL/admin/org/id</td>
<td>None</td>
<td>AdminOrg</td>
</tr>
<tr>
<td>Create an organization.</td>
<td>POST API-URL/admin/orgs</td>
<td>AdminOrg</td>
<td>AdminOrg</td>
</tr>
<tr>
<td>Enable an organization.</td>
<td>POST API-URL/admin/org/id/action/enable</td>
<td>None</td>
<td>204 No Content</td>
</tr>
<tr>
<td>Disable an organization.</td>
<td>POST API-URL/admin/org/id/action/disable</td>
<td>None</td>
<td>204 No Content</td>
</tr>
<tr>
<td>Update an organization.</td>
<td>PUT API-URL/admin/org/id</td>
<td>AdminOrg</td>
<td>AdminOrg</td>
</tr>
<tr>
<td>Create an Edge Gateway.</td>
<td>POST API-URL/admin/vdc/id/edgeGateways</td>
<td>EdgeGateway</td>
<td>EdgeGateway</td>
</tr>
<tr>
<td>Retrieve an Edge Gateway.</td>
<td>GET API-URL/admin/edgeGateway/id</td>
<td>None</td>
<td>EdgeGateway</td>
</tr>
<tr>
<td>Update Edge Gateway services.</td>
<td>POST API-URL/admin/edgeGateway/id/action/configureServices</td>
<td>EdgeGatewayServiceConfiguration</td>
<td>Task</td>
</tr>
<tr>
<td>Reapply Edge Gateway Services</td>
<td>POST API-URL/admin/edgeGateway/id/action/reapplyServices</td>
<td>None</td>
<td>204 No Content</td>
</tr>
<tr>
<td>Redeploy an Edge Gateway</td>
<td>POST API-URL/admin/edgeGateway/id/action/redeploy</td>
<td>None</td>
<td>204 No Content</td>
</tr>
<tr>
<td>Delete an Edge Gateway.</td>
<td>DELETE API-URL/admin/edgeGateway/id</td>
<td>None</td>
<td>Task</td>
</tr>
<tr>
<td>Create an organization VDC network.</td>
<td>POST API-URL/admin/vdc/id/networks</td>
<td>OrgVdcNetwork</td>
<td>Task</td>
</tr>
<tr>
<td>Update an organization VDC network.</td>
<td>PUT API-URL/admin/network/id</td>
<td>OrgVdcNetwork</td>
<td>Task</td>
</tr>
<tr>
<td>Retrieve a list of networks in this organization VDC.</td>
<td>GET API-URL/admin/vdc/id/networks</td>
<td>None</td>
<td>QueryResultRecords</td>
</tr>
<tr>
<td>Delete an organization VDC network.</td>
<td>DELETE API-URL/admin/network/id</td>
<td>None</td>
<td>Task</td>
</tr>
<tr>
<td>Synchronize syslog server settings for a vApp network.</td>
<td>POST API-URL/admin/network/id/action/syncSyslogServerSettings</td>
<td>None</td>
<td>Task</td>
</tr>
<tr>
<td>Synchronize syslog server settings for an Edge Gateway.</td>
<td>POST API-URL/admin/edgeGateway/id/action/syncSyslogServerSettings</td>
<td>None</td>
<td>Task</td>
</tr>
<tr>
<td>Reset a network.</td>
<td>POST API-URL/admin/network/id/action/reset</td>
<td>None</td>
<td>Task</td>
</tr>
</tbody>
</table>
### Table 6-1. Summary of Administrative Requests (Continued)

<table>
<thead>
<tr>
<th>Operation</th>
<th>Request</th>
<th>Request Body</th>
<th>Response</th>
</tr>
</thead>
<tbody>
<tr>
<td>Retrieve the admin view of a provider VDC.</td>
<td>GET API-URL/admin/vdc/id</td>
<td>None</td>
<td>ProviderVdc</td>
</tr>
<tr>
<td>List the organization VDCs supported by a provider VDC.</td>
<td>GET API-URL/admin/vdc/id/vdcReferences</td>
<td>None</td>
<td>VdcReferences</td>
</tr>
<tr>
<td>Create an organization VDC.</td>
<td>POST API-URL/admin/vdc/id/vdcReferences</td>
<td>AdminVdc</td>
<td>AdminVdc</td>
</tr>
<tr>
<td>Enable an organization VDC.</td>
<td>POST API-URL/admin/vdc/id/action/enable</td>
<td>None</td>
<td>204 No Content</td>
</tr>
<tr>
<td>Disable an organization VDC.</td>
<td>POST API-URL/admin/vdc/id/action/disable</td>
<td>None</td>
<td>204 No Content</td>
</tr>
<tr>
<td>Delete an organization VDC.</td>
<td>DELETE API-URL/admin/vdc/id</td>
<td>None</td>
<td>Task</td>
</tr>
<tr>
<td>Create a catalog.</td>
<td>POST API-URL/admin/collections</td>
<td>AdminCatalog</td>
<td>AdminCatalog</td>
</tr>
<tr>
<td>Share a catalog with all organizations in your cloud.</td>
<td>POST API-URL/admin/catalog/id/action/publish</td>
<td>PublishCatalogParams</td>
<td>204 No Content</td>
</tr>
<tr>
<td>Update catalog properties.</td>
<td>PUT API-URL/admin/catalog/id</td>
<td>AdminCatalog</td>
<td>AdminCatalog</td>
</tr>
<tr>
<td>Delete a catalog.</td>
<td>DELETE API-URL/admin/catalog/id</td>
<td>None</td>
<td>204 No Content</td>
</tr>
<tr>
<td>Update the owner of a catalog.</td>
<td>PUT API-URL/admin/catalog/id/owner</td>
<td>Owner</td>
<td>204 No Content</td>
</tr>
<tr>
<td>Create or import a user.</td>
<td>POST API-URL/admin/users</td>
<td>User</td>
<td>User</td>
</tr>
<tr>
<td>Update user properties.</td>
<td>PUT API-URL/admin/user/id</td>
<td>User</td>
<td>User</td>
</tr>
<tr>
<td>Retrieve a list of rights granted to a user.</td>
<td>GET API-URL/admin/user/id/grantedRights</td>
<td>None</td>
<td>Permissions</td>
</tr>
<tr>
<td>Delete a user.</td>
<td>DELETE API-URL/admin/user/id</td>
<td>None</td>
<td>204 No Content</td>
</tr>
<tr>
<td>Transfer ownership of a user's vApps, media, and catalogs to the caller.</td>
<td>POST API-URL/admin/user/id/action/takeOwnership</td>
<td>None</td>
<td>204 No Content</td>
</tr>
<tr>
<td>Import a group.</td>
<td>POST API-URL/admin/groups</td>
<td>Group</td>
<td>Group</td>
</tr>
<tr>
<td>Delete a group.</td>
<td>DELETE API-URL/admin/group/id</td>
<td>None</td>
<td>204 No Content</td>
</tr>
<tr>
<td>Update group properties.</td>
<td>PUT API-URL/admin/group/id</td>
<td>Group</td>
<td>Group</td>
</tr>
<tr>
<td>Create a role.</td>
<td>POST API-URL/admin/role/id</td>
<td>Role</td>
<td>Role</td>
</tr>
<tr>
<td>Update role properties.</td>
<td>PUT API-URL/admin/role/id</td>
<td>Role</td>
<td>Role</td>
</tr>
</tbody>
</table>
### 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 49.

### 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:

   ```xml
   <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 173.

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 173.

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 173, 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"
<Description>Example Corporation's Finance Organization</Description>
<FullName>Finance</FullName>
<Settings>
<OrgGeneralSettings>
<CanPublishCatalogs>false</CanPublishCatalogs>
<CanPublishExternally>true</CanPublishExternally>
<CanSubscribe>false</CanSubscribe>
<DeployedVMQuota>0</DeployedVMQuota>
<StoredVmQuota>0</StoredVmQuota>
<UseServerBootSequence>false</UseServerBootSequence>
<DelayAfterPowerOnSeconds>0</DelayAfterPowerOnSeconds>
</OrgGeneralSettings>
<OrgLdapSettings>
<OrgLdapMode>SYSTEM</OrgLdapMode>
</OrgLdapSettings>
<OrgEmailSettings>
<IsDefaultSmtpServer>true</IsDefaultSmtpServer>
<IsDefaultOrgEmail>true</IsDefaultOrgEmail>
</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 180.

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

<?xml version="1.0" encoding="UTF-8"?>
<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>
Example Corporation's Finance Organization

<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" />
</Settings>
<OrgGeneralSettings type="application/vnd.vmware.admin.organizationGeneralSettings+xml" href="https://vcloud.example.com/api/admin/org/26/settings/general">
  <CanPublishCatalogs>false</CanPublishCatalogs>
  <CanPublishExternally>true</CanPublishExternally>
  <CanSubscribe>false</CanSubscribe>
  <DeployedVMQuota>0</DeployedVMQuota>
  <StoredVmQuota>0</StoredVmQuota>
  <UseServerBootSequence>false</UseServerBootSequence>
  <DelayAfterPowerOnSeconds>0</DelayAfterPowerOnSeconds>
</OrgGeneralSettings>

</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 storage and deployment quotas and other behaviors for virtual machines that members of this organization own. Sets the scope of catalog publication and subscription in this organization.
- **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 LdapSettings or a custom LDAP service defined here.
- **OrgEmailSettings**: Defines whether this organization uses the email service defined in the system EmailSettings 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 PasswordPolicySettings.
- **OrgOperationLimitsSettings**: Specifies limits to be placed on simultaneous resource-intensive operations and console sessions for members of this organization.
- **OrgGuestPersonalizationSettings**: Default values for GuestCustomizationSection elements in virtual machines created by this organization. See “Retrieve or Modify the GuestCustomizationSection of a Virtual Machine,” on page 154
- **OrgFederationSettings**: Settings related to the identity provider that this organization shares with other applications or enterprises to enable single sign-on. See “Enabling and Managing Federation,” on page 182.

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 178.

**Example: Update Organization General Settings**

This example updates the general settings of the organization created in “Example: Create an Organization,” on page 173. When you create or retrieve an AdminOrg, these settings are contained in the OrgGeneralSettings element. To update them, you must use a GeneralOrgSettings element, which has the same contents as OrgGeneralSettings. This 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
<?xml version="1.0" encoding="UTF-8"?>
<GeneralOrgSettings
   xmlns="http://www.vmware.com/vcloud/v1.5"
   type="application/vnd.vmware.admin.organizationGeneralSettings+xml">
   <CanPublishCatalogs>false</CanPublishCatalogs>
   <CanPublishExternally>true</CanPublishExternally>
   <CanSubscribe>false</CanSubscribe>
   <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
...
```

```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"
```
When you retrieve the organization after updating its GeneralOrgSettings, you can see the results of the update in the OrgGeneralSettings element of the AdminOrg.

GET https://vcloud.example.com/api/admin/org/26
...
  <AdminOrg ...>
    ...
    <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>
      <CanPublishExternally>true</CanPublishExternally>
      <CanSubscribe>false</CanSubscribe>
      <DeployedVMQuota>10</DeployedVMQuota>
      <StoredVmQuota>100</StoredVmQuota>
      <UseServerBootSequence>false</UseServerBootSequence>
      <DelayAfterPowerOnSeconds>0</DelayAfterPowerOnSeconds>
    </OrgGeneralSettings>
    ...
  </AdminOrg>

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 52.

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, remove all the objects it contains, then disable and remove it.

a Delete all of the following from the organization:
   - local users and groups
   - catalogs
   - VDCs

b POST a request to the action/disable link to disable the organization.
   After the organization is disabled, its representation includes a rel="remove" link.

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 173.

Request:

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

Response:

204 No Content

Truststore and Keytab Maintenance for Organizations

You can use the vCloud API to upload and manage SSL certificates, keystores, and Kerberos keytabs for your organization's LDAP service. You can also use the vCloud API to configure SSPI, the Microsoft Security Support Provider Interface, for use with Active Directory.

The OrgLdapSettings element includes links that enable an organization administrator to maintain certificates and truststores for the organization's LDAP service.

<AdminOrg ... >
   ...
   <OrgLdapSettings ... >
      ...
      <Link
         rel="certificate:update"
         type="application/vnd.vmware.admin.certificateUpdateParams+xml"
      <Link
         rel="certificate:reset"
      <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"/>
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>
<thead>
<tr>
<th>Upload Type</th>
<th>Maximum Size in Megabytes</th>
</tr>
</thead>
<tbody>
<tr>
<td>LDAP certificate</td>
<td>2</td>
</tr>
<tr>
<td>LDAP keystore</td>
<td>2</td>
</tr>
<tr>
<td>LDAP SSPI keytab</td>
<td>2</td>
</tr>
</tbody>
</table>

**Prerequisites**

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

**Procedure**

1. Determine whether the request requires a body.
   - Requests whose rel value includes the string reset do not require a body. For details about other request bodies, see the schema reference.
2. POST the request to the request URL.
   - Include the request body if one is required.
3. Take any action required by the response.
   - The response to an update request includes an uploadLocation parameter whose value is a URL to which you can upload the certificate, keytab, or keystore with a PUT request. Requests whose rel value includes the string reset return a No Content response and require no further action.

**Example: Upload an SSL Certificate for an Organization LDAP Service**

This example uploads an SSL certificate whose size is 892 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="892"
  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:

```xml
<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: 892
    ...serialized contents of certificate...
EOF
```

Response:

```
200 OK
```

**Enabling and Managing Federation**

An organization that defines an external identity provider that it shares with other applications or enterprises is said to be federated with those applications or enterprises.

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 OrgFederationSettings element that contains SAML metadata retrieved from the federation’s identity provider. By default, this element is empty.

**NOTE** When an organization is created, it is provided with a self-signed certificate for use when establishing trust with an identity provider. This certificate expires after one year. You can regenerate this certificate by making a request of the following form.

```
POST https://vcloud.example.com/api/admin/org/id/settings/federation/action/regenerateCertificate
```

You can retrieve this certificate with a request of the following form, where `name` is the name of the organization:

```
GET https://vcloud.example.com/cloud/org/name/saml/metadata/alias/vcd
```

You do not have to be authenticated to make this request.

**Prerequisites**

Verify that you are logged in to the vCloud API as an organization administrator or system administrator.
Download the appropriate SAML metadata in XML format from your identity provider. The SAML metadata must provide mappings for the user attributes shown in this XML fragment:

```xml
<saml:Attribute
    FriendlyName="Groups"
    Name="http://rsa.com/schemas/attr-names/2009/01/GroupIdentity"
    NameFormat="urn:oasis:names:tc:SAML:2.0:attrname-format:uri"/>

<saml:Attribute
    FriendlyName="givenName"
    Name="http://schemas.xmlsoap.org/ws/2005/05/identity/claims/givenname"
    NameFormat="urn:oasis:names:tc:SAML:2.0:attrname-format:uri"/>

<saml:Attribute
    FriendlyName="surname"
    Name="http://schemas.xmlsoap.org/ws/2005/05/identity/claims/surname"
    NameFormat="urn:oasis:names:tc:SAML:2.0:attrname-format:uri"/>

<saml:Attribute
    FriendlyName="Subject Type"
    Name="http://vmware.com/schemas/attr-names/2011/07/isSolution"
    NameFormat="urn:oasis:names:tc:SAML:2.0:attrname-format:uri"/>

<saml:Attribute
    FriendlyName="userPrincipalName"
    Name="http://schemas.xmlsoap.org/claims/UPN"
    NameFormat="urn:oasis:names:tc:SAML:2.0:attrname-format:uri"/>

<saml:Attribute
    FriendlyName="email"
    Name="http://schemas.xmlsoap.org/ws/2005/05/identity/claims/emailaddress"
    NameFormat="urn:oasis:names:tc:SAML:2.0:attrname-format:uri"/>
```

**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 link to the OrgFederationSettings.

3. Retrieve the OrgFederationSettings.
   
   Make a GET request to the URL in the element's href attribute value.

4. Modify the retrieved OrgFederationSettings element to add your identity provider's SAML metadata as the value of the SAMLMetadata element.
   
   XML entities must be encoded, as shown in “Example: Update Organization Federation Settings,” on page 183

5. Update the OrgFederationSettings 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 Federation Settings,” on page 183.

**Example: Update Organization Federation Settings**

This example updates the OrgFederationSettings of the organization created in “Example: Create an Organization,” on page 173. The update adds SAML metadata retrieved from an identity provider, and enables federation by setting Enabled to true. Only a subset of the SAML metadata is shown.
Request:

PUT https://vcloud.example.com/api/admin/org/26/settings/federation
Content-Type: application/vnd.vmware.admin.organizationFederationSettings+xml
...
<?xml version="1.0" encoding="UTF-8"?>
<OrgFederationSettings type="application/vnd.vmware.admin.organizationFederationSettings+xml">
  <SAMLMetadata>
    <EntitiesDescriptor xmlns="urn:oasis:names:tc:SAML:2.0:metadata"
                         xmlns:saml="urn:oasis:names:tc:SAML:2.0:assertion" ...
    &gt; ...
    &lt;/EntitiesDescriptor&gt; </SAMLMetadata>
  <Enabled>true</Enabled>
</OrgFederationSettings>

NOTE To update or remove OrgFederationSettings after a SAML identity provider is specified, you must include an empty SAMLMetadata element in the update request. If you do not, the request fails and the OrgFederationSettings are not changed.

The response contains information extracted from the request, and includes Link elements that the server creates.

Response:

200 OK
Content-Type: application/vnd.vmware.admin.organizationFederationSettings+xml
...
<OrgFederationSettings type="application/vnd.vmware.admin.organizationFederationSettings+xml">
  <Link rel="up" href="https://vcloud.example.com/api/admin/org/26/settings"
        type="application/vnd.vmware.admin.organization+xml"/>
  <Link rel="edit" href="https://vcloud.example.com/api/admin/org/26/settings/federation"
        type="application/vnd.vmware.admin.organizationFederationSettings+xml"/>
  <Link rel="federation:regenerateFederationCertificate"
  <SAMLMetadata>&lt;EntitiesDescriptor xmlns="urn:oasis:names:tc:SAML:2.0:metadata"
                                         xmlns:saml="urn:oasis:names:tc:SAML:2.0:assertion" ...
                               &gt; ...
                               &lt;/EntitiesDescriptor&gt; </SAMLMetadata>
  <Enabled>true</Enabled>
</OrgFederationSettings>
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 277.
- 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 272 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/vdcsparams" 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:
      
      | AllocationModel | Description |
      |-----------------|-------------|
      | 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. |
      
      **NOTE** If you choose AllocationPool or ReservationPool, you can also include an OverCommitAllowed element in the CreateVdcParams request. Setting its value to false prevents creation of the VDC if the ComputeCapacity you specified is greater than what the backing ProviderVDC can supply.
      
   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 187.
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 187.

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 187.

**Example: Create an Organization VDC**

This example adds an AllocationVApp VDC to the organization created in “Example: Create an Organization,” on page 173. The new organization VDC is provisioned from the Provider VDC created in “Create a Provider VDC,” on page 277, 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 272 for information on how to find a NetworkPoolReference to use.

Request:

```
POST https://vcloud.example.com/api/admin/org/26/vdcsparams
Content-Type: application/vnd.vmware.admin.createVdcParams+xml
```

```
<?xml version="1.0" encoding="UTF-8"?>
<CreateVdcParams
  xmlns="http://www.vmware.com/vcloud/v1.5"
  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"/>
```
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 57.
- There is an empty AvailableNetworks element. To add networks to this organization VDC, see “Create an Organization VDC Network,” on page 211.
- Additional Link elements are included for operations that are now valid, but that were not valid while the VDC was under construction.

```xml
<AdminVdc
 xmlns="http://www.vmware.com/vcloud/v1.5"
 xmlns:vcloud="http://www.vmware.com/vcloud/v1.5"
 xmlns:vimext="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">
 <VimObjectRef>
   <VimServerRef
    type="application/vnd.vmware.admin.vmwvirtualcenter+xml"
    name="vc1"
    href="https://vcloud.example.com/api/admin/extension/vimServer/9" />
   <MoRef>resgroup-949</MoRef>
   <VimObjectType>RESOURCE_POOL</VimObjectType>
 </VimObjectRef>
</VCloudExtension>
</AdminVdc>
```

Chapter 6 Creating and Managing Organizations
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.

**NOTE** Storage profiles are represented as Storage Policies in the vCloud Director Web console.

**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 187. 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 187.

The response might look something like this:

```xml
<?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. This example creates a storage profile that is not a default storage profile, and has a specific value for `Limit`, 5038 MB. To specify unlimited storage (subject to the capacity of the underlying Provider VDC), set the value of `Limit` to 0.

Request:

```
POST https://vcloud.example.com/api/admin/vdc/44/vdcStorageProfiles
Content-Type: application/vnd.vmware.admin.updateVdcStorageProfiles+xml
```

```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>
```

The response is a Task.

Response:

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

```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 an organization 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, remove all the objects it contains, then disable and remove it.
  
  a  Relocate or remove any vApps that have been uploaded to the VDC.
  b  Remove any organization VDC networks that the VDC contains.
  c  Remove any Edge Gateways that the VDC contains.
  d  POST a request to its action/disable link to disable the VDC.
    
    After the VDC is disabled, its representation includes a rel="remove" link.
  e  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 145.

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 148
- 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 eternal 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 6-3. Types of Organization VDC Networks and Their Requirements**

<table>
<thead>
<tr>
<th>Organization VDC Network Connection</th>
<th>Description</th>
<th>Requirements</th>
</tr>
</thead>
<tbody>
<tr>
<td>Direct connection to an external network.</td>
<td>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.</td>
<td>The cloud must contain an external network.</td>
</tr>
<tr>
<td>Routed connection to an external network.</td>
<td>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.</td>
<td>The VDC must contain an Edge Gateway and a network pool.</td>
</tr>
<tr>
<td>No connection to an external network.</td>
<td>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.</td>
<td>The VDC must contain a network pool.</td>
</tr>
</tbody>
</table>

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 203.

- **GatewayDhcpService** Provides DHCP services to virtual machines on the network. A variant of this service, `DhcpService`, is intended to provide DHCP services in vApp networks. See “Gateway DHCP Service Configurations,” on page 211.

- **GatewayIpsecVpnService** Defines one or more virtual private networks that connect an Edge Gateway to another network in or outside of the cloud.


- **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 206.

For an example of adding services to an Edge Gateway, see “Configure Edge Gateway Services,” on page 201. 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 287 and “Create a Network Pool,” on page 290. 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 272.

**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 Provider VDC 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: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 198 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 198 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 198.

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 185. 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:

```xml
<Configuration>
  <IpScopes>
    <IpScope>
      ...
      <Gateway>10.147.115.190</Gateway>
      <Netmask>255.255.255.0</Netmask>
      ...
    </IpScope>
    <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>
    ...
  </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
<?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 201.

Response:

```xml
<?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">
```
<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>
<FirewallService>
<IsEnabled>true</IsEnabled>
<DefaultAction>drop</DefaultAction>
</FirewallService>
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 196. 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 196, 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:

   ```xml
   <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
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 196. For details about this FirewallService, see “Firewall Service Configurations,” on page 203.

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>
```
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:

<table>
<thead>
<tr>
<th>Policy</th>
<th>The default policy value, allow, causes the firewall to forward packets that match the rules. Specify drop to drop packets that match the rules.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Protocols</td>
<td>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.</td>
</tr>
<tr>
<td>SourcePortRange</td>
<td>Specify a source IP port or port range, or set to any to match any port.</td>
</tr>
<tr>
<td>DestinationPortRange</td>
<td>Specify a destination IP port or port range, or set to any to match any port.</td>
</tr>
<tr>
<td>SourceIp</td>
<td>Specify a source IP address, or use one of these strings.</td>
</tr>
</tbody>
</table>

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>
Table 6-4. SourceIp and DestinationIp Values

<table>
<thead>
<tr>
<th>Value</th>
<th>Result</th>
</tr>
</thead>
<tbody>
<tr>
<td>Any</td>
<td>Matches any IP address</td>
</tr>
<tr>
<td>Internal</td>
<td>Matches any IP address originating on an organization VDC network</td>
</tr>
<tr>
<td></td>
<td>connected to this EdgeGateway. When used in a vApp network, matches</td>
</tr>
<tr>
<td></td>
<td>any IP address assigned to a virtual machine in the vApp.</td>
</tr>
<tr>
<td>External</td>
<td>Matches any IP address originating on an external network connected</td>
</tr>
<tr>
<td></td>
<td>to this EdgeGateway. When used in a vApp network, matches any IP</td>
</tr>
<tr>
<td></td>
<td>address except those assigned to a virtual machine in the vApp.</td>
</tr>
</tbody>
</table>

**DestinationIp**
Specify a source IP address, or use one of the strings shown in Table 6-4.

**EnableLogging**
Set to true to log all packets that trigger this rule. See “Firewall Rule Logging,” on page 203.

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.

```xml
<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>
  </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 202.

**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 198. 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 an SNAT rule, you do not need to specify values for `TranslatedPort` and `OriginalPort`, which default to `any`.

```xml
<?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>192.168.0.10</TranslatedIp>
      <TranslatedPort>any</TranslatedPort>
      <Protocol>any</Protocol>
      <IcmpSubType>any</IcmpSubType>
    </GatewayNatRule>
  </NatRule>
</NatService>
```

**Note** The system assigns an `Id` value to each rule you create and uses these values when logging rule actions.
<NatRule>
    <RuleType>SNAT</RuleType>
    <IsEnabled>true</IsEnabled>
    <GatewayNatRule>
        <Interface href="https://vcloud.example.com/api/admin/network/297" />
        <OriginalIp>192.168.0.10-192.168.0.255</OriginalIp>
        <TranslatedIp>10.147.115.155</TranslatedIp>
        <Protocol>any</Protocol>
    </GatewayNatRule>
</NatRule>

To add this service to an Edge Gateway, include it in an EdgeGatewayServiceConfiguration. See "Example: Configure Services on an Edge Gateway," on page 202.

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 ExternalIp element of the RouterInfo from the network to which this static route connects.

**Interface** Specify internal if NextHopIp contains an IP address in the same network. Specify external if NextHopIp contains an IP address in a different network.

**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>
</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.

```xml
<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 202.
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 148.

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 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.

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>IP_HASH</td>
<td>Selects a server based on a hash of the source and destination IP address of each packet.</td>
</tr>
<tr>
<td>LEAST_CONN</td>
<td>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.</td>
</tr>
<tr>
<td>ROUND_ROBIN</td>
<td>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.</td>
</tr>
<tr>
<td>URI</td>
<td>The request URL is hashed and divided by the total weight of the running servers. (If the request URL includes a query, it is discarded, and only the fragment of the URL to the left of the ? is considered.) 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.</td>
</tr>
</tbody>
</table>

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. 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.

```xml
<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>
  </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 202.

**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**

```xml
<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>
  </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 202.
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 202.

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 212, “Create an Organization VDC Network With a Routed Connection,” on page 215, and “Create an Isolated Organization VDC Network,” on page 218.
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 193.

**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:
   
   ```xml
   <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 196.
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 214.

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 214.
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 214.

**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 185. 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" />
```
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 196.

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 216.
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 216

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 216.

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 185. The network connects through the Edge Gateway created in “Create an Edge Gateway,” on page 196. 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="RoutedOVDCNet"
   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>
   </Configuration>
</OrgVdcNetwork>
<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"
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.

Prerequisites

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

Procedure

1. Create an OrgNetwork element.

   Specify a value of isolated in the FenceMode element of the network Configuration. See the request portion of “Example: Create an Isolated Organization VDC Network,” on page 218.

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 218.

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 218.

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 185. 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>
<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"/>

Response:

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

...
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/2008/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 196.

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

Response:

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

...<Task ...
  status="running"
...
  operation="Synchronized syslog settings for EdgeGateway theEdge(2000)"
...
</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 use it as a destination for uploads or a source of subscription-based content.

Catalogs contain references to vApp templates and media images. You can configure a catalog in several different ways:

- as a repository for local content that can remain private to the catalog owner or can be shared with other users, groups, or organizations in your cloud
- as a source of published content, to which other clouds can subscribe.
- as a local repository for content published by another cloud or any Web site that hosts a VMware Content Subscription Protocol (VCSP) endpoint.

An organization administrator or catalog owner controls catalog sharing. Organization administrators in organizations that have permission to publish catalogs control publication and subscription options for catalogs in their organization. A system administrator can enable background synchronization of catalogs with external sources and set background synchronization schedules to regulate consumption of network bandwidth by this activity.

**Access to Catalogs**

A catalog initially grants full control to its owner and no access to other users. The catalog owner or a user with organization administrator or catalog author rights can grant catalog access to other members of the organization, individually or collectively. See “Controlling Access to vApps and Catalogs,” on page 86. Organization administrators and system administrators can share a catalog with other organizations in the cloud.

**Synchronization**

The VMware Content Subscription Protocol (VCSP) is an open standard that can be implemented by any system that can provide HTTP or HTTPS access. Because vCloud Director implements this protocol, catalogs can subscribe to content that originates in another instance of vCloud Director or at any remote site that supports a VCSP endpoint. When content at a remote site changes, you must synchronize the catalog...
items that hold local copies of that content. Synchronization keeps a catalog up to date with its external subscription. A catalog owner can synchronize individual catalog items or entire catalogs at any time. A system administrator can also schedule background synchronization for catalogs, so that all externally subscribed catalogs in the system are synchronized on a common schedule.

**Version Numbers**

As part of VCSP support, catalogs and catalog items have version numbers, which are integer values that increment monotonically. A catalog with an external subscription contains the most recent version of each of its catalog items if it has been synchronized with its external source.

The catalog item version number increases whenever any of the following changes occur.

- A file in the referenced entity is added, removed, or changed.
- The name or description of the catalog item is changed.

The catalog version number increases whenever any of the following changes occur.

- A catalog item is added to or removed from the catalog.
- The version number of any contained catalog item changes.
- The name or description of the catalog is changed.

**Add a Catalog to an Organization**

Every organization has an add URL for catalogs. An administrator or catalog author can create a catalog by POSTing an AdminCatalog element to this URL.

A new Catalog object is an empty container for catalog items, which are references to vApp templates and media images. There are several options for creating a catalog:

- You can create a catalog to use in your own organization or cloud, to hold content that you create locally by uploading or importing.
- You can enable external publication of a catalog that you create locally. See “Publish an Existing Catalog Externally,” on page 235.
- You can create a catalog that is published externally. See “Create a Catalog For External Publication,” on page 225.
- You can create a catalog that has an external subscription. Contents of these catalogs are downloaded from an external source.

**Prerequisites**

- Verify that you are logged in as a system administrator, an organization administrator, or a user with the Catalog Author role.
- Verify that at least one VDC exists in your organization. You cannot create a catalog in an organization that has no VDCs.

**Procedure**

1. Retrieve the XML representation of the organization to which 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 223.

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

See the request portion of “Example: Create a Catalog,” on page 223.

The server creates an empty catalog and returns its representation in the response. See the response portion of “Example: Create a Catalog,” on page 223.

**Example: Create a Catalog**

This example adds a catalog to the organization created in “Example: Create an Organization,” on page 173. Because the request does not specify a CatalogStorageProfile, the catalog is created on the default storage profile for the first VDC created in the organization. To create the catalog on a specific storage profile, you can add a CatalogStorageProfiles element to the request. See “Specify a Storage Profile for a Catalog,” on page 229.

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 implement operations on 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 shared with other organizations. See “Share a Catalog with All Organizations in a Cloud,” on page 232.
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="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" />
  <Tasks>
    <Task
      status="running"
      operation="Creating Catalog Example Catalog (32)"
    />
  </Tasks>
  <Description>New Catalog for Example Org</Description>
  <CatalogItems/>
  <IsPublished>false</IsPublished>
  <DateCreated>2013-06-18T09:52:59.260-07:00</DateCreated>
  <VersionNumber>1</VersionNumber>
  <CatalogStorageProfiles/>
</AdminCatalog>
Create a Catalog For External Publication

An organization administrator in an organization that has permission to publish externally can create a catalog whose contents, catalog items created locally, are made available to external consumers on a subscription basis.

Organizations that are permitted to publish externally can enable any of their catalogs for external publication. A catalog that is enabled for external publication becomes accessible at a URL assigned by the system, and can be protected by a password. A catalog in another instance of vCloud Director can subscribe to an externally published catalog and maintain an up-to-date set of catalog items. See “Synchronization,” on page 221.

You can use this procedure to create a new catalog that is enabled for external publication, or you can use the procedure shown in “Publish an Existing Catalog Externally,” on page 235 to enable an existing catalog for external publication. If a catalog has an external subscription, you cannot enable it for external publication.

Prerequisites

- Verify that you are logged in as a system administrator, an organization administrator, or a user with the Catalog Author role.
- Verify that at least one VDC exists in your organization. You cannot create a catalog in an organization that has no VDCs.
- Verify that your organization has permission to publish externally.

The OrgGeneralSettings in the AdminOrg element that represents your organization must have a CanPublishExternally element with a value of true.

```
<CanPublishExternally>true</CanPublishExternally>
```

Retrieve the XML representation of the organization to which to add the catalog, and examine the response to locate the Link element that contains the URL for adding catalogs to the organization. See “Add a Catalog to an Organization,” on page 222.

Procedure

1. Create an AdminCatalog element that includes PublishExternalCatalogParams.

   See the request portion of “Example: Create a Catalog For External Publication,” on page 225.

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

   See the request portion of “Example: Create a Catalog For External Publication,” on page 225.

The server creates an empty catalog and returns its representation in the response. See the response portion of “Example: Create a Catalog,” on page 223.

Example: Create a Catalog For External Publication

This request is similar to the one used in “Example: Create a Catalog,” on page 223 but includes a PublishExternalCatalogParams element that specifies that the catalog is to be published externally. This request creates the catalog with a content cache, which can improve synchronization performance for external catalogs that subscribe to this one, but requires enough space on the system’s transfer storage to accommodate all of the items in the catalog. This request also sets the PreserveIdentityInfoFlag element to false, which prevents values such as the BIOS UUID and MAC address associated with virtual machine identities from being published. If you omit this element or set its value to true, these values are published for all catalog items that represent virtual machines.
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="PublishedExternally">
 <Description>Example Catalog for External Publication</Description>
 <PublishExternalCatalogParams>
  <IsPublishedExternally>true</IsPublishedExternally>
  <IsCacheEnabled>true</IsCacheEnabled>
  <PreserveIdentityInfoFlag>false</PreserveIdentityInfoFlag>
  <Password>Pa55w0rd</Password>
 </PublishExternalCatalogParams>
</AdminCatalog>

The response is similar to the one shown in “Example: Create a Catalog,” on page 223, but includes the PublishExternalCatalogParams element. The embedded Task element tracks the creation of the catalog and its accompanying Web site. The Password element in the request is never returned.

Response:

201 Created
Content-Type: application/vnd.vmware.admin.catalog+xml

...  
<AdminCatalog
 xmlns:vcloud="http://www.vmware.com/vcloud/v1.5"
 name="PublishedExternally"
 id="urn:vcloud:catalog:33"
 type="application/vnd.vmware.admin.catalog+xml"
 href="https://vcloud.example.com/api/admin/catalog/33" >
 ...
 <Link
  rel="up"
  type="application/vnd.vmware.admin.organization+xml"
  href="https://vcloud.example.com/api/admin/org/26" />
 <Link ... />
 ...
 <Tasks>
  <Task
   status="running"
   ...
   operation="Creating Catalog PublishedExternally (33)"
   ...>
   ...
  </Task>
 <Tasks>
 <Description>Example Catalog for External Publication</Description>
 <CatalogItems/>
 <IsPublished>false</IsPublished>
 <DateCreated>2013-06-18T09:59:13.110-07:00</DateCreated>
 <VersionNumber>1</VersionNumber>
 <CatalogStorageProfiles/>
 <PublishExternalCatalogParams>
  <IsPublishedExternally>true</IsPublishedExternally>
  ...
When the task completes, the `PublishExternalCatalogParams` includes a URL at which external consumers can connect to the catalog.

```
</AdminCatalog>
```

Create a Catalog With an External Subscription

An organization administrator can create a catalog whose contents are downloaded from an external source. You cannot use catalogs created this way to hold catalog items that are created locally.

Organizations that are permitted to subscribe to external sources can create catalogs whose contents are downloaded from a catalog hosted on another instance of vCloud Director, or any Web site that implements the VMware Content Subscription Protocol (VCSP). See “Synchronization,” on page 221.

If a catalog has an external subscription, you cannot enable it for external publication.

Prerequisites

- Verify that you are logged in as a system administrator, an organization administrator, or a user with the Catalog Author role.
- Verify that at least one VDC exists in your organization. You cannot create a catalog in an organization that has no VDCs.
- Verify that your organization has permission to create catalogs with external subscriptions.
  - The `CanSubscribe` element with a value of `true` in the `OrgGeneralSettings` in the `AdminOrg` element that represents your organization.
- The external source must implement the VMware Content Subscription Protocol. See “Synchronization,” on page 221.
- You must know the URL of the external source, and the password if one is required.

Retrieve the XML representation of the organization to which you want to add the catalog, and examine the response to locate the `Link` element that contains the URL for adding catalogs to the organization. See “Add a Catalog to an Organization,” on page 222.

Procedure

1. Create an `AdminCatalog` element that includes `ExternalCatalogSubscriptionParams`.
   - See the request portion of “Example: Create a Catalog With an External Subscription,” on page 228.
2. POST the `AdminCatalog` element to the organization's add URL for catalogs.
   - See the request portion of “Example: Create a Catalog With an External Subscription,” on page 228.

The server creates an empty catalog and returns its representation in the response. See the response portion of “Example: Create a Catalog,” on page 223.
Example: Create a Catalog With an External Subscription

This request creates a catalog with a subscription to the VCSP URL https://vcloud2.example.com/vcsp/catalog/5. Because the LocalCopy element in ExternalCatalogSubscriptionParams has a value of false, files that comprise a vApp template or media image that a catalog item references are not downloaded until a user requests them. If you create a catalog where LocalCopy has a value of true, these files are downloaded the first time that the catalog subscription is synchronized, and on each subsequent synchronization where any of the catalog items has a newer version number. The default value of LocalCopy is false.

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="SubscribedExternally">
 <Description>Example Catalog With External Subscription</Description>
 <ExternalCatalogSubscriptionParams>
  <SubscribeToExternalFeeds>true</SubscribeToExternalFeeds>
  <Location>https://vcloud2.example.com/vcsp/catalog/5</Location>
  <Password>Pa55w0rd</Password>
  <LocalCopy>false</LocalCopy>
 </ExternalCatalogSubscriptionParams>
</AdminCatalog>

The response is similar to the one shown in “Example: Create a Catalog,” on page 223, but includes the ExternalCatalogSubscriptionParams element that you supplied in the request. The Password element in the request is never returned.

Response:

201 Created
Content-Type: application/vnd.vmware.admin.catalog+xml
...
<AdminCatalog
 xmlns:vcloud="http://www.vmware.com/vcloud/v1.5"
 name="SubscribedExternally"
 id="urn:vcloud:catalog:34"
 type="application/vnd.vmware.admin.catalog+xml"
 href="https://vcloud.example.com/api/admin/catalog/34">
 ...
 <Link
  rel="up"
  type="application/vnd.vmware.admin.organization+xml"
  href="https://vcloud.example.com/api/admin/org/26" />
 <Link ...
 <Link
  rel="sync"
  href="https://vcloud.example.com/api/catalog/34/action/sync"/>
 <Link ...
 ...
 <Tasks>
  <Task
   status="running"
   ...
operation="Creating Catalog SubscribedExternally (34)"
...
<Task>
<Description>Example Catalog With External Subscription</Description>
</Task>
<AdminCatalog>
<IsPublished>false</IsPublished>
<DateCreated>2013-06-18T10:00:03.012-07:00</DateCreated>
<VersionNumber>1</VersionNumber>
<ExternalCatalogSubscriptionParams>
<SubscribeToExternalFeeds>true</SubscribeToExternalFeeds>
<Location>https://vcloud2.example.com/vcsp/catalog/5</Location>
</ExternalCatalogSubscriptionParams>
</AdminCatalog>

What to do next
Synchronize the catalog with its external source. Any catalog with an external subscription includes a link of the form:

```xml
<Link rel="sync" href="https://vcloud.example.com/api/catalog/id/action/sync"/>
```

As a catalog owner or administrator, you can POST a request to the rel="sync" URL to synchronize the catalog with its external source. See “Synchronize a Catalog or Catalog Item,” on page 81.

Specify a Storage Profile for a Catalog
Any request that creates a catalog can also specify a storage profile for the items in that catalog.

If you do not specify a CatalogStorageProfile when creating a catalog, the catalog is created on the default storage profile for the first VDC created in the organization. To create the catalog on a specific storage profile, retrieve the list of available storage profiles in your organization and specify one in the CatalogStorageProfiles element of the request.

Prerequisites
- Verify that you are logged in as a system administrator, an organization administrator, or a user with the Catalog Author role.
- Verify that at least one VDC exists in your organization. You cannot create a catalog in an organization that has no VDCs.

Procedure
1. Find the list of storage profiles in your organization.
   Use a query like this one.
   ```xml
   https://vcloud.example.com/api/query?type=orgVdcStorageProfile&format=references
   ```
   The response is a list of references to all the storage profiles used by VDCs in your organization.

   ```xml
   <OrgVdcStorageProfileReferences ... >
   <OrgVdcStorageProfileReference
       type="application/vnd.vmware.vcloud.vdcStorageProfile+xml"
       name="a"
       id="urn:vcloud:vdcstorageProfile:59"
       href="https://vcloud.example.com/api/vdcStorageProfile/59" />
   ```
NOTE If you are a system administrator, you must filter the response to show only those storage profiles in the organization where you are creating the catalog.

2 Create an AdminCatalog element that includes a CatalogStorageProfiles element.

You can include a single CatalogStorageProfile in the request. Only the type and href attributes are required. See the request portion of “Example: Specify a Storage Profile for a Catalog,” on page 230.

3 POST the AdminCatalog element to the organization’s add URL for catalogs.

See the request portion of “Example: Specify a Storage Profile for a Catalog,” on page 230.

The server creates an empty catalog on the specified storage profile and returns its representation in the response. See the response portion of “Example: Create a Catalog,” on page 223.

Example: Specify a Storage Profile for a Catalog

This example modifies the request shown in “Example: Create a Catalog,” on page 223 to add a CatalogStorageProfiles element.

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>
 <CatalogStorageProfiles>
  <VdcStorageProfile
type="application/vnd.vmware.admin.vdcStorageProfile+xml"
 href="https://vcloud.example.com/api/vdcStorageProfile/59" />
 </CatalogStorageProfiles>
</AdminCatalog>

The response is similar to the one shown in “Example: Create a Catalog,” on page 223, and includes a CatalogStorageProfiles element derived form the one that you specified in the request.

Response:

201 Created
Content-Type: application/vnd.vmware.admin.catalog+xml
...
<AdminCatalog
 xmlns:vcloud="http://www.vmware.com/vcloud/v1.5"
 name="Example Catalog"
...

<Description>New Catalog for Example Org</Description>
<CatalogItems/>
...
Update Catalog Access Controls

If you are an administrator or catalog owner, you can use the controlAccess links in your organization to grant or restrict access to a catalog.

A catalog initially grants full access to its owner and no access to other users. An administrator or the catalog owner can use the vCloud API access control mechanism to view or modify catalog access controls. For a general discussion of access controls in vCloud Director, see “Controlling Access to vApps and Catalogs,” on page 86.

Procedure

1. Retrieve the XML representation of the organization that contains the catalog.
   Use a request like this one:
   ```
   GET https://vcloud.example.com/api/org/26
   ```

2. Examine the Org element to find the controlAccess links for the catalogs that it contains.
   These links have the following form:
   ```
   <Link rel="down" type="application/vnd.vmware.vcloud.controlAccess+xml" href="https://vcloud.example.com/api/org/id/catalog/id/controlAccess/">
   <Link rel="controlAccess" type="application/vnd.vmware.vcloud.controlAccess+xml" href="https://vcloud.example.com/api/org/id/catalog/id/action/controlAccess/">
   ```

   **Note**: The controlAccess links for catalogs are not returned in the AdminOrg response when you retrieve the organization in the admin view.

3. Create a ControlAccessParams element request body that specifies the details of the update.

4. POST the ControlAccessParams element to the action/controlAccess link for the catalog.

**Example: Update Catalog Access Controls**

This request updates the access controls of a catalog to grant full control to one user and read-only access to another user. The request body, a ControlAccessParams element, specifies a value of false for the IsSharedToEveryone element, and contains an AccessSetting element for each user whose access rights are being modified. Each user is identified by a reference to a User object. See “User and Group Administration,” on page 243. The response, a subset of which appears in this example, echoes the request.

**Request:**

```
POST https://vcloud.example.com/api/org/9/catalog/32/action/controlAccess
Content-Type: application/vnd.vmware.vcloud.controlAccess+xml
... 
```
Share a Catalog with All Organizations in a Cloud

An organization administrator can share a catalog to make it visible to the administrators in all other organizations in a cloud.

The owner of a catalog can make it available to other users in the containing organization. See “Update Catalog Access Controls,” on page 231. If you are an organization administrator, you can also share catalogs with all other organizations in your cloud if your organization's CanPublishCatalogs element has a value of true. The value of this element is controlled by the system administrator. To share 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 share.

   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 this example shows:

```xml
<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 shared and becomes available to members of all other organizations in the cloud.

**Example: Share a Catalog With All Organizations in a Cloud**

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

**Share a Catalog With Specific Organizations in a Cloud**

A system administrator can share a catalog to make it visible to the administrators of specific organizations in a cloud.

The owner of a catalog can share it to make it available to all organization in the cloud. When more selective sharing is required, a system administrator can share a catalog with specific organizations if that catalog is not already shared to all organizations.

**Prerequisites**

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

**Procedure**

1 Retrieve the XML representation of the organization that contains the catalog.

   Use a request like this one:

   GET https://vcloud.example.com/api/org/26

2 Examine the Org element to find the controlAccess links for the catalogs that it contains.

   These links have the following form:

   ```xml
   <Link
       rel="down"
       type="application/vnd.vmware.vcloud.controlAccess+xml"
       href="https://vcloud.example.com/api/org/id/catalog/id/controlAccess/"/>
   ```
NOTE The controlAccess links for catalogs are not returned in the AdminOrg response when you retrieve the organization in the admin view.

3 Create a ControlAccessParams element that contains an AccessSettings element that specifies the organizations with which to share the catalog.

4 POST the ControlAccessParams element to the action/controlAccess link for the catalog.

The catalog is shared with administrators of the specified organizations.

**Example: Share a Catalog With Specific Organizations in a Cloud**

This request updates the access controls of a catalog to share it to a single organization, giving read-only access to all users in that organization.

Request:

POST https://vcloud.example.com/api/org/9/catalog/32/action/controlAccess/"
Content-Type: application/vnd.vmware.vcloud.controlAccess+xml
...
<?xml version="1.0" encoding="UTF-8"?>
<ControlAccessParams
xmlns="http://www.vmware.com/vcloud/v1.5">
<IsSharedToEveryone>false</IsSharedToEveryone>
<AccessSettings>
  <AccessSetting>
    <Subject
type="application/vnd.vmware.admin.org+xml"
href="https://vcloud.example.com/api/admin/org/355"/>
    <AccessLevel>ReadOnly</AccessLevel>
  </AccessSetting>
</AccessSettings>
</ControlAccessParams>

Response:

200 OK
Content-Type: application/vnd.vmware.vcloud.controlAccess+xml
...
<ControlAccessParams
xmlns="http://www.vmware.com/vcloud/v1.5">
<IsSharedToEveryone>false</IsSharedToEveryone>
<AccessSettings>
  <AccessSetting>
    <Subject
type="application/vnd.vmware.admin.org+xml"
href="https://vcloud.example.com/api/admin/org/355"/>
    <AccessLevel>ReadOnly</AccessLevel>
  </AccessSetting>
</AccessSettings>
</ControlAccessParams>
Publish an Existing Catalog Externally

An organization administrator in an organization that has permission to publish externally can update an existing catalog to make its contents available to external consumers on a subscription basis.

Organizations that are permitted to publish externally can enable any of their local catalogs for external publication. A catalog that is enabled for external publication becomes accessible at a URL assigned by the system, and can be protected by a password. A catalog in another instance of vCloud Director can subscribe to an externally published catalog and maintain an up-to-date set of catalog items. See “Synchronization,” on page 221.

Prerequisites

- Verify that you are logged in as a system administrator, an organization administrator, or a user with the Catalog Author role.
- Verify that your organization has permission to publish externally.
  
  Verify that the OrgGeneralSettings in the AdminOrg element that represents your organization has a CanPublishExternally element with a value of true.
  
  `<CanPublishExternally>true</CanPublishExternally>`
- Verify that the catalog you want to enable for external publication does not have an external subscription.

Procedure

1. In the admin view, retrieve the XML representation of the catalog.
   
   Use a request like this one:
   
   `GET https://vcloud.example.com/api/admin/catalog/32`

2. Examine the AdminCatalog element to find the publishToExternalOrganizations link it contains.
   
   This link has the following form:
   
   `<Link rel="publishToExternalOrganizations"`  
   
   `type="application/vnd.vmware.admin.publishExternalCatalogParams+xml"`  
   
   `href="https://vcloud.example.com/api/admin/catalog/32/action/publishToExternalOrganizations"`  
   
   `/>`

3. Create a PublishExternalCatalogParams element.
   
   See “Example: Publish an Existing Catalog,” on page 235.

4. POST the PublishExternalCatalogParams element to the publishToExternalOrganizations URL.

Example: Publish an Existing Catalog

This request updates the catalog created in “Example: Create a Catalog,” on page 223 to publish it externally.

Request:

`POST https://vcloud.example.com/api/admin/catalog/32/action/publishToExternalOrganizations`

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

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

...<PublishExternalCatalogParams>`
Content Subscription Endpoint Specification

This document specifies the requirements for creating an endpoint that is compatible with a client using version 1 of the VMware Content Subscription Protocol (VCSP). VMware vCloud Director catalogs use this protocol when synchronizing content from an external subscription.

Introduction

VMware vCloud Director enables organizations to create catalogs that acquire their content from a subscription to an external source. The external source can be an externally published catalog hosted on another instance of vCloud Director or a Web site that hosts a VCSP endpoint.

Configuration parameters for vCloud Director catalogs that have external subscriptions include a VCSP endpoint URL and, if the endpoint requires authentication, a user name and password. Users periodically synchronize to update catalog contents from the subscribed endpoint. You can request synchronization for an entire catalog, or for individual catalog items. The process is similar in both cases.

1. The VCSP client makes a GET request to retrieve the endpoint descriptor. If the version in the descriptor matches the version of the catalog being synchronized, no content has changed at the endpoint, and no further action is required.

2. If the version in the descriptor is greater than the version of the catalog, the client makes a GET request to retrieve the endpoint index file.

3. To update the entire catalog, the client makes GET requests to retrieve all catalog items whose version at the endpoint is greater than their version in the catalog. To update an individual catalog item, the client makes a GET request to retrieve it only if its version number at the endpoint is greater than its version in the catalog.

The endpoint is a passive partner in this process. Its only responsibilities are to keep the version numbers up to date in the endpoint descriptor and endpoint index files, and to return the HTTP status codes and other responses expected by the VCSP client.

URLs

A VCSP endpoint URL must have a scheme of HTTP or HTTPS and a path that terminates in the name of the endpoint descriptor. A VCSP endpoint whose contents are stored on a host named vcsp.example.com in a directory named /vcsp on the host and whose descriptor file is named descriptor.json would have the following endpoint URL:

http://vcsp.example.com/vcsp/descriptor.json
Endpoint Descriptor

The endpoint descriptor is a file whose contents define the catalog available at the endpoint. The contents, which must be expressed in JavaScript Object Notation (media type application/json) as defined by RFC4627, define a JSON object, and must specify values for the following key names.

- **vcspVersion**: An integer that specifies the version of the VCSP protocol to which this endpoint conforms.
- **version**: An integer that specifies the version of the catalog, as described in "Version Numbers," on page 243.
- **id**: The object identifier, expressed in URN format. This value uniquely identifies the catalog, persists for the life of the catalog, and must never be reused.
- **name**: The display name of the catalog. Must be a nonempty string.
- **created**: Time and date when the endpoint was created.
- **itemType**: Must have value vcsp.CatalogItem.
- **itemsHref**: A reference to the endpoint index file relative to the location of the endpoint descriptor. Replacing the final component of the endpoint URL with the value of the itemsHref key must create a valid URL.
- **capabilities**: A JSON object that describes the capabilities of this catalog. The object must define all of the following properties:
  - **transferIn**: An array with a single member that must be the string httpGet.
  - **transferOut**: An array with a single member that must be the string httpGet.
  - **generateIds**: A Boolean value that must be set to true.
- **metadata**: An array of catalog object metadata. See “Metadata,” on page 241.
- **maintenanceMessage**: A string indicating that the endpoint is inaccessible because it is undergoing maintenance. For example:

  "maintenanceMessage" : "This catalog is currently in maintenance mode"

If the endpoint descriptor includes this key, attempts by subscribers to synchronize with this endpoint fail and display the string.

The following example shows a typical endpoint descriptor.

```json
{
    "vcspVersion" : "1",
    "version" : "10",
    "id" : "urn:uuid:ccdd2c56-e54e-4883-bc0a-619baaca92a6",
    "name" : "published",
    "created" : "2012-09-14T22:17:50.807Z",
    "itemType" : "vcsp.CatalogItem",
    "itemsHref" : "items",
    "capabilities" : {
        "transferIn" : [ "httpGet" ],
        "transferOut" : [ "httpGet" ],
```
Endpoint Index

The endpoint index is a file whose contents define the set of items available in the catalog. The contents, which must be expressed in JavaScript Object Notation (media type application/json) as defined by RFC4627, define a JSON object, and must specify values for the following key names.

**itemType**

Must be vcsp.CatalogItem.

**items**

An array of zero or more item objects. Each item object must specify values for the following key names.

**version**

An integer that specifies the version of the item, as described in “Version Numbers,” on page 243.

**id**

The object identifier of the item, expressed in URN format. This value uniquely identifies the item, persists for the life of the item, and must never be reused.

**name**

The display name of the item. Must be a nonempty string.

**created**

Time and date when the item was created.

**type**

Item type. Must have one of the following values:

- vcsp.iso if the item is an ISO image.
- vcsp.ovf if the item is an OVF package.
An array of file objects that includes all the files that represent the item. Each file object is represented as an array with three elements:

- **etag**: An integer representing the version of the file. The value of this key must be the same for each file in the array. When any file in this array gets updated, you must increment the value of the etag key for all files in the array.
- **name**: The name of the file.
- **hrefs**: An array of pathnames to the file. Must contain a single pathname to the file from the root of the endpoint, written as a URL fragment.

An array of additional properties of the item. This array has a single member:

- **selfHref**: A URL to the item descriptor for this item. See “Item Descriptors,” on page 241.

An array of catalog item metadata. See “Metadata,” on page 241.

If this item represents a vApp template, you must include an array representing the virtual machines referenced in the template.

- **name**: The name of the virtual machine.
- **metadata**: An array of virtual machine metadata. See “Metadata,” on page 241.

An integer that specifies the version of the endpoint index, as described in “Version Numbers,” on page 243.

The following example shows a typical endpoint index.

```json
{
  "itemType": "vcsp.CatalogItem",
  "items": [
    {
      "version": "1",
      "id": "urn:uuid:6dfa4596-a7c5-4d62-9a84-c559968baa26",
      "name": "vapp-demo",
      "created": "2012-09-17T17:59:15.161Z",
      "type": "vcsp.ovf",
      "files": [
        {
          "etag": "37",
          "name": "descriptor.ovf",
          "hrefs": [ "/vcsp/item/6dfa4596-a7c5-4d62-9a84-c559968baa26/file/descriptor.ovf" ]
        }
      ]
    }
  ]
}
```
Metadata

The endpoint descriptor and endpoint index allow you to include object metadata that is used to create vCloud Director object metadata when endpoint contents are synchronized by a subscribed catalog. For more information about the design and operation of vCloud Director object metadata, see Chapter 8, “Working With Object Metadata,” on page 307.

**metadata**

An array of object metadata. You may include an arbitrary number of metadata objects in this array, subject to the size limitations documented in “vCloud API Object Metadata Limits,” on page 309.

- **type**
  
  The type of the metadata value. One of STRING, BOOLEAN, DATETIME or NUMBER. See “vCloud API Object Metadata Contents,” on page 307.

- **domain**
  
  The access domain of the metadata. One of GENERAL or SYSTEM. See “Access Control for vCloud API Object Metadata,” on page 308. When vCloud Director object metadata is created from this endpoint, the metadata domain is always set to SYSTEM, regardless of the value you supply here.

- **key**
  
  An arbitrary key name.

- **value**
  
  The value of the key.

- **visibility**
  
  The visibility the metadata. One of PRIVATE, READONLY or READWRITE. See “Access Control for vCloud API Object Metadata,” on page 308. When vCloud Director object metadata is created from this catalog item, the metadata visibility is always set to READONLY, regardless of the value you supply here.

Item Descriptors

Every item listed in the endpoint index must include an item descriptor file in the directory that holds all the item's files. The contents, which must be expressed in JavaScript Object Notation (media type application/json) as defined by RFC4627, define a JSON object, and must specify values for the following key names.

- **version**
  
  An integer that specifies the version of the item, as described in “Version Numbers,” on page 243.

- **id**
  
  The object identifier, expressed in URN format. This value uniquely identifies the catalog, persists for the life of the catalog, and must never be reused.

- **name**
  
  The display name of the catalog. Must be a nonempty string.

- **created**
  
  Time and date when the endpoint was created.

- **description**
  
  A description of the item, as it will appear in the destination catalog.

- **type**
  
  Item type. Must have one of the following values:
  
  - vcsp.iso if the item is an ISO image.
  - vcsp.ovf if the item is an OVF package.
files
An array of file objects that includes all the files that represent the item. Each file object is represented as an array with two elements:

- **name** The name of the file.
- **hrefs** An array of pathnames to the file. Must contain a single pathname to the file from the root of the endpoint, written as a URL fragment.

properties
An array of additional properties of the item. Must be empty.

The following example shows a typical item descriptor for an OVF package.

```
{
  "version" : "1",
  "id" : "urn:uuid:b63c9bbe-3614-4153-82fd-d5f4916a1327",
  "name" : "template1",
  "created" : "2012-09-14T22:18:12.858Z",
  "description" : "Added with VMware Ovf Tool.",
  "type" : "vcsp.ovf",
  "files" : [ {
    "name" : "descriptor.ovf",
    "size" : 9985,
    "hrefs" : [ "descriptor.ovf" ]
  }, {
    "name" : "vm-3fe8a95b-ccd1-4816-b4ed-d631f3e2bbf3-disk-0.vmdk",
    "size" : 833536,
    "hrefs" : [ "vm-3fe8a95b-ccd1-4816-b4ed-d631f3e2bbf3-disk-0.vmdk" ]
  } ],
  "properties" : { }
}
```

Responses
VCSP clients retrieve catalog files, including the descriptor and the endpoint index, with GET requests. The endpoint must return one of the following responses:

- HTTP status 200 (OK) followed by the file content
- HTTP status 503 (Service unavailable) followed by a document that provides additional information. This document has media type `application/json`, and provides values for the following keys:

  - **status** Status of the requested file.
  - **progress** File generation progress, expressed as an integer in the range 0 to 100.
  - **message** An error message describing any errors that prevented the file from being generated.

The client continues to make GET requests until it receives either a 200 response or a 503 response with a nonempty message.

For example, an in-progress operation could return the following response.

```
{
  "status" : "",
  "progress" : 10
}
```
An operation that has encountered an error could return the following response.

```
{
    "status" : "failed",
    "message" : "File Generation failed"
}
```

**Authentication**

An endpoint can require authentication. VCSP clients always present the user name `vcsp` when logging in. The endpoint can specify any password for this user, but must accept the user name `vcsp`. The user name and password are encoded as specified for Basic HTTP authentication.

**Version Numbers**

Version numbers appear in the endpoint descriptor and endpoint index as version values, which are integer values that increment monotonically. It is the responsibility of the endpoint to increment the appropriate version value whenever any of the following changes occur.

- Changes to a catalog item
  - A file in the item is added, removed, or changed.
  - The name or description of the item is changed.

- Changes to a catalog
  - An item is added to or removed from the catalog.
  - The version value of any contained catalog item changes.
  - The name or description of the catalog is changed.

**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 “Enabling and Managing Federation,” on page 182.

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 244.

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 53. For more information about creating new roles, see “Create a Role,” on page 257.

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 244.

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 173. 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 244.

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 247. 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 173. 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
    IsExternal=true
    >
    <Role
        type="application/vnd.vmware.admin.role+xml"
        href="https://vcloud.example.com/api/admin/role/105" />
</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 246.

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 173. 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
  xmlns="http://www.vmware.com/vcloud/v1.5"
  name="Engineering"
  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" ...
  <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\8D\28</NameInSource>

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\8D\28</NameInSource>
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 you import a user from a SAML identity provider, you must include the domain name, such as user@domain.com. When you import a group from a SAML identity provider, you must use its fully distinguished name.

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 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.
Include the following line in the User or Group element.

```
<ProviderType>SAML</ProviderType>
```

POST the element to the organization’s users or groups URL.

**Example: Import a User from a SAML Identity Provider**

This example imports a user from the SAML identity provider defined by the organization. This example is identical to the one shown in “Example: Import a User from an LDAP Database,” on page 246, 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.

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 in the encoding used by the identity provider.

**Request:**

```
POST https://vcloud.example.com/api/admin/org/26/users
Content-Type: application/vnd.vmware.admin.user+xml
```

**Response:**

```
201 Created
Content-Type: application/vnd.vmware.admin.user+xml
```

```
<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>
```
Take Control of a User’s Objects

An organization administrator or system administrator can take control of a user's vApps, media, and catalogs.

When an administrator retrieves a User element, it contains a link that the administrator can use to take ownership of that user's vApps, media, and catalogs. This action is typically required when a user leaves the organization and an administrator needs to transfer ownership of that user's objects to other users.

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 user.
   Use a request like this one:
   ```
   GET https://vcloud.example.com/api/admin/user/85
   ```

2. Examine the response to find the link where rel="takeOwnership".
   ```
   <User ...>
   ...
   <Link rel="takeOwnership"
   href="https://vcloud.example.com/api/admin/user/85/action/takeOwnership"/>
   ...
   </User>
   ```

3. Make a POST request to the URL in the href value of that link.
   ```
   POST https://vcloud.example.com/api/admin/user/85/action/takeOwnership
   ```
   The response is empty (204 No Content).

All vApps, media, and catalogs that belonged to the user now belong to the administrator who made the request.

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 375

## Predefined Roles and Their Rights

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

### System Administrator

The system administrator has super-user rights for the entire system. System administrator credentials are established during installation and configuration. A system administrator can create additional system administrator accounts. All system administrators are members of the system organization. You cannot modify the rights associated with this role.

### Organization Roles

After creating an organization, a system administrator can assign the role of organization administrator to any user in the organization. An organization administrator has super-user rights within that organization, and can assign any of the predefined roles to the organization's users and groups.

<table>
<thead>
<tr>
<th>Role</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Organization Administrator</td>
<td>An organization administrator can assign the role of organization administrator to any member of an organization.</td>
</tr>
<tr>
<td>Catalog Author</td>
<td>The rights associated with the catalog author role allow a user to create and publish catalogs.</td>
</tr>
<tr>
<td>vApp Author</td>
<td>The rights associated with the vApp Author role allow a user to use catalogs and create vApps.</td>
</tr>
<tr>
<td>vApp User</td>
<td>The rights associated with the vApp User role allow a user to use existing vApps.</td>
</tr>
<tr>
<td>Console Access Only</td>
<td>The rights associated with the Console Access Only role allow a user to view virtual machine state and properties and to use the guest OS.</td>
</tr>
</tbody>
</table>

Each predefined role includes a set of default rights. If an organization administrator modifies the set of rights associated with a predefined role, those modifications apply only in the context of that organization. If a system administrator modifies the set of rights associated with a predefined role, those modifications apply to all organizations in the system.

You classify rights according to the objects to which they apply.

### Rights Associated with Catalogs

Admin rights are granted to the system administrator throughout the system, and to an organization administrator within the organization.

**Table 6-5. Rights Associated With Catalogs**

<table>
<thead>
<tr>
<th>Catalog: Add vApp from My Cloud</th>
<th>Description</th>
<th>Admin</th>
<th>Catalog Author</th>
<th>vApp Author</th>
<th>vApp User</th>
<th>Console Access Only</th>
</tr>
</thead>
<tbody>
<tr>
<td>Catalog: Change Owner</td>
<td>Permission to change the owner of a catalog.</td>
<td>X</td>
<td></td>
<td>X</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Each pre-defined role includes a set of default rights. If an organization administrator modifies the set of rights associated with a predefined role, those modifications apply only in the context of that organization. If a system administrator modifies the set of rights associated with a predefined role, those modifications apply to all organizations within the system.
### Table 6-5. Rights Associated With Catalogs (Continued)

<table>
<thead>
<tr>
<th>Description</th>
<th>Admin</th>
<th>Catalog Author</th>
<th>vApp Author</th>
<th>vApp User</th>
<th>Console Access Only</th>
</tr>
</thead>
<tbody>
<tr>
<td>Catalog: Create/Delete a Catalog</td>
<td>X</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Permission to create and delete catalogs.</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Catalog: Edit Catalog Properties</td>
<td>X</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Permission to edit catalog properties.</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Catalog: Allow External Publishing/Subscription for the Catalogs</td>
<td>X</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Permission to publish catalogs for external consumption and to subscribe to external catalog feeds.</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Catalog: Share a Catalog to Users/Groups within Current Organization</td>
<td>X</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Permission to share catalogs to users and groups in the same organization.</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Catalog: View Private and Shared Catalogs within Current Organization</td>
<td>X</td>
<td>X</td>
<td></td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>Permission to view both private and shared catalogs in the organization.</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Catalog: View Shared Catalogs from Other Organizations</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Permission to view catalogs shared from other organizations.</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

### Rights Associated with Independent Disks

### Table 6-6. Rights Associated With Independent Disks

<table>
<thead>
<tr>
<th>Description</th>
<th>Admin</th>
<th>Catalog Author</th>
<th>vApp Author</th>
<th>vApp User</th>
<th>Console Access Only</th>
</tr>
</thead>
<tbody>
<tr>
<td>Disk: Create</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Permission to create independent disks.</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Disk: Delete</td>
<td>X</td>
<td>X</td>
<td></td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>Permission to delete independent disks.</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Disk: Edit Properties</td>
<td>X</td>
<td>X</td>
<td></td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>Permission to edit the properties of an independent disk.</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Disk: View Properties</td>
<td>X</td>
<td>X</td>
<td></td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>Permission to view the properties of an independent disk.</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
### Rights Associated with vApp Templates and Media

Table 6-7. Rights Associated With vApp Templates and Media

<table>
<thead>
<tr>
<th>Description</th>
<th>Admin</th>
<th>Catalog Author</th>
<th>vApp Author</th>
<th>vApp User</th>
<th>Console Access Only</th>
</tr>
</thead>
<tbody>
<tr>
<td>Catalog Item: Add to My Cloud</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>Catalog Item: Copy/Move a vApp Template/Media</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Catalog Item: Create/Upload a vApp Template/Media</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Catalog Item: Enable vApp Template/Media Download</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Catalog Item: Edit vApp Template/Media Properties</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Catalog Item: View vApp Templates/Media</td>
<td>X</td>
<td></td>
<td>X</td>
<td>X</td>
<td></td>
</tr>
</tbody>
</table>

### Rights Associated with vApps

Table 6-8. Rights Associated With vApps

<table>
<thead>
<tr>
<th>Description</th>
<th>Admin</th>
<th>Catalog Author</th>
<th>vApp Author</th>
<th>vApp User</th>
<th>Console Access Only</th>
</tr>
</thead>
<tbody>
<tr>
<td>vApp: Change Owner</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>vApp: Copy a vApp</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>vApp: Create/Reconfigure a vApp</td>
<td>X</td>
<td></td>
<td>X</td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>vApp: Delete a vApp</td>
<td>X</td>
<td></td>
<td>X</td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>vApp: Download a vApp</td>
<td>X</td>
<td></td>
<td>X</td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>vApp: Edit vApp Properties</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td></td>
</tr>
</tbody>
</table>
Table 6-8. Rights Associated With vApps (Continued)

<table>
<thead>
<tr>
<th>Description</th>
<th>Admin</th>
<th>Catalog Author</th>
<th>vApp Author</th>
<th>vApp User</th>
<th>Console Access Only</th>
</tr>
</thead>
<tbody>
<tr>
<td>vApp: Edit VM CPU</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Permission to edit virtual machine CPUs.</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>vApp: Edit VM Hard Disk</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Permission to edit virtual machine hard disks.</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>vApp: Edit VM Memory</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>Permission to edit virtual machine memory</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>vApp: Edit VM Network</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>Permission to edit virtual machine network</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>configuration.</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>vApp: Edit VM Properties</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>Permission to edit virtual machine properties.</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>vApp: Manage VM Password Settings</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>Permission to edit virtual machine password</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>settings.</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>vApp: Start/Stop/Suspend/Reset a vApp</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>Permission to start, stop, suspend, and reset a vApp</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>vApp: Share a vApp</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>Permission to share vApps.</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>vApp: Create/Revert a VM Snapshot</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>Permission to create, revert, and delete virtual machine snapshots.</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>vApp: Upload a vApp</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>Permission to upload a vApp.</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>vApp: Access to a VM Console</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>Permission to use the virtual machine console.</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>vApp: View VM Metrics</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Permission to view virtual machine metrics.</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Administrative Rights**

All of these rights are granted to the system administrator throughout the system, and to an organization administrator within the organization. These rights are not granted to any other predefined role.
<table>
<thead>
<tr>
<th>Description</th>
<th>Admin</th>
<th>Catalog Author</th>
<th>vApp Author</th>
<th>vApp User</th>
<th>Console Access Only</th>
</tr>
</thead>
<tbody>
<tr>
<td>General: Administrator Control</td>
<td>Permission to use all administrator privileges.</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>General: Administrator View</td>
<td>Permission to view vCloud Director as an administrator.</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>General: Send Notification</td>
<td>Permission to send vCloud Director user notifications.</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Gateway: Configure Services</td>
<td>Permission to configure gateway services.</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Organization VDC Network: Edit Properties</td>
<td>Permission to edit the properties of an organization virtual data center network.</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Organization VDC Network: View Properties</td>
<td>Permission to view the properties of an organization virtual data center network.</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Organization VDC: Set Default Storage Policy</td>
<td>Permission to set the default storage policy for an organization virtual data center.</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Organization VDC: View Organization VDCs</td>
<td>Permission to view organization virtual data centers.</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Organization: Edit Federation Settings</td>
<td>Permission to edit an organization’s federation settings.</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Organization: Edit Leases Policy</td>
<td>Permission to edit an organization’s leases policy.</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Organization: Edit Organization Network Properties</td>
<td>Permission to edit an organization’s network properties</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Organization: Edit Organization Properties</td>
<td>Permission to edit organization properties.</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Organization: Edit Password Policy</td>
<td>Permission to edit an organization’s password policy.</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Organization: Edit Quotas Policy</td>
<td>Permission to edit an organization’s quotas policy.</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
### Table 6-9. Other Administrative Rights (Continued)

<table>
<thead>
<tr>
<th>Organization:</th>
<th>Description</th>
<th>Admin</th>
<th>Catalog Author</th>
<th>vApp Author</th>
<th>vApp User</th>
<th>Console Access Only</th>
</tr>
</thead>
<tbody>
<tr>
<td>Edit SMTP Settings</td>
<td>Permission to edit an organization's SMTP settings.</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Organization:</td>
<td>Permission to edit an organization's associations.</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Edit Organization</td>
<td>Permission to view organization networks.</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Associations</td>
<td>Permission to view organizations.</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

#### 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 that populate the Role, retrieve the administrative view of the cloud, using a request like this one:
   
   GET https://vcloud.example.com/api/admin
   
   The VCloud element returned by this request includes a RightReferences element that contains RightReference elements that show the name and href for each right defined in the cloud. For example:
   
   `<RightReference
type="application/vnd.vmware.admin.right+xml"
name="vApp: Copy"
href="https://vcloud.example.com/api/admin/right/4965b0e7-9ed8-371d-8b08-fc716d20bf4b" />
   
2. POST the Role element to the organization's add URL for roles.
   
   See the request portion of “Example: Create a Role,” on page 257.

   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 173. The rights associated with this new role are less comprehensive than those associated with the built-in vApp Author role, but still include rights to perform many common vApp operations.

**Note** This example uses href attributes that contain actual UUID values for specific rights, since these are invariant across vCloud Director installations and releases.
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: Copy"
         href="https://vcloud.example.com/api/admin/right/4965b0e7-9ed8-371d-8b08-fc716d20bf4b" />
      <RightReference
         type="application/vnd.vmware.admin.right+xml"
         name="vApp: Create / Reconfigure"
         href="https://vcloud.example.com/api/admin/right/2dc8abec-2e0d-3789-a5f9-ce04531f0b53" />
      <RightReference
         type="application/vnd.vmware.admin.right+xml"
         name="vApp: Delete"
         href="https://vcloud.example.com/api/admin/right/df05c07f-c537-3777-8d9b-a9cfe8d49014" />
      <RightReference
         type="application/vnd.vmware.admin.right+xml"
         name="vApp: Edit Properties"
         href="https://vcloud.example.com/api/admin/right/c2a29357-1b2a-3f9d-9cd6-de3d525d49f3" />
      <RightReference
         type="application/vnd.vmware.admin.right+xml"
         name="vApp: Edit VM CPU"
         href="https://vcloud.example.com/api/admin/right/729a3828-8b63-31b2-88db-f56612a06722" />
      <RightReference
         type="application/vnd.vmware.admin.right+xml"
         name="vApp: Edit VM Hard Disk"
         href="https://vcloud.example.com/api/admin/right/cd02b5f8-c54a-334a-b782-5d31a1d77d85" />
      <RightReference
         type="application/vnd.vmware.admin.right+xml"
         name="vApp: Edit VM Memory"
         href="https://vcloud.example.com/api/admin/right/c6c827dc-fc42-33a8-844f-8ab5a91f8a6c" />
      <RightReference
         type="application/vnd.vmware.admin.right+xml"
         name="vApp: Edit VM Network"
         href="https://vcloud.example.com/api/admin/right/f24fffde-f953-3976-9f2b-8b355b25881d" />
      <RightReference
         type="application/vnd.vmware.admin.right+xml"
         name="vApp: Edit VM Properties"
         href="https://vcloud.example.com/api/admin/right/5250ab79-8f50-33f9-8af5-015cb39c380b" />
   </RightReferences>
</Role>
The response is a Role element, most of which does not appear in this excerpt. The response includes links that an administrator can use to edit or remove the 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>
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 172.

This chapter includes the following topics:

- “Summary of System Administration Requests,” on page 261
- “Retrieve or Update System Settings,” on page 265
- “Attach a vCenter Server,” on page 266
- “Finding Available vCenter Resources,” on page 268
- “Create a Provider VDC,” on page 277
- “Create an External Network,” on page 287
- “Create a Network Pool,” on page 290
- “Import a Virtual Machine from vCenter,” on page 296
- “Relocate a Virtual Machine to a Different Datastore,” on page 299
- “System Truststore and Keytab Maintenance,” on page 301
- “Retrieve the vSphere URL of an Object,” on page 304

Summary of System Administration Requests

System administration requests allow a system administrator to create and manage resources consumed by all organizations in a cloud, and 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>
<thead>
<tr>
<th>Table 7-1. Summary of System Administration Requests</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Operation</strong></td>
</tr>
<tr>
<td>Share a catalog with specific organizations in your cloud.</td>
</tr>
<tr>
<td>Retrieve a list of Provider VDCs in the system.</td>
</tr>
<tr>
<td>Retrieve a list of external networks in the system.</td>
</tr>
<tr>
<td>Retrieve a list of network pools in the system.</td>
</tr>
<tr>
<td>Retrieve a list of vCenter servers registered to vCloud Director.</td>
</tr>
<tr>
<td>Retrieve information about a vCenter server.</td>
</tr>
<tr>
<td>Retrieve a list of available resource pools on a vCenter server.</td>
</tr>
<tr>
<td>Update vCenter server settings.</td>
</tr>
<tr>
<td>Register a vCenter server and vShield manager.</td>
</tr>
<tr>
<td>Unregister a vCenter server and vShield manager.</td>
</tr>
<tr>
<td>Force reconnection to a vCenter server.</td>
</tr>
<tr>
<td>Retrieve a list of hosts in the system.</td>
</tr>
<tr>
<td>Retrieve information about an ESXi host.</td>
</tr>
<tr>
<td>Operation</td>
</tr>
<tr>
<td>-------------------------</td>
</tr>
<tr>
<td>Prepare an ESXi host.</td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td>Unprepare an ESXi host.</td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td>Enable an ESXi host.</td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td>Disable an ESXi host.</td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td>Repair an ESXi host.</td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td>Upgrade an ESXi host.</td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td>Create a Provider VDC.</td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td>Retrieve a Provider VDC.</td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td>Update a Provider VDC.</td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td>Enable a Provider VDC.</td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td>Disable a Provider VDC.</td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td>Delete a Provider VDC.</td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td>Create an external network.</td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td>Retrieve an external network.</td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td>Update an external network.</td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td>Delete an external network.</td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td>Create a network pool.</td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td>Operation</td>
</tr>
<tr>
<td>-----------</td>
</tr>
<tr>
<td>Retrieve a network pool.</td>
</tr>
<tr>
<td>Update a network pool.</td>
</tr>
<tr>
<td>Delete a network pool.</td>
</tr>
<tr>
<td>Import a virtual machine from vCenter as a vApp.</td>
</tr>
<tr>
<td>Import a virtual machine from vCenter as a vApp template.</td>
</tr>
<tr>
<td>Retrieve the representation of a datastore.</td>
</tr>
<tr>
<td>Enable a datastore.</td>
</tr>
<tr>
<td>Disable a datastore.</td>
</tr>
<tr>
<td>Update a datastore.</td>
</tr>
<tr>
<td>Delete a datastore.</td>
</tr>
<tr>
<td>Import a virtual machine from vCenter into an existing vApp.</td>
</tr>
<tr>
<td>Import a media image from vCenter.</td>
</tr>
<tr>
<td>Enable a resource pool.</td>
</tr>
<tr>
<td>Disable a resource pool.</td>
</tr>
<tr>
<td>Place a vApp in maintenance mode to prevent users from changing vApp metadata.</td>
</tr>
<tr>
<td>Remove a vApp from maintenance mode.</td>
</tr>
</tbody>
</table>
Table 7-1. Summary of System Administration Requests (Continued)

<table>
<thead>
<tr>
<th>Operation</th>
<th>Request</th>
<th>Request Body</th>
<th>Response</th>
</tr>
</thead>
<tbody>
<tr>
<td>Migrate a virtual machines to a different resource pool.</td>
<td>POST API-URL/admin/extension/resourcePool/id/action/migrateVms</td>
<td>MigrateParams</td>
<td>Task</td>
</tr>
<tr>
<td>Retrieve a URL that you can use to access a vSphere object with a vSphere client.</td>
<td>GET API-URL/admin/extension/vimServer/id/vimObjType/vimObjMoref/vSphereWebClientUrl</td>
<td>None</td>
<td>VSphereWebClientUrl</td>
</tr>
<tr>
<td>Merge Provider VDCs.</td>
<td>PUT API-URL/admin/extension/providervdc/id/action/merge</td>
<td>None</td>
<td>204 No Content</td>
</tr>
<tr>
<td>Retrieve the list of storage profiles that exist on a vCenter server.</td>
<td>GET API-URL/admin/extension/vimServer/id/storageProfiles</td>
<td>None</td>
<td>VMWStorageProfiles</td>
</tr>
<tr>
<td>Refresh the list of storage profiles that exist on a vCenter server.</td>
<td>POST API-URL/admin/extension/vimServer/id/action/refreshStorageProfiles</td>
<td>None</td>
<td>Task</td>
</tr>
<tr>
<td>Register vCloud Director so that it appears as an extension in the vSphere Client Solutions Manager tab.</td>
<td>POST API-URL/admin/extension/vimServer/id/action/refresh</td>
<td>None</td>
<td>Task</td>
</tr>
</tbody>
</table>

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
- Customize the branding of the vCloud Director Web Console 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 organizations.
LookupServiceSettings
Specify the vSphere Lookup Service URL so system administrators can authenticate using vSphere Single Sign-On (SSO).

CatalogSettings
Specify the schedule for background synchronization of catalogs that have an external subscription. These settings apply to all organizations in the system.

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
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:

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

Create a RegisterVimServerParams element that includes the information required to register the vCenter server and vShield manager.

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 267.

**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
<?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

```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
        type="application/vnd.vmware.admin.vmwarevirtualcenter+xml"
        name="VC-22"
        href="https://vcloud.example.com/api/admin/extension/vimServer/100">
```
Finding Available vCenter Resources

Many of the operations required to import virtual machines or create Provider VDCs, external networks, and network pools require you to identify vCenter resources and obtain references to them. You use these references to make the vCenter resources available in the cloud.

Every vCenter server registered to your cloud is represented as a VimServerReference element in the cloud’s vimServerReferences list. You can retrieve one of these references to get a detailed representation of the server object, including links to the server’s resource pools, networks, 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 VimServerReference elements. You can make a GET request to any of these references to retrieve the XML representation of a vCenter server registered to this cloud.

   The VimServer element returned in response to the request you made in Step 2 includes several Link elements where rel="down". These links contain URLs that you can use to retrieve lists of references to vCenter resources on this server.

<table>
<thead>
<tr>
<th>List URL</th>
<th>List Contents</th>
</tr>
</thead>
<tbody>
<tr>
<td><a href="https://vcloud.example.com/api/admin/extension/vimServer/id/vmsList">https://vcloud.example.com/api/admin/extension/vimServer/id/vmsList</a></td>
<td>References to virtual machines in this vCenter server's inventory</td>
</tr>
<tr>
<td><a href="https://vcloud.example.com/api/admin/extension/vimServer/id/hostReferences">https://vcloud.example.com/api/admin/extension/vimServer/id/hostReferences</a></td>
<td>References to hosts in this vCenter server's inventory</td>
</tr>
<tr>
<td><a href="https://vcloud.example.com/api/admin/extension/vimServer/id/networks">https://vcloud.example.com/api/admin/extension/vimServer/id/networks</a></td>
<td>VimObjectRef elements of type DV_PORTGROUP and NETWORK</td>
</tr>
<tr>
<td><a href="https://vcloud.example.com/api/admin/extension/vimServer/id/storageProfiles">https://vcloud.example.com/api/admin/extension/vimServer/id/storageProfiles</a></td>
<td>VMWStorageProfile objects</td>
</tr>
</tbody>
</table>

**Example: Resources on a vCenter Server**

Request:

```
GET https://vcloud.example.com/api/admin/extension/vimServer/9
```
Retrieve a List of Resource Pools from a vCenter Server

You can retrieve the list of resource pools available on a vCenter server registered to a cloud. To retrieve the list, you make a GET request to the server's resourcePoolList link.

The ResourcePoolList of a VimServer element contains an entry for every available resource pool on the server. Resource pools that a provider VDC is already using are not listed, because they are considered unavailable. See “Finding Available vCenter Resources,” on page 268.
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 268.

Procedure

1. Examine the VimServer element to locate its resourcePoolList link.

   The link has the following form:

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

2. GET the URL in the value of this link's href attribute to retrieve the list of resource pools.

   See “Example: Retrieve a List of Resource Pools from a vCenter Server,” on page 270. 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: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:VimObjectRef>
    </vmext:DataStoreRefs>
  </vmext:ResourcePool>
  ...
```

vCloud API Programming Guide

270  VMware, Inc.
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_PORTGoup, 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 268.

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 GET the URL in the value of this link's href attribute to retrieve the list of network resources.
   See “Example: Retrieve a List of Available Portgroups and Switches from a vCenter Server,” on page 271. 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
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 287 and “Create a Network Pool,” on page 290.

Note When you create a Provider VDC, a VxLANPoolType network pool is created automatically on the vCenter server that backs the Provider VDC. See “Create a Network Pool,” on page 290.

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:

```xml
<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.


**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:

```xml
<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 vmsList link.

When you import a virtual machine from vCenter, your request must supply a reference to the vCenter server and a VIM object reference to the virtual machine. See “Finding Available vCenter Resources,” on page 268.

**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:

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

3 GET the URL in the value of this link's href attribute to retrieve the list of virtual machines.

See the request portion of “Example: Retrieve a List of Virtual Machines from a vCenter Server,” on page 274. If the list is empty, there are no virtual machines 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

```xml
...<vmext:VmObjectRefsList
  xmlns:vmext="http://www.vmware.com/vcloud/extension/v1.5"
  page="1"
  numberOfPages="1"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" ... >
  <vmext:VmObjectRef
    name="RH5u3_32bit">
    <vmext:VimServerRef
      type="application/vnd.vmware.admin.vmwvirtualcenter+xml"
      name="vc2-v41u1"
      href="https://vcloud.example.com/api/admin/extension/vimServer/9" />
    <vmext:MoRef>vm-41</vmext:MoRef>
    <vmext:VimObjectType>VIRTUAL_MACHINE</vmext:VimObjectType>
  </vmext:VmObjectRef>
  <vmext:VmObjectRef
    name="W2K3 64 R2">
    <vmext:VimServerRef
      type="application/vnd.vmware.admin.vmwvirtualcenter+xml"
      name="vc2-v41u1"
      href="https://vcloud.example.com/api/admin/extension/vimServer/9" />
    <vmext:MoRef>vm-43</vmext:MoRef>
    <vmext:VimObjectType>VIRTUAL_MACHINE</vmext:VimObjectType>
  </vmext:VmObjectRef>
  <vmext:VmObjectRef
    name="Ubuntu91_32_vt4">
    <vmext:VimServerRef
      type="application/vnd.vmware.admin.vmwvirtualcenter+xml"
      name="vc2-v41u1"
      href="https://vcloud.example.com/api/admin/extension/vimServer/9" />
    <vmext:MoRef>vm-44</vmext:MoRef>
    <vmext:VimObjectType>VIRTUAL_MACHINE</vmext:VimObjectType>
  </vmext:VmObjectRef>
</vmext:VmObjectRefsList>
...
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 268.

NOTE Storage profiles are represented as Storage Policies in the vCloud Director Web console.

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 268.

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. GET the URL in the value of this link's href attribute to retrieve the list of storage profiles.

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
<?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:VimObjectType>STORAGE_PROFILE</vmext:VimObjectType>
   <vmext:FreeStorageMb>169203.0</vmext:FreeStorageMb>
   <vmext:TotalStorageMb>325120.0</vmext:TotalStorageMb>
   <vmext:DataStoreRefs>
     <vmext:VimObjectRef>
       <vmext:VimServerRef
         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:VimObjectType>DATASTORE</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>datastore-45</vmext:MoRef>
       <vmext:VimObjectType>DATASTORE</vmext:VimObjectType>
     </vmext:VimObjectRef>
   </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 268.

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/providervdcsparams"/>
   ```
   
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` element and element from the resource pool list. See the request portion of “Example: Create a Provider VDC,” on page 278.

   ```
   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 278.
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 290.

**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 270 and a storage profile extracted from “Example: Retrieve a List of Storage Profiles from a vCenter Server,” on page 276. The vCenter server that provides the resources is referenced in the `VimServerRef` and `VimServer` elements.

Request:

POST https://vcloud.example.com/api/admin/extension/providervdcsparams

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>
  <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:VimServer>  
<vmext:VimServerRef 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>
Chapter 7 Managing and Monitoring a Cloud

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

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 278. 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
<?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="up"
   type="application/vnd.vmware.admin.vmwprovidervdc+xml"
   href="https://vcloud.example.com/api/admin/extension/providervdc/35" />
   <vmext:VMWProviderVdcResourcePool
   primary="false">
   <vcloud:Link
   rel="up"
   type="application/vnd.vmware.admin.vmwprovidervdc+xml"
   href="https://vcloud.example.com/api/admin/extension/providervdc/35" />
</vmext:VMWProviderVdcResourcePoolSet
</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 269.

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"
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 278. 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 269 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
<?xml version="1.0" encoding="UTF-8"?>  
<vmext:UpdateResourcePoolSetParams  
xmlns:vmext="http://www.vmware.com/vcloud/extension/v1.5"  
xmlns:vcloud="http://www.vmware.com/vcloud/v1.5">  
<vmext:AddItem>  
<vmext:VimServerRef  
type="application/vnd.vmware.admin.vmwvirtualcenter+xml"  
href="https://vcloud.example.com/api/admin/extension/vimServer/9" />  
<vmext:MoRef>resgroup-230</vmext:MoRef>  
<vmext:VimObjectType>RESOURCE_POOL</vmext:VimObjectType>  
</vmext:AddItem>  
</vmext:UpdateResourcePoolSetParams>
```

Response:

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

```xml
<?xml version="1.0" encoding="UTF-8"?>  
<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 281. The response is a 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.

**NOTE** Storage profiles are represented as Storage Policies in the vCloud Director Web console.

**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 278. 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 275 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
<?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

```xml
<Task ... >
  ...  
</Task>
```
Merge Provider VDCs

You can merge two Provider VDCs by specifying a target Provider VDC and a contributor Provider VDC to merge with it. The merged Provider VDC contains all resources from the target and contributor Provider VDCs.

Because merging Provider VDCs is a resource-intensive operation, vCloud Director allows only a single contributor to a merge. You can merge multiple Provider VDCs into a single target by making multiple merge requests, each specifying the same target and a new contributor.

When the merge is complete:

- Networks, network pools, storage profiles, resource pools, and datastores from the contributor Provider VDC are available in the target Provider VDC.
- Organization VDCs that were backed by the contributor are now backed by the target.
- The contributor Provider VDC is deleted.

Prerequisites

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

Procedure

1 Identify the Provider VDCs to merge.

   Select a contributor that is 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 whose ProviderVdcReference references the contributor.

   See the request portion of “Example: Merge Provider VDCs,” on page 286.

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 merges the Provider VDC at https://vcloud.example.com/api/admin/extension/providervdc/46 with the target Provider VDC specified in the request URL.

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
   ...
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 271.

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 288.
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 271.

Request:

**POST** https://vcloud.example.com/api/admin/extension/externalnets

```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**

```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" 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.vmwwirtualcenter+xml"
    href="https://vcloud.example.com/api/admin/extension/vimServer/9" />
  <vmext:MoRef>network-175</vmext:MoRef>
  <vmext:VimObjectType>NETWORK</vmext:VimObjectType>
</vmext:VimPortGroupRef>
</vmext:VMWExternalNetwork>
Create a Network Pool

Network pools provide support for isolated and routed networks in organization VDCs. Although every Provider VDC includes a VXLAN network pool that can support most networking use cases, a system administrator can create other types of network pools if they are needed.

A network pool object represents a collection of vSphere network resources that are contained by a Provider VDC and available to the organization VDCs backed by that Provider VDC. 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 `VxlanPoolType` 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 `VMNetworkPool` element. The contents of this element depend on its type, which is specified in its `xsi:type` attribute. The following values of `xsi:type` are supported for pools created by a system administrator.

**VlanPoolType**  
This pool type is based on ESXi VLANs on the vCenter server that backs the Provider VDC, and is backed by a range of VLAN IDs.

**FencePoolType**  
This pool type 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.

**PortGroupPoolType**  
This pool type is based on distributed port groups of a vSphere distributed switch or third-party distributed switch on the vCenter server that backs the Provider VDC.

**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:

   ```xml
   <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 290.
- 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 271

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 291.

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 291.

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

<?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"
  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:Description>Example VLAN-backed network pool</vcloud:Description>
  <vcloud:Tasks>
    <vcloud:Task
      vCloud API Programming Guide
      292  VMware, Inc.
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 290.
- 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 271

**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 293.

**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"
...
```
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:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:type="vmext:FencePoolType"
  name="example-isolation-pool"
  type="application/vnd.vmware.admin.networkPool+xml">
  <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: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 290.
- 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 271

**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 295.
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 290.

**Example: Create a Portgroup-Backed Network Pool**

Request:

```xml
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
    href="https://vcloud.example.com/api/admin/extension/vimServer/9"/>
  <vmext:MoRef>dvportgroup-32</vmext:MoRef>
  <vmext:VimObjectType>DV_PORTGROUP</vmext:VimObjectType>
</vmext:PortGroupRefs>
</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.vmmNetworkPoolReferences+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 273.
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 273.

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 274. 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.importVmsAsVAppParams+xml
...
<?xml version="1.0" encoding="UTF-8"?>
<ImportVmAsVAppParams
   xmlns="http://www.vmware.com/vcloud/extension/v1.5"
   name="ImportedWin2K8"
   sourceMove="false">
```
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 273.

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 274 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.
Relocate a Virtual Machine to a Different Datastore

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.

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 164, 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>
System Truststore and Keytab Maintenance

You can use the vCloud API to upload and manage SSL certificates, keystores, and Kerberos keytabs for the system LDAP and AMQP services. You can also use the vCloud API to configure SSPI, the Microsoft Security Support Provider Interface, for use with Active Directory.

Valid SSL certificates and truststores are required if the system LDAP and AMQP services use the secure sockets layer (https).

The SystemSettings element contains several elements that enable the system administrator to maintain certificates and truststores for the system LDAP and AMQP services. Links in the LdapSettings element allow the system administrator to manage the system LDAP truststore and keystore.

```xml
<SystemSettings ... >
  ...
  <LdapSettings ... >
    ...
    <vcloud:Link
      rel="certificate:update"
    <vcloud:Link
      rel="certificate:reset"
      href="https://vcloud.example.com/api/admin/extension/settings/ldapSettings/action/resetLdapCertificate"/>
    <vcloud:Link
      rel="keystore:update"
      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"
      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"/>
  ...
  </LdapSettings>
  ...
</SystemSettings>
```
The AmqpSettings element includes links that allow the system administrator to manage the system AMQP truststore and certificate.

```xml
<SystemSettings ...
...
<AmqpSettings>
...
<vcloud:Link
rel="certificate:update"
<vcloud:Link
rel="certificate:reset"
<vcloud:Link
rel="truststore:update"
href="https://vcloud.example.com/api/admin/extension/settings/amqp/action/updateAmqpTruststore" type="application/vnd.vmware.admin.trustStoreUpdateParams+xml"/>
<vcloud:Link
rel="truststore:reset"
href="https://vcloud.example.com/api/admin/extension/settings/amqp/action/resetAmqpTruststore"/>
...
</AmqpSettings>
...
</SystemSettings>
```

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

<table>
<thead>
<tr>
<th>Upload Type</th>
<th>Maximum Size in Megabytes</th>
</tr>
</thead>
<tbody>
<tr>
<td>vCenter truststore</td>
<td>10</td>
</tr>
<tr>
<td>LDAP certificate</td>
<td>2</td>
</tr>
<tr>
<td>LDAP keystore</td>
<td>2</td>
</tr>
<tr>
<td>LDAP SSPI keytab</td>
<td>2</td>
</tr>
<tr>
<td>AMQP certificate</td>
<td>2</td>
</tr>
<tr>
<td>AMQP truststore</td>
<td>10</td>
</tr>
</tbody>
</table>

**Prerequisites**

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

**Procedure**

1. Determine whether the request requires a body.
   
   Requests whose rel value includes the string reset do not require a body. For details about other request bodies, see the schema reference.

2. POST the request to the request URL.
   
   Include the request body if one is required.
3 Take any action that the response requires.

The response to an update request includes an uploadLocation parameter whose value is a URL to which you can upload the certificate, keytab, or keystore with a PUT request. Requests whose rel value includes the string reset return a No Content response and require no further action.

Example: Upload an SSL Certificate for the System LDAP Service

This example uploads an SSL certificate whose size is 892 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/extension/settings/ldap/action/updateLdapCertificate
Content-type: application/vnd.vmware.admin.certificateUpdateParams+xml

```xml
<?xml version="1.0" encoding="UTF-8"?>
<CertificateUpdateParams
  fileSize="892"
  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:

```xml
<CertificateUploadSocket
  xmlns="http://www.vmware.com/vcloud/v1.5"
  uploadLocation="https://vcloud.example.com/transfer/53bc1/ldapCertificate">
  <Task
    status="running"
    operation="Updating LDAP certificate"
  />
</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: 892

```xml
...serialized contents of certificate...
```

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 vimObjectType and vimObjMoref identify.
- **vimObjectType** 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 305. 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 281, “Example: Retrieve a List of Available Portgroups and Switches from a vCenter Server,” on page 271, and “Example: Retrieve a List of Storage Profiles from a vCenter Server,” on page 276.

---

**Note** See “Mapping a vCloud Director Object to a vSphere Object,” on page 305 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 305.

**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 281.

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 object to its representation in vSphere.

<table>
<thead>
<tr>
<th>vCloud Director Object</th>
<th>vSphere Object</th>
</tr>
</thead>
<tbody>
<tr>
<td>Provider VDC:Hosts</td>
<td>Host</td>
</tr>
<tr>
<td>Provider VDC:Datastores</td>
<td>Datastore</td>
</tr>
<tr>
<td>Provider VDC:Storage Profiles</td>
<td>Storage Profile</td>
</tr>
<tr>
<td>Provider VDC:External Network (portgroup-backed)</td>
<td>Standard virtual switch</td>
</tr>
<tr>
<td>Provider VDC:External Network(VLAN-backed)</td>
<td>Distributed virtual switch</td>
</tr>
<tr>
<td>Provider VDC:Resource Pool (root resource pool or Cluster)</td>
<td>Cluster</td>
</tr>
<tr>
<td>Provider VDC:Resource Pool (sub-resource pool only)</td>
<td>Resource Pool</td>
</tr>
<tr>
<td>External Network (Portgroup-backed)</td>
<td>Standard virtual switch</td>
</tr>
<tr>
<td>Direct Org VDC Networks (Portgroup)</td>
<td>Standard virtual switch</td>
</tr>
<tr>
<td>External Network (VLAN-backed)</td>
<td>Distributed virtual switch</td>
</tr>
<tr>
<td>Direct Org VDC Networks (VLAN)</td>
<td>Distributed virtual switch</td>
</tr>
<tr>
<td>Network Pool (VLAN-backed)</td>
<td>Distributed virtual switch</td>
</tr>
<tr>
<td>Network Pools (Portgroup-backed)</td>
<td>Standard virtual switch</td>
</tr>
<tr>
<td><strong>vCloud Director Object</strong></td>
<td><strong>vSphere Object</strong></td>
</tr>
<tr>
<td>---------------------------</td>
<td>-------------------</td>
</tr>
<tr>
<td>vCenter</td>
<td>vCenter Server</td>
</tr>
<tr>
<td>Resource Pool (root</td>
<td>Cluster</td>
</tr>
<tr>
<td>resource pool or Cluster</td>
<td></td>
</tr>
<tr>
<td>Resource Pool (sub-resource</td>
<td>Resource Pool</td>
</tr>
<tr>
<td>pool only)</td>
<td></td>
</tr>
<tr>
<td>Host</td>
<td>Host</td>
</tr>
<tr>
<td>Datastores</td>
<td>Datastore</td>
</tr>
<tr>
<td>Datastore Clusters</td>
<td>Datastore Cluster</td>
</tr>
<tr>
<td>Storage Profiles</td>
<td>N/A</td>
</tr>
<tr>
<td>vDSwitch</td>
<td>Distributed virtual switch</td>
</tr>
<tr>
<td>Port Group</td>
<td>Standard virtual switch</td>
</tr>
<tr>
<td>Home:vApp</td>
<td>Folder</td>
</tr>
<tr>
<td>vApp</td>
<td>Folder</td>
</tr>
<tr>
<td>vApp:vApp Diagram</td>
<td>Folder</td>
</tr>
<tr>
<td>vApp:vApp Diagram:VM</td>
<td>VM</td>
</tr>
<tr>
<td>vApp:VM List:VM</td>
<td>VM</td>
</tr>
<tr>
<td>vApp:Networking:vApp</td>
<td>Distributed Port Group</td>
</tr>
<tr>
<td>Networks</td>
<td></td>
</tr>
<tr>
<td>VM</td>
<td>VM</td>
</tr>
<tr>
<td>Expired Items:vApps</td>
<td>Folder</td>
</tr>
<tr>
<td>Expired Items:vApp</td>
<td>Folder</td>
</tr>
<tr>
<td>Templates</td>
<td></td>
</tr>
<tr>
<td>Catalog:vApp</td>
<td>Folder</td>
</tr>
<tr>
<td>Catalog:vApp Template</td>
<td>Folder</td>
</tr>
<tr>
<td>Catalog:vApp Template:VM</td>
<td>VM</td>
</tr>
<tr>
<td>Catalog:vApp Template:Shadow VMs</td>
<td>VM</td>
</tr>
<tr>
<td>Organization VDC:vApp</td>
<td>Folder</td>
</tr>
<tr>
<td>Organization VDC:vApp</td>
<td>Folder</td>
</tr>
<tr>
<td>Template</td>
<td></td>
</tr>
<tr>
<td>Organization VDC:Storage</td>
<td>Storage Profile</td>
</tr>
<tr>
<td>Profiles</td>
<td></td>
</tr>
<tr>
<td>Organization VDC:Org VDC</td>
<td>Distributed virtual switch</td>
</tr>
<tr>
<td>Networks (VLAN-backed)</td>
<td></td>
</tr>
<tr>
<td>Organization VDC:Org VDC</td>
<td>Standard virtual switch</td>
</tr>
<tr>
<td>Networks (Portgroup-backed)</td>
<td></td>
</tr>
<tr>
<td>Organization VDC:Resource</td>
<td>Cluster</td>
</tr>
<tr>
<td>Pools (root resource pool</td>
<td></td>
</tr>
<tr>
<td>or Cluster)</td>
<td></td>
</tr>
<tr>
<td>Organization VDC:Resource</td>
<td>Resource Pool</td>
</tr>
<tr>
<td>Pools (sub-resource pool</td>
<td></td>
</tr>
<tr>
<td>only)</td>
<td></td>
</tr>
</tbody>
</table>
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 329).

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.

```xml
<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.

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


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**

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

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, only one of which can be specified explicitly.

**GENERAL**

This is the default access domain for metadata, and cannot be explicitly specified in a Domain element. A MetadataEntry that does not include a Domain element is considered to be in the GENERAL domain. 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.

**SYSTEM**

To be placed in the SYSTEM domain, a MetadataEntry must include a Domain element with a value of 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

<table>
<thead>
<tr>
<th>Value</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>PRIVATE</td>
<td>The metadata is visible to system administrators only.</td>
</tr>
<tr>
<td>READONLY</td>
<td>the metadata is visible to system administrators and the object owner.</td>
</tr>
</tbody>
</table>

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 309
- “Retrieve or Update a Metadata Value,” on page 312

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 312.

**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`

<table>
<thead>
<tr>
<th>Cannot Contain</th>
<th>Cannot Start with</th>
<th>Cannot End with</th>
</tr>
</thead>
<tbody>
<tr>
<td>/../</td>
<td>./</td>
<td>/</td>
</tr>
<tr>
<td>../.</td>
<td>./</td>
<td>/</td>
</tr>
<tr>
<td>../</td>
<td>../</td>
<td></td>
</tr>
</tbody>
</table>

Prerequisites

Verify that you have rights to read or modify the metadata. See “Access Control for vCloud API Object Metadata,” on page 308
**Procedure**

1. Retrieve the representation of the object.

   Examine the response to find its metadata link. This example shows the metadata link from an AdminOrg.

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

2. Retrieve the Metadata element.

   If the object has no metadata, the element contains only a rel="add" link that you can use to add metadata and a rel="up" link that references the containing object, as shown in this example.

   ```xml
   <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="up"
       type="application/vnd.vmware.vcloud.org+xml"
       href="https://vcloud.example.com/api/admin/org/37" />
     <Link
       rel="add"
       type="application/vnd.vmware.vcloud.metadata+xml"
       href="https://vcloud.example.com/api/admin/org/26/metadata" />
   </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 307.

4. POST the Metadata element to the rel="add" link described in Step 2.

   See “Example: Update a Metadata Element,” on page 310.

**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:

```plaintext
POST https://vcloud.example.com/api/admin/org/26/metadata
```
The response is a Task.

Response:

```xml
<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

```xml
<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"/>
  </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 309
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 313

   **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 310) 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:

```xml
<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 307.
- 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 316
- “Packaged Queries,” on page 320
- “Query Parameters,” on page 325
- “Add a Metadata Filter to a Query,” on page 329
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 325.
- filter is an optional filter expression. At most one filter expression is allowed. See “Filter Expressions,” on page 326.

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 325.

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&amp;format=references" />
  <Link
    rel="down"
    type="application/vnd.vmware.vcloud.query.records+xml"
    name="organization"
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:

```xml
<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"
```
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:**

```xml
<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:

```xml
<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=adminUser&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&format=references" />
  <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
<?xml version="1.0" encoding="UTF-8"?>
<UserReferences
  xmlns="http://www.vmware.com/vcloud/v1.5"
  total="2"
  pageSize="25"
  page="1"
  name="user"
```
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 325.
- **filter** is an optional filter expression. At most one filter expression is allowed. See “Filter Expressions,” on page 326.

Query Categories

Packaged queries are divided into the following categories:

**User queries**

The queries have the form `API-URL/object-type/query`. Any user can run these queries.

**Administrator queries**

The queries have the form `API-URL/admin/object-type/query`. An organization administrator can run these queries.

**Extension queries**

The queries have the form `API-URL/admin/extension/object-type/query`. A system administrator can run these queries.

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 325.

Table 9-1. Packaged Queries

<table>
<thead>
<tr>
<th>Query URL</th>
<th>Result Set</th>
</tr>
</thead>
<tbody>
<tr>
<td>API-URL/catalogs/query</td>
<td>All catalogs in your organization that you have rights to view or modify</td>
</tr>
<tr>
<td>API-URL/mediaList/query</td>
<td>All media that you can view or modify</td>
</tr>
<tr>
<td>API-URL/vAppTemplates/query</td>
<td>All vApp templates that you can view or modify</td>
</tr>
<tr>
<td>API-URL/vApps/query</td>
<td>All vApps that you can view or modify</td>
</tr>
<tr>
<td>API-URL/vms/query</td>
<td>All virtual machines that you can view or modify</td>
</tr>
<tr>
<td>API-URL/admin/groups/query</td>
<td>Groups in all organizations in the system</td>
</tr>
<tr>
<td>API-URL/admin/users/query</td>
<td>Users in all organizations in the system</td>
</tr>
<tr>
<td>API-URL/admin/strandedUsers/query</td>
<td>Stranded users in the organization. When you delete an LDAP group, users who were imported from that group become stranded and cannot log in.</td>
</tr>
<tr>
<td>API-URL/admin/roles/query</td>
<td>All roles defined in the system</td>
</tr>
<tr>
<td>API-URL/admin/rights/query</td>
<td>All rights defined in the system</td>
</tr>
<tr>
<td>API-URL/admin/orgs/query</td>
<td>All organizations visible to you</td>
</tr>
<tr>
<td>API-URL/admin/vdcs/query</td>
<td>All VDCs in the system</td>
</tr>
<tr>
<td>API-URL/admin/extension/hostReferences/query</td>
<td>All hosts registered to the system</td>
</tr>
<tr>
<td>API-URL/admin/extension/datastores/query</td>
<td>All datastores in the system</td>
</tr>
<tr>
<td>API-URL/admin/extension/externalNetworkReferences/query</td>
<td>External networks in the system</td>
</tr>
<tr>
<td>API-URL/admin/extension/networkPoolReferences/query</td>
<td>All network pools in the system</td>
</tr>
<tr>
<td>API-URL/admin/extension/providerVdcReferences/query</td>
<td>All provider VDCs in the system</td>
</tr>
<tr>
<td>API-URL/admin/extension/vimServerReferences/query</td>
<td>All vCenter servers registered to the system</td>
</tr>
<tr>
<td>API-URL/admin/extension/orgNetworks/query</td>
<td>All organization networks in the system</td>
</tr>
<tr>
<td>API-URL/admin/extension/vapps/query</td>
<td>All vApps in the system</td>
</tr>
<tr>
<td>API-URL/admin/extension/orgVdcs/query</td>
<td>All VDCs in the system</td>
</tr>
</tbody>
</table>

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"
<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="2011-03-22T17:10:067-07:00"
  href="https://vcloud.example.com/api/catalog/43" />

<CatalogRecord
  ownerName="system"
Packaged query using references format.

GET https://vcloud.example.com/api/catalogs/query?format=references

Response:

```xml
<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?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?format=records"
    type="catalog" page="1" pageSize="25"
  />
  <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?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?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"
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**

<table>
<thead>
<tr>
<th>Parameter Name</th>
<th>Parameter Description</th>
<th>Default</th>
</tr>
</thead>
<tbody>
<tr>
<td>type</td>
<td>The type of the query. Type names are case-sensitive. See “Query Types,” on page 316.</td>
<td>None. This parameter is required for all typed queries, and is not allowed for any packaged query.</td>
</tr>
<tr>
<td>sortAsc=attribute-name</td>
<td>Sort results by <code>attribute-name</code> in ascending order. <code>attribute-name</code> cannot include metadata. See the vCloud API Schema Reference for information about which attributes can be used with <code>sortDesc</code>.</td>
<td>Sorted by database ID</td>
</tr>
<tr>
<td>sortDesc=attribute-name</td>
<td>Sort results by <code>attribute-name</code> in descending order. <code>attribute-name</code> cannot include metadata. See the vCloud API Schema Reference for information about which attributes can be used with <code>sortDesc</code>.</td>
<td>Sorted by database ID</td>
</tr>
<tr>
<td>page</td>
<td>If the query results span multiple pages, return this page.</td>
<td><code>page=1</code></td>
</tr>
<tr>
<td>pageSize</td>
<td>Number of results per page, to a maximum of 128.</td>
<td><code>pageSize=25</code></td>
</tr>
<tr>
<td>format</td>
<td>One of the following types:</td>
<td><code>format=records</code></td>
</tr>
<tr>
<td>references</td>
<td>Returns a reference to each object, including its <code>name</code>, <code>type</code>, and <code>href</code> attributes.</td>
<td></td>
</tr>
<tr>
<td>records</td>
<td>Returns all database records for each object, with each record as an attribute.</td>
<td></td>
</tr>
<tr>
<td>idrecords</td>
<td>Identical to the <code>records</code> format, except that object references are returned in <code>id</code> format rather than <code>href</code> format.</td>
<td></td>
</tr>
<tr>
<td>fields</td>
<td>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 326.</td>
<td>Returns all static attribute names. Returns metadata only if the object has a non-empty <code>Metadata</code> element.</td>
</tr>
<tr>
<td>offset</td>
<td>Integer value specifying the first record to return. Record numbers &lt; <code>offset</code> are not returned.</td>
<td><code>offset=0</code></td>
</tr>
<tr>
<td>filterEncoded</td>
<td>Specify whether the filter expression is encoded as described in RFC3986 (pct-encoded). See “Encoding Filter Expressions,” on page 328.</td>
<td><code>filterEncoded=false</code></td>
</tr>
<tr>
<td>filter</td>
<td>Filter expression. See Table 9-3</td>
<td>None</td>
</tr>
</tbody>
</table>
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

<table>
<thead>
<tr>
<th>Operator</th>
<th>Example</th>
<th>Operation</th>
</tr>
</thead>
<tbody>
<tr>
<td>==</td>
<td>attribute==value</td>
<td>Matches. The example evaluates to true if attribute has a value that matches value in a case-sensitive comparison. <strong>Note</strong> Asterisk (*) characters that appear anywhere in value are treated as wildcards that match any character string. When value includes wildcards, the comparison with attribute becomes case-insensitive.</td>
</tr>
<tr>
<td>!=</td>
<td>attribute!=value</td>
<td>Does not match. The example evaluates to true if attribute has a value that does not match value in a case-sensitive comparison. Wildcard characters are not allowed.</td>
</tr>
<tr>
<td>;</td>
<td>attribute1==value1;attribute2!=value2</td>
<td>Logical AND. The example evaluates to true only if attribute1 has a value that matches value1 and attribute2 has a value that does not match value2 in a case-sensitive comparison.</td>
</tr>
<tr>
<td>,</td>
<td>attribute1==value1,attribute2==value2</td>
<td>Logical OR. The example evaluates to true if attribute1 has a value that matches value1 or attribute2 has a value that matches value2 in a case-sensitive comparison.</td>
</tr>
<tr>
<td>=gt=</td>
<td>attribute=gt=value</td>
<td>Greater than. The example evaluates to true if attribute has a value that is greater than value. Both attribute and value must be of type int, long, or dateTime.</td>
</tr>
<tr>
<td>=lt=</td>
<td>attribute=lt=value</td>
<td>Less than. The example evaluates to true if attribute has a value that is less than value. Both attribute and value must be of type int, long, or dateTime.</td>
</tr>
<tr>
<td>=ge=</td>
<td>attribute=ge=value</td>
<td>Greater than or equal to. The example evaluates to true if attribute has a value that is greater than or equal to value. Both attribute and value must be of type int, long, or dateTime.</td>
</tr>
<tr>
<td>=le=</td>
<td>attribute=le=value</td>
<td>Less than or equal to. The example evaluates to true if attribute has a value that is less than or equal to value. Both attribute and value must be of type int, long, or dateTime.</td>
</tr>
</tbody>
</table>

Not all attributes can be used in a filter expression. For details, see the reference pages for query result types in the vCloud API Schema Reference.

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**

<table>
<thead>
<tr>
<th>Type Name as Specified in TypedValue</th>
<th>Type Name as a Filter Expression Keyword</th>
</tr>
</thead>
<tbody>
<tr>
<td>MetadataStringValue</td>
<td>STRING</td>
</tr>
<tr>
<td>MetadataNumberValue</td>
<td>NUMBER</td>
</tr>
<tr>
<td>MetadataDateTimeValue</td>
<td>DATETIME</td>
</tr>
<tr>
<td>MetadataBooleanValue</td>
<td>BOOLEAN</td>
</tr>
</tbody>
</table>

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**

<table>
<thead>
<tr>
<th>Operator</th>
<th>Example</th>
<th>Operation</th>
</tr>
</thead>
<tbody>
<tr>
<td>=gt=</td>
<td>metadata:attribute=gt=[NUMBER</td>
<td>DATETIME]:value</td>
</tr>
<tr>
<td>=lt=</td>
<td>metadata:attribute=lt=[NUMBER</td>
<td>DATETIME]:value</td>
</tr>
<tr>
<td>=ge=</td>
<td>metadata:attribute=ge=[NUMBER</td>
<td>DATETIME]:value</td>
</tr>
<tr>
<td>=le=</td>
<td>metadata:attribute=le=[NUMBER</td>
<td>DATETIME]:value</td>
</tr>
</tbody>
</table>

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`
Encoding Filter Expressions

When you are comparing a *key* with a literal *value* that includes characters that might be subject to interpretation when used in a URL (often termed unsafe characters), the *value* must be encoded as described in RFC3986 (pct-encoded). For example, to create a filter expression to match a `hostName` property whose value is 12&345, encode the value as shown here:

```
filter=hostName==12%26345
```

When a filter expression includes metadata, you must encode both the *key* and the *value* this way. In addition, if you specify `filterEncoded=true` as part of a query that includes metadata, you must encode the `%` symbol as `%25`. For example, this query, which includes metadata would require this sort of additional encoding if you specified `filterEncoded=true`.

```
https://vcloud.example.com/api/query?
type=organization&format=records
&fields=metadata:rank,metadata@SYSTEM:expiry
&filter=metadata@SYSTEM:expiry=lt=DATETIME:DATETIME%253A2012-05-01T00%3A00%3A00.000-04%3A00
&filterEncoded=true
```

If you did not specify `filterEncoded=true`, you would not need the additional encoding.

```
https://vcloud.example.com/api/query?
type=organization&format=records
&fields=metadata:rank,metadata@SYSTEM:expiry
&filter=metadata@SYSTEM:expiry=lt=DATETIME:DATETIME%20%05-01T00%253A00%253A00.000-04%253A00
```

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. You can use the schema reference to find the attribute names included in a particular result set.

1. In the vCloud API Schema Reference, select All Queries.
2. Type the query name (or any part of it) in the Quick Index window, then open the query reference page.
3. On the query reference page, click the link in the Record Result section to open a page that shows the name, type, description, and other information for each attribute returned by the query. Attributes that can be used in a filter expression have a YES in the FILTER column. Attributes that can be used with sortAsc or sortDesc have a YES in the SORT column.
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:
   
   \[\text{fields=metadata:} \text{key-name} \[, \text{metadata:} \text{key-name} \] \ldots\]

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 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.

\[\text{GET https://vcloud.example.com/api/query?type=organization&format=records&fields=metadata:rank,metadata@SYSTEM:expiry}^\text{"}\]

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:

\[
\text{<QueryResultRecords xmlns="http://www.vmware.com/vcloud/v1.5" total="3" pageSize="25" page="1" ...}
\text{<OrgRecord storedVMQuota="0" numberOfVdcs="5" ... href="https://vcloud.example.com/api/org/70" numberOfRunningVMs="3">}
\text{<Metadata>}
\text{<MetadataEntry>}
\text{<Domain>}
\]
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>

**NOTE** When a filter expression includes metadata, you must encode both the *key* and the *value* as described in RFC3986. See “Encoding Filter Expressions,” on page 328.
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. Notifications are always generated in two formats:

- An XML document, which is sent to the AMQP exchange specified in the system AmqpSettings.
- A JSON object, which is sent to an AMQP exchange whose name has the form prefix.notifications20, where prefix is the value of the AmqpPrefix element in the system AmqpSettings.
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 338.

### 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 333. 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 334
- “Retrieve or Update Blocking Task Settings,” on page 344
- “Monitor Blocking Tasks,” on page 349
- “Take Action on a Blocking Task,” on page 350
- “Extend The Timeout Expiration of an Active Task,” on page 352

### 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 336.

### 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 335

Example: Update AMQP Settings

This request modifies the AMQP settings for a cloud to require the use of SSL for AMQP connections. It also overrides the default value for AmqpPrefix, vcd, with a new value, myCloud.

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>systemExchange</AmqpExchange>
    <AmqpVHost>/</AmqpVHost>
    <AmqpUseSSL>true</AmqpUseSSL>
    <AmqpSslAcceptAll>false</vmext:AmqpSslAcceptAll>
    <AmqpPrefix>myCloud</vmext:AmqpPrefix>
  </AmqpSettings>

The response includes information supplied in the request, and contains a number of Link elements inserted by the server. 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"
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 265.
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 335 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 335

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>
    <AmqpSslAcceptAll>false</AmqpSslAcceptAll>
    <AmqpPrefix>myCloud</AmqpPrefix>
</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

...<vnext:AmqpSettingsTest
    xmlns:vnext="http://www.vmware.com/vcloud/extension/v1.5"
Notification Message Format

All messages that the vCloud Director AMQP service sends contain an AMQP routing key and a Notification element.

The Notification element is defined in the vCloud API schema. The routing key format is defined by the AMQP specification.

Routing Key Format

The routing key for a vCloud Director AMQP message has the following form:

```
operationSuccess.entity.org.user.subType1.subType2...subTypeN.[taskName]
```

Routing key components include:

- **operationSuccess**: A Boolean value denoting whether the operation that triggered the notification succeeded or failed.
- **entity**: The object identifier of the object on which an operation, an event of type com/vmware/vcloud/event/, triggered the notification. For more information about object identifiers, see “Objects, References, and Representations,” on page 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 subType is a single component of the event type name. See “Notification Types,” on page 340.
- **taskName**: If entity is a task or blocking task, the task name is appended to the routing key.

The following routing key, in which the object identifiers are truncated to save space, is an example of a routing key that might have been created for a successful com/vmware/vcloud/event/vapp/create event:

```
true.dc6a-xxx.0b8a-xxx.832c-xxx.com.vmware.vcloud.event.vapp.create
```

Notification Headers

The vCloud API defines notification headers and prepends them to every notification.

**Table 10-1. Notification Headers**

<table>
<thead>
<tr>
<th>Header</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>notification.type</td>
<td>See “Notification Types,” on page 340.</td>
</tr>
<tr>
<td>notification.entityType</td>
<td>The type of vCloud entity is associated with this notification. For example, vm.</td>
</tr>
</tbody>
</table>
### Table 10-1. Notification Headers (Continued)

<table>
<thead>
<tr>
<th>Header</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>notification.entityUUID</td>
<td>The object identifier of the object on which an operation, an event of type com/vmware/vcloud/event/, triggered the notification.</td>
</tr>
<tr>
<td>notification.orgUUID</td>
<td>The object identifier of the organization that owns the affected object.</td>
</tr>
<tr>
<td>notification.userUUID</td>
<td>The object identifier of the user who made the request.</td>
</tr>
<tr>
<td>notification.operationSuccess</td>
<td>A Boolean value denoting whether the operation that triggered the notification succeeded or failed.</td>
</tr>
</tbody>
</table>

### Example: Notification Message Format

A notification message can be formatted as XML or JSON. XML notifications are published on the AMQP exchange specified in the system AMQP settings. The JSON notifications are published on an exchange created by vCloud Director with a name of the form `prefix.notifications20` where `prefix` is the value of the `AmqpPrefix` element in the system `AmqpSettings`.

Here is an example of a message generated by a blocking task, delivered as an XML document.

```xml
<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>
```
An XML 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 392.

Here is the same message delivered as a JSON object.

```json
{
    "eventId" : "a1440dd8-60ae-46c7-b216-44693bc00c90",
    "type" : "com/vmware/vcloud/event/blockingtask/create",
    "timestamp" : "2011-06-18T14:33:27.787+03:00",
    "operationSuccess" : true,
    "user" : "urn:vcloud:user:44",
    "org" : "urn:vcloud:org:70",
    "entity" : "urn:vcloud:blockingTask:25"
    "task" : "urn:vcloud:task:34",
    "taskOwner" : "urn:vcloud:vapp:26"
}
```

This request retrieves the blocking task that generated this notification.

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 351.

### 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`. Notification types can be grouped based on the object type affected by the event.

#### User, Group, Role, and Session Events

<table>
<thead>
<tr>
<th>Type (com/vmware/vcloud/event/)</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>session/login</td>
<td>A login session was created.</td>
</tr>
<tr>
<td>user/import</td>
<td>A user was imported from LDAP.</td>
</tr>
<tr>
<td>user/remove</td>
<td>An imported user was removed from the organization.</td>
</tr>
<tr>
<td>user/modify</td>
<td>One or more properties of a user were modified.</td>
</tr>
<tr>
<td>user/lockout</td>
<td>An account was locked based on the organization's password policy settings.</td>
</tr>
<tr>
<td>user/unlock</td>
<td>A locked account was unlocked.</td>
</tr>
<tr>
<td>user/lock_expired</td>
<td>The lock on an account has expired.</td>
</tr>
<tr>
<td>user/create</td>
<td>A local user was created in an organization.</td>
</tr>
<tr>
<td>user/delete</td>
<td>A local user was removed from the organization.</td>
</tr>
<tr>
<td>group/import</td>
<td>A group was imported from LDAP.</td>
</tr>
<tr>
<td>group/remove</td>
<td>A group was removed from an organization.</td>
</tr>
<tr>
<td>role/create</td>
<td>A new role was created.</td>
</tr>
</tbody>
</table>
### Table 10-2. User, Group, Role, and Session Events (Continued)

<table>
<thead>
<tr>
<th>Type (com/vmware/vcloud/event/)</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>role/modify</td>
<td>An existing role was modified.</td>
</tr>
<tr>
<td>role/delete</td>
<td>A role was deleted.</td>
</tr>
</tbody>
</table>

### Organization, Network, Catalog, and VDC Events

### Table 10-3. Organization, Network, Catalog, and VDC Events

<table>
<thead>
<tr>
<th>Type (com/vmware/vcloud/event/)</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>org/create</td>
<td>An organization was created.</td>
</tr>
<tr>
<td>org/modify</td>
<td>An organization was modified.</td>
</tr>
<tr>
<td>org/delete</td>
<td>An organization was deleted.</td>
</tr>
<tr>
<td>network/create</td>
<td>A network was created.</td>
</tr>
<tr>
<td>network/modify</td>
<td>A network was modified.</td>
</tr>
<tr>
<td>network/delete</td>
<td>A network was deleted.</td>
</tr>
<tr>
<td>network/deploy</td>
<td>A network was deployed.</td>
</tr>
<tr>
<td>network/undeploy</td>
<td>A network was undeployed.</td>
</tr>
<tr>
<td>catalog/create</td>
<td>A catalog was created.</td>
</tr>
<tr>
<td>catalog/delete</td>
<td>A catalog was deleted.</td>
</tr>
<tr>
<td>catalog/modify</td>
<td>One or more properties of a catalog were modified</td>
</tr>
<tr>
<td>catalog/publish</td>
<td>A catalog was published.</td>
</tr>
<tr>
<td>catalogItem/create</td>
<td>An item was added to a catalog.</td>
</tr>
<tr>
<td>catalogItem/delete</td>
<td>An item was removed from a catalog.</td>
</tr>
<tr>
<td>vdc/create_request</td>
<td>A request to create a VDC was blocked pending administrative action.</td>
</tr>
<tr>
<td>vdc/create</td>
<td>A VDC was created.</td>
</tr>
<tr>
<td>vdc/modify</td>
<td>One or more properties of a VDC was modified.</td>
</tr>
<tr>
<td>vdc/delete_request</td>
<td>A request to delete a VDC was blocked pending administrative action.</td>
</tr>
<tr>
<td>vdc/delete</td>
<td>A VDC was deleted.</td>
</tr>
<tr>
<td>vdc/fast_provisioning/modify</td>
<td>The UsesFastProvisioning value of a VDC was modified.</td>
</tr>
<tr>
<td>vdc/thin_provisioning/modify</td>
<td>The IsThinProvision value of a VDC was modified.</td>
</tr>
</tbody>
</table>

### vApp, vApp Template, Vm, and Media Events

### Table 10-4. vApp, vApp Template, Vm, and Media Events

<table>
<thead>
<tr>
<th>Type (com/vmware/vcloud/event/)</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>vappTemplate/create</td>
<td>A vApp template was created.</td>
</tr>
<tr>
<td>vappTemplate/import</td>
<td>A virtual machine was imported from vSphere as a vApp template.</td>
</tr>
<tr>
<td>vappTemplate/modify</td>
<td>One or more properties of a vApp template were modified.</td>
</tr>
<tr>
<td>vappTemplate/delete</td>
<td>A vApp template was deleted.</td>
</tr>
</tbody>
</table>
Table 10-4. vApp, vApp Template, Vm, and Media Events (Continued)

<table>
<thead>
<tr>
<th>Type (com/vmware/vcloud/event/)</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>vappTemplate/create_request</code></td>
<td>A request to create a vApp template was blocked pending administrative action.</td>
</tr>
<tr>
<td><code>vappTemplate/import_request</code></td>
<td>A request to import a vApp template was blocked pending administrative action.</td>
</tr>
<tr>
<td><code>vappTemplate/modify_request</code></td>
<td>A request to modify a vApp template was blocked pending administrative action.</td>
</tr>
<tr>
<td><code>vappTemplate/delete_request</code></td>
<td>A request to delete a vApp template was blocked pending administrative action.</td>
</tr>
<tr>
<td><code>vapp/create</code></td>
<td>A vApp was created (instantiated)</td>
</tr>
<tr>
<td><code>vapp/import</code></td>
<td>A virtual machine was imported from vSphere as a vApp.</td>
</tr>
<tr>
<td><code>vapp/modify</code></td>
<td>One or more properties of a vApp were modified.</td>
</tr>
<tr>
<td><code>vapp/delete</code></td>
<td>A vApp was deleted.</td>
</tr>
<tr>
<td><code>vapp/deploy</code></td>
<td>A vApp was deployed.</td>
</tr>
<tr>
<td><code>vapp/undeploy</code></td>
<td>A vApp was undeployed.</td>
</tr>
<tr>
<td><code>vapp/runtime_lease_expiry</code></td>
<td>The runtime lease of a vApp has expired.</td>
</tr>
<tr>
<td><code>vapp/create_request</code></td>
<td>A request to instantiate a vApp template was blocked pending administrative action.</td>
</tr>
<tr>
<td><code>vapp/import_request</code></td>
<td>A request to import a vApp was blocked pending administrative action.</td>
</tr>
<tr>
<td><code>vapp/modify_request</code></td>
<td>A request to modify a vApp was blocked pending administrative action.</td>
</tr>
<tr>
<td><code>vapp/delete_request</code></td>
<td>A request to delete a vApp was blocked pending administrative action.</td>
</tr>
<tr>
<td><code>vapp/deploy_request</code></td>
<td>A request to deploy a vApp was blocked pending administrative action.</td>
</tr>
<tr>
<td><code>vapp/undeploy_request</code></td>
<td>A request to undeploy a vApp was blocked pending administrative action.</td>
</tr>
<tr>
<td><code>vm/create_request</code></td>
<td>A request to create a virtual machine was blocked pending administrative action.</td>
</tr>
<tr>
<td><code>vapp/quarantine_reject</code></td>
<td>An uploaded OVF was rejected after quarantine.</td>
</tr>
<tr>
<td><code>vapp/upload_timeout</code></td>
<td>An OVF upload has timed out.</td>
</tr>
<tr>
<td><code>vapp/lease_expiration_changed</code></td>
<td>The lease expiration of a vApp has changed.</td>
</tr>
<tr>
<td><code>vm/ip_address_changed</code></td>
<td>The IP address of a virtual machine has changed.</td>
</tr>
<tr>
<td><code>vm/create</code></td>
<td>A virtual machine was created by instantiating a vApp.</td>
</tr>
<tr>
<td><code>vm/modify_request</code></td>
<td>A request to modify a virtual machine was blocked pending administrative action.</td>
</tr>
<tr>
<td><code>vm/modify</code></td>
<td>One or more properties of a virtual machine were modified.</td>
</tr>
<tr>
<td><code>vm/delete</code></td>
<td>A virtual machine was deleted.</td>
</tr>
<tr>
<td><code>vm/change_state</code></td>
<td>The power state of a virtual machine has changed.</td>
</tr>
<tr>
<td><code>vm/deploy_request</code></td>
<td>A request to deploy a virtual machine was blocked pending administrative action.</td>
</tr>
<tr>
<td><code>vm/deploy</code></td>
<td>A virtual machine was deployed.</td>
</tr>
</tbody>
</table>
### Table 10-4. vApp, vApp Template, Vm, and Media Events (Continued)

<table>
<thead>
<tr>
<th>Type (com/vmware/vcloud/event/)</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>vm/undeploy_request</td>
<td>A request to undeploy a virtual machine was blocked pending administrative action.</td>
</tr>
<tr>
<td>vm/undeploy</td>
<td>A virtual machine was undeployed.</td>
</tr>
<tr>
<td>vm/consolidate_request</td>
<td>A request to consolidate a virtual machine was blocked pending administrative action.</td>
</tr>
<tr>
<td>vm/consolidate</td>
<td>A virtual machine was consolidated.</td>
</tr>
<tr>
<td>vm/relocate_request</td>
<td>A request to relocate a virtual machine was blocked pending administrative action.</td>
</tr>
<tr>
<td>vm/relocate</td>
<td>A virtual machine was relocated.</td>
</tr>
<tr>
<td>media/create</td>
<td>A media object was created by upload or import.</td>
</tr>
<tr>
<td>media/import</td>
<td>A media object was imported.</td>
</tr>
<tr>
<td>media/modify</td>
<td>One or more properties of a media object were modified.</td>
</tr>
<tr>
<td>media/delete</td>
<td>A media object was deleted.</td>
</tr>
<tr>
<td>media/create_request</td>
<td>A request to create a media object was blocked pending administrative action.</td>
</tr>
<tr>
<td>media/import_request</td>
<td>A request to import a media object was blocked pending administrative action.</td>
</tr>
<tr>
<td>media/modify_request</td>
<td>A request to modify a media object was blocked pending administrative action.</td>
</tr>
<tr>
<td>media/delete_request</td>
<td>A request to delete a media object was blocked pending administrative action.</td>
</tr>
<tr>
<td>media/upload_timeout</td>
<td>A media upload has timed out.</td>
</tr>
<tr>
<td>media/quarantine_reject</td>
<td>An uploaded media object was rejected after quarantine.</td>
</tr>
</tbody>
</table>

### Other System Events

#### Table 10-5. Other System Events

<table>
<thead>
<tr>
<th>Type (com/vmware/vcloud/event/)</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>providerVdc/create_request</td>
<td>A request to create a provider VDC was blocked pending administrative action.</td>
</tr>
<tr>
<td>providerVdc/create</td>
<td>A provider VDC was created.</td>
</tr>
<tr>
<td>providerVdc/modify</td>
<td>One or more properties of a provider VDC were modified.</td>
</tr>
<tr>
<td>providerVdc/delete_request</td>
<td>A request to delete a provider VDC was blocked pending administrative action.</td>
</tr>
<tr>
<td>providerVdc/delete</td>
<td>A provider VDC was deleted.</td>
</tr>
<tr>
<td>vc/create</td>
<td>A vCenter server was registered.</td>
</tr>
<tr>
<td>vc/modify</td>
<td>One or more properties of a registered vCenter server were modified.</td>
</tr>
<tr>
<td>vc/delete</td>
<td>A registered vCenter server was registered.</td>
</tr>
<tr>
<td>task/create</td>
<td>A task was created.</td>
</tr>
<tr>
<td>task/start</td>
<td>A non-blocking task has started or a blocking task has resumed.</td>
</tr>
<tr>
<td>task/abort</td>
<td>A task was aborted.</td>
</tr>
<tr>
<td>task/complete</td>
<td>A task has completed.</td>
</tr>
</tbody>
</table>
Table 10-5. Other System Events (Continued)

<table>
<thead>
<tr>
<th>Type (com/vmware/vcloud/event/)</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>task/fail</td>
<td>A task has failed.</td>
</tr>
<tr>
<td>blockingtask/create</td>
<td>A task was blocked and a notification created.</td>
</tr>
<tr>
<td>blockingtask/resume</td>
<td>A blocking task was resumed.</td>
</tr>
<tr>
<td>blockingtask/abort</td>
<td>A blocking task was aborted.</td>
</tr>
<tr>
<td>blockingtask/fail</td>
<td>A blocking task was failed.</td>
</tr>
<tr>
<td>datastore/modify</td>
<td>One or more properties of a datastore object were modified.</td>
</tr>
<tr>
<td>datastore/delete</td>
<td>A datastore object was deleted.</td>
</tr>
</tbody>
</table>

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 339.

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 346.

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 265.

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:

   ```html
   <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 345.
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 346 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

...?

<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 344.

Note These operation names also appear in the operationName attribute of the Task element that tracks the operation.

Table 10-6. Blocking Task Operations for vApp Tasks

<table>
<thead>
<tr>
<th>Operation Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>vappAttachDisk</td>
<td>Attach an independent disk to any virtual machine in a vApp.</td>
</tr>
<tr>
<td>vappCheckVmCompliance</td>
<td>Check the storage profile compliance of a virtual machine.</td>
</tr>
<tr>
<td>vappConsolidateVm</td>
<td>Consolidate the disks of any virtual machine in a vApp.</td>
</tr>
<tr>
<td>vappCreateSnapshot</td>
<td>Create a snapshot for the vApp.</td>
</tr>
<tr>
<td>vappDeleteShadowVm</td>
<td>Delete a shadow VM for any virtual machine in a vApp.</td>
</tr>
<tr>
<td>vappDeploy</td>
<td>Deploy a vApp.</td>
</tr>
<tr>
<td>vappDetachDisk</td>
<td>Detach an independent disk from any virtual machine in a vApp.</td>
</tr>
<tr>
<td>vappDiscardSuspendedState</td>
<td>Discard the suspended state of any virtual machine in a vApp.</td>
</tr>
<tr>
<td>vappEjectCdFloppy</td>
<td>Eject media from any virtual machine in a vApp.</td>
</tr>
<tr>
<td>vappInsertCdFloppy</td>
<td>Insert media in any virtual machine in a vApp.</td>
</tr>
<tr>
<td>vappInstallTools</td>
<td>Install VMware Tools on any virtual machine in a vApp.</td>
</tr>
<tr>
<td>vappMigrateVms</td>
<td>Migrate the virtual machines in a vApp to a new host.</td>
</tr>
<tr>
<td>vappPowerOff</td>
<td>Power-off a vApp.</td>
</tr>
<tr>
<td>vappRebootGuest</td>
<td>Reboot any virtual machine in a vApp.</td>
</tr>
<tr>
<td>vappRelocateVm</td>
<td>Relocate any virtual machine in a vApp.</td>
</tr>
<tr>
<td>vappRemoveAllSnapshots</td>
<td>Remove all snapshots for this vApp.</td>
</tr>
<tr>
<td>vappReset</td>
<td>Reset any virtual machine in a vApp.</td>
</tr>
<tr>
<td>vappResume</td>
<td>Resume any virtual machine in a vApp.</td>
</tr>
<tr>
<td>vappRevertToCurrentSnapshot</td>
<td>Revert the vApp to its current snapshot.</td>
</tr>
<tr>
<td>vappShutdownGuest</td>
<td>Shut down any virtual machine in a vApp.</td>
</tr>
<tr>
<td>vappSuspend</td>
<td>Suspend any virtual machine in a vApp.</td>
</tr>
<tr>
<td>vappUndeployPowerOff</td>
<td>Undeploy any virtual machine in a vApp by powering it off.</td>
</tr>
<tr>
<td>vappUndeploySuspend</td>
<td>Undeploy any virtual machine in a vApp by suspending it.</td>
</tr>
<tr>
<td>vappUpdateVm</td>
<td>Modify one or more properties of a virtual machine.</td>
</tr>
<tr>
<td>vappUpgradeHwVersion</td>
<td>Upgrade the hardware version of any virtual machine in a vApp.</td>
</tr>
</tbody>
</table>

Table 10-7. Blocking Task Operations for VDC Tasks

<table>
<thead>
<tr>
<th>Operation Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>vdcCaptureTemplate</td>
<td>Capture a vApp as a vApp template.</td>
</tr>
<tr>
<td>vdcComposeVapp</td>
<td>Compose a vApp.</td>
</tr>
<tr>
<td>vdcCopyMedia</td>
<td>Copy a media object.</td>
</tr>
</tbody>
</table>
### Table 10-7. Blocking Task Operations for VDC Tasks (Continued)

<table>
<thead>
<tr>
<th>Operation Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>vdcCopyTemplate</td>
<td>Copy a vApp template.</td>
</tr>
<tr>
<td>vdcCopyVapp</td>
<td>Clone a vApp.</td>
</tr>
<tr>
<td>vdcCreateDisk</td>
<td>Create an independent disk.</td>
</tr>
<tr>
<td>vdcCreateVdc</td>
<td>Create an organization VDC.</td>
</tr>
<tr>
<td>vdcDeleteDisk</td>
<td>Delete an independent disk.</td>
</tr>
<tr>
<td>vdcDeleteMedia</td>
<td>Delete a media object.</td>
</tr>
<tr>
<td>vdcDeleteTemplate</td>
<td>Delete a vApp template.</td>
</tr>
<tr>
<td>vdcDeleteVapp</td>
<td>Delete a vApp.</td>
</tr>
<tr>
<td>vdcDeleteVdc</td>
<td>Delete an organization VDC.</td>
</tr>
<tr>
<td>vdcEnableDownload</td>
<td>Enable a vApp template for download as OVF.</td>
</tr>
<tr>
<td>vdcInstantiateVapp</td>
<td>Instantiate a vApp template.</td>
</tr>
<tr>
<td>vdcRecomposeVapp</td>
<td>Recompose a vApp.</td>
</tr>
<tr>
<td>vdcUpdateDisk</td>
<td>Update an independent disk.</td>
</tr>
<tr>
<td>vdcUpdateMedia</td>
<td>Update one or more properties of a media object.</td>
</tr>
<tr>
<td>vdcUpdateStorageProfiles</td>
<td>Update the storage profiles of a VDC.</td>
</tr>
<tr>
<td>vdcUpdateTemplate</td>
<td>Update one or more properties of a vApp template.</td>
</tr>
<tr>
<td>vdcUpdateVapp</td>
<td>Update any section of a vApp.</td>
</tr>
<tr>
<td>vdcUpdateVappNetworkSection</td>
<td>Update the NetworkSection of a vApp.</td>
</tr>
<tr>
<td>vdcUpdateVdc</td>
<td>Modify one or more properties of an organization VDC.</td>
</tr>
</tbody>
</table>

### Table 10-8. Blocking Task Operations for Network Tasks

<table>
<thead>
<tr>
<th>Operation Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>networkCreateExternalNetwork</td>
<td>Create an external network.</td>
</tr>
<tr>
<td>networkCreateFencePoolTypeNetworkPool</td>
<td>Create an isolation-backed FencePoolType network pool.</td>
</tr>
<tr>
<td>networkCreateNetworkPool</td>
<td>Create a network pool.</td>
</tr>
<tr>
<td>networkCreateOrgNetwork</td>
<td>Create an organization network.</td>
</tr>
<tr>
<td>networkCreateOrgVdcNetwork</td>
<td>Create an organization VDC network.</td>
</tr>
<tr>
<td>networkCreateVappNetwork</td>
<td>Create a vApp network.</td>
</tr>
<tr>
<td>networkDelete</td>
<td>Delete a network.</td>
</tr>
<tr>
<td>networkDeleteNetworkPool</td>
<td>Delete a network pool.</td>
</tr>
<tr>
<td>networkMergeNetworkPools</td>
<td>Merge network pools.</td>
</tr>
<tr>
<td>networkRepairNetworkPool</td>
<td>Repair a network pool.</td>
</tr>
<tr>
<td>networkReset</td>
<td>Reset any network.</td>
</tr>
<tr>
<td>networkResetOrgNetwork</td>
<td>Reset an organization network.</td>
</tr>
<tr>
<td>networkResetOrgVdcNetwork</td>
<td>Reset an organization VDC network.</td>
</tr>
<tr>
<td>networkSyncSyslogSettings</td>
<td>Synchronize syslog settings for a network.</td>
</tr>
<tr>
<td>networkUpdateIpsecVpnTunnelStatus</td>
<td>Update the status of a VPN tunnel.</td>
</tr>
<tr>
<td>networkUpdateNetwork</td>
<td>Modify one or more properties of a network object.</td>
</tr>
</tbody>
</table>
### Table 10-8. Blocking Task Operations for Network Tasks (Continued)

<table>
<thead>
<tr>
<th>Operation Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>networkUpdateNetworkPool</td>
<td>Modify one or more properties of a VlanPoolType network pool object.</td>
</tr>
<tr>
<td>networkUpdateVlanPool</td>
<td>Modify one or more properties of any network pool object.</td>
</tr>
</tbody>
</table>

### Table 10-9. Blocking Task Operations for System Tasks

<table>
<thead>
<tr>
<th>Operation Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>systemForceDisconnectVimserver</td>
<td>Force disconnection of a vCenter server.</td>
</tr>
<tr>
<td>systemForceReconnectVimserver</td>
<td>Force reconnection of a vCenter server.</td>
</tr>
<tr>
<td>systemPurgeAllStrandedItems</td>
<td>Purge all stranded items.</td>
</tr>
<tr>
<td>systemPurgeStrandedItem</td>
<td>Purge a specific stranded item.</td>
</tr>
<tr>
<td>systemRefreshStorageProfiles</td>
<td>Refresh vCenter storage profiles.</td>
</tr>
<tr>
<td>systemRefreshVimserver</td>
<td>Refresh a vCenter server.</td>
</tr>
<tr>
<td>systemRegisterLookupService</td>
<td>Register with the vCenter lookup service.</td>
</tr>
<tr>
<td>systemRegisterVimserver</td>
<td>Register a vCenter server.</td>
</tr>
<tr>
<td>systemUnregisterLookupService</td>
<td>Unregister with the vCenter lookup service.</td>
</tr>
<tr>
<td>systemUnregisterVimserver</td>
<td>Unregister a vCenter server.</td>
</tr>
<tr>
<td>systemUpdateAmqpCertificate</td>
<td>Update the system AMQP certificate.</td>
</tr>
<tr>
<td>systemUpdateAmqpTruststore</td>
<td>Update the system AMQP truststore.</td>
</tr>
<tr>
<td>systemUpdateLdapCertificate</td>
<td>Update the system LDAP certificate.</td>
</tr>
<tr>
<td>systemUpdateLdapKeystore</td>
<td>Update the system LDAP keystore.</td>
</tr>
<tr>
<td>systemUpdateLdapSspiKeytab</td>
<td>Update the system LDAP SSPI keytab.</td>
</tr>
<tr>
<td>systemUpdateVcTruststore</td>
<td>Update the vCenter truststore.</td>
</tr>
<tr>
<td>systemUpdateVimserver</td>
<td>Update a vCenter server.</td>
</tr>
</tbody>
</table>

### Table 10-10. Blocking Task Operations for Resource Creation Tasks

<table>
<thead>
<tr>
<th>Operation Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>rclCreateProviderVdc</td>
<td>Create a provider VDC.</td>
</tr>
<tr>
<td>rclDeleteProviderVdc</td>
<td>Delete a provider VDC.</td>
</tr>
<tr>
<td>rclDisableHost</td>
<td>Disable a host.</td>
</tr>
<tr>
<td>rclEnableHost</td>
<td>Enable a host.</td>
</tr>
<tr>
<td>rclEnableVxlanForProviderVdc</td>
<td>Enable a VXLAN pool associated with a new Provider VDC.</td>
</tr>
<tr>
<td>rclMergePvdc</td>
<td>Merge Provider VDCs.</td>
</tr>
<tr>
<td>rclPrepareHost</td>
<td>Prepare a host</td>
</tr>
<tr>
<td>rclRedeployAllVms</td>
<td>Redeploy all virtual machines on this host.</td>
</tr>
<tr>
<td>rclRepairHost</td>
<td>Repair a host</td>
</tr>
<tr>
<td>rclUnprepareHost</td>
<td>Unprepare a host</td>
</tr>
<tr>
<td>rclUpdateHost</td>
<td>Update host operating system.</td>
</tr>
<tr>
<td>rclUpdateProviderVdcStorageProfiles</td>
<td>Update provider VDC storage profiles in this cluster.</td>
</tr>
</tbody>
</table>
Table 10-10. Blocking Task Operations for Resource Creation Tasks (Continued)

<table>
<thead>
<tr>
<th>Operation Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>rclUpdateResourcePoolSet</td>
<td>Update resource pools in this cluster.</td>
</tr>
<tr>
<td>rclUpgradeHostAgent</td>
<td>Upgrade the host agent on a host.</td>
</tr>
</tbody>
</table>

Table 10-11. Blocking Task Operations for Miscellaneous Tasks

<table>
<thead>
<tr>
<th>Operation Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>commonPurgeDeletedItem</td>
<td>Purge a deleted item.</td>
</tr>
<tr>
<td>importIntoExistingVapp</td>
<td>Import a virtual machine from vCenter to an existing vApp.</td>
</tr>
<tr>
<td>importLocalizationResource</td>
<td>Import external localization resources.</td>
</tr>
<tr>
<td>importMedia</td>
<td>Import a media object from vSphere.</td>
</tr>
<tr>
<td>importSingletonTemplate</td>
<td>Import a virtual machine from vCenter as a vApp template.</td>
</tr>
<tr>
<td>importSingletonVapp</td>
<td>Import a virtual machine from vCenter as a vApp.</td>
</tr>
<tr>
<td>catalogDelete</td>
<td>Delete a catalog.</td>
</tr>
<tr>
<td>catalogDeleteBacking</td>
<td>Delete the file(s) representing a catalog item.</td>
</tr>
<tr>
<td>catalogItemEnableDownload</td>
<td>Enable a catalog item for download</td>
</tr>
<tr>
<td>catalogSync</td>
<td>Update a subscribed catalog item from its external source.</td>
</tr>
<tr>
<td>catalogSyncAll</td>
<td>Update an externally subscribed catalog from its source.</td>
</tr>
<tr>
<td>metadataDelete</td>
<td>Delete a metadata item.</td>
</tr>
<tr>
<td>metadataUpdate</td>
<td>Update a metadata item.</td>
</tr>
<tr>
<td>vdcUploadMedia</td>
<td>Upload media.</td>
</tr>
<tr>
<td>vdcUploadOvfContents</td>
<td>Upload other OVF package contents</td>
</tr>
<tr>
<td>vdcUploadOvfDescriptor</td>
<td>Upload an OVF descriptor</td>
</tr>
</tbody>
</table>

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 338), 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 350. 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 351.
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 351.

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>

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 349.

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 339.

The following actions are allowed:

resume Unblock the task and allow it to continue.
abort End the task, cleaning up any transient objects that it created. Task status is set to ABORTED.
fail

End the task, setting the status of any transient objects that it created to ERROR. Task status is set to ERROR.

updateProgress

Reset the timeout value and timeout action for an active task. Use this action to keep the task alive when it might become subject to a timeout action.

Prerequisites

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

Procedure

1. Retrieve the list of active blocking tasks.

   See “Monitor Blocking Tasks,” on page 349. 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 351.

3. Make a request.

<table>
<thead>
<tr>
<th>Action</th>
<th>Request</th>
</tr>
</thead>
<tbody>
<tr>
<td>resume</td>
<td>POST a BlockingTaskOperationParams element to the Link where rel=&quot;resume&quot;</td>
</tr>
<tr>
<td>abort</td>
<td>POST a BlockingTaskOperationParams element to the Link where rel=&quot;abort&quot;</td>
</tr>
<tr>
<td>fail</td>
<td>POST a BlockingTaskOperationParams element to the Link where rel=&quot;fail&quot;</td>
</tr>
<tr>
<td>updateProgress</td>
<td>POST a BlockingTaskUpdateProgressParams element to the Link where rel=&quot;updateProgress&quot;</td>
</tr>
</tbody>
</table>

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 339 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"
   createdTime="2011-05-07T16:25:18.857+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">
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 349. 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 351.

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 353.

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 351 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 Extension Services Framework Requests,” on page 356
- “Register an Extension Service,” on page 358
- “Service-Specific Links,” on page 362
- “Service-Specific Tasks and Events,” on page 365
- “Localization Framework for Extension Services,” on page 376
- “REST APIs for Extension Services,” on page 379
- “Extension Service AMQP Message Format,” on page 382

Summary of vCloud API Extension Services Framework Requests

vCloud API extension services framework requests extend the vCloud API to allow you to extend existing vCloud Director services and integrate new services with 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

<table>
<thead>
<tr>
<th>Operation</th>
<th>Request</th>
<th>Request Body</th>
<th>Response</th>
</tr>
</thead>
<tbody>
<tr>
<td>Register an extension service with vCloud Director</td>
<td>POST API-URL/admin/extension/service</td>
<td>Service</td>
<td>Service</td>
</tr>
<tr>
<td>Retrieve a list of registered extension services.</td>
<td>GET API-URL/admin/extension/service</td>
<td>None</td>
<td>ExtensionServices</td>
</tr>
<tr>
<td>Update a registered extension service.</td>
<td>PUT API-URL/admin/extension/service/id</td>
<td>Service</td>
<td>Service</td>
</tr>
<tr>
<td>Delete a registered extension service.</td>
<td>DELETE API-URL/admin/extension/service/id</td>
<td>None</td>
<td>204 No Content</td>
</tr>
<tr>
<td>Register extension service links with vCloud Director objects.</td>
<td>POST API-URL/admin/extension/service/id/links</td>
<td>ServiceLink</td>
<td>ServiceLink</td>
</tr>
<tr>
<td>Create resource class definitions for a registered extension service.</td>
<td>POST API-URL/admin/extension/service/id/resourceClasses</td>
<td>ResourceClass</td>
<td>ResourceClass</td>
</tr>
<tr>
<td>Associate actions with a resource class.</td>
<td>POST API-URL/admin/extension/service/resourceClass/id/resourceClassActions</td>
<td>ResourceClassAction</td>
<td>ResourceClassAction</td>
</tr>
<tr>
<td>Create an ACL rule for a resource class.</td>
<td>POST API-URL/admin/extension/service/resourceClassAction/id/aclRules</td>
<td>AclRule</td>
<td>AclRule</td>
</tr>
<tr>
<td>Associate a right with an extension service.</td>
<td>POST API-URL/admin/extension/service/id/rights</td>
<td>Right</td>
<td>Right</td>
</tr>
<tr>
<td>Update a right associated with an extension service.</td>
<td>PUT API-URL/admin/right/id</td>
<td>Right</td>
<td>Right</td>
</tr>
<tr>
<td>Delete rights no longer associated with a role or ACL rule after the associated extension service has been deleted.</td>
<td>POST API-URL/admin/extension/service/action/clearUnusedRights</td>
<td>None</td>
<td>204 No Content</td>
</tr>
<tr>
<td>List the access rights for a specific combination of user and entity</td>
<td>POST API-URL/admin/user/id/entityRights</td>
<td>EntityReferences</td>
<td>UserEntityRights</td>
</tr>
<tr>
<td>Create a Task object in the organization/id</td>
<td>POST API-URL/tasksList/id</td>
<td>Task</td>
<td>Task</td>
</tr>
<tr>
<td>Update a Task object</td>
<td>PUT API-URL/task/id</td>
<td>Task</td>
<td>Task</td>
</tr>
</tbody>
</table>
### Table 11-1. Summary of vCloud API Extensibility Requests (Continued)

<table>
<thead>
<tr>
<th>Operation</th>
<th>Request</th>
<th>Request Body</th>
<th>Response</th>
</tr>
</thead>
<tbody>
<tr>
<td>Post an extension service event to the system.</td>
<td>POST API-URL/admin/org/id/events</td>
<td>Event</td>
<td>204 No Content</td>
</tr>
<tr>
<td>Initiate the upload of a localization bundle for an extension service.</td>
<td>POST API-URL/admin/extension/service/id/localizationBundles</td>
<td>BundleUploadParams</td>
<td>BundleUploadSocket</td>
</tr>
<tr>
<td>Create API definitions for a registered extension service.</td>
<td>POST API-URL/admin/extension/service/id/definitions</td>
<td>ApiDefinition</td>
<td>ApiDefinition</td>
</tr>
<tr>
<td>Create file descriptors for a registered extension service.</td>
<td>POST API-URL/admin/extension/service/definition/id/files</td>
<td>FileDescriptor</td>
<td>FileDescriptor</td>
</tr>
</tbody>
</table>

### 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 with vCloud Director 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, `com.example.service.backup`) are more likely to be unique. If a service tries to register a namespace that is already registered with this vCloud Director installation, 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 direct.

- **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 `java.util.regex.Pattern` recognizes. See “Create an API Filter for an Extension Service,” on page 380.

Registration can also specify the following optional properties:

- Definitions of Link elements that the service adds to the representations of vCloud API objects. See “Service-Specific Links,” on page 362.


- Locations of schema files if the service provides its own API. See “REST APIs for Extension Services,” on page 379

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 359.

4. POST the Service element to the URL described in Step 2.
   See the request portion of “Example: Register an Extension Service,” on page 359.

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 “Service-Specific Links,” on page 362.

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: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"
 ... >
<vmcloud:Link
 rel="remove"
 href="https://vcloud.example.com/api/admin/extension/service/45" />
<vmcloud:Link
 rel="edit"
 type="application/vnd.vmware.admin.service+xml"
 href="https://vcloud.example.com/api/admin/extension/service/45" />
<vmcloud:Link
 rel="rights"
 type="application/vnd.vmware.admin.rights+xml"
 href="https://vcloud.example.com/api/admin/extension/service/45/rights" />
<vmcloud:Link
 rel="down:serviceLinks"
 type="application/vnd.vmware.vcloud.query.records+xml"
 href="https://vcloud.example.com/api/admin/extension/service/45/links" />
<vmcloud:Link
 rel="bundle:upload"
 type="application/vnd.vmware.admin.bundleUploadParams+xml"
 href="https://vcloud.example.com/api/admin/extension/service/45/localizationbundles" />
<vmcloud:Link

vCloud API Programming Guide
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:

```xml
<vcloud:Link
  rel="down:serviceLinks"
  type="application/vnd.vmware.vcloud.query.records+xml"
  href="https://vcloud.example.com/api/admin/extension/service/45/links" />
```
Service-Specific 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-specific 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 new ones.

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 href attribute of the Link. This can be any URI, and can include the variables {baseUrl} and {resourceId}. When constructing the href value of the Link, vCloud Director replaces {baseUrl} with the vCloud Director REST API base URL, and replaces {resourceId} with the UUID portion of the id attribute value of the resource in which the Link is inserted. The following example might expand to the string https://vcloud.example.com/org/17.

```
<LinkHref>
    {baseUrl}/org/{resourceId}
</LinkHref>
```

**MimeType**

The value, specified as a MIME content type, of the type attribute of the Link.

**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.

**ResourceType**

The object type, specified as a MIME content type, of the object in which the Link appears.

**NOTE** You can constrain the Link to appear in a specific resource by including aResourceId 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 359:

   ```
   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:
   
   ```xml
   <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 363.

**Example: Add a Service Link**

This request adds a ServiceLink to the service created in “Example: Register an Extension Service,” on page 359

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 359:

   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
<?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}/backups"
        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"
```
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 376.

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+xml" 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"
  operation="Backup in progress for virtual machine with id 7b91b053-2b..."
  operationName="backupInProgress"
  name="task">
  <Owner type="application/vnd.vmware.vcloud.org+xml"
    name="Finance"
    href="https://vcloud.example.com/api/org/26"
    id="26" />
  <User type="application/vnd.vmware.admin.user+xml"
    name="bob"
    href="https://vcloud.example.com/api/admin/user/39" />
  <Progress>10</Progress>
</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">
    <Owner
        type="application/vnd.vmware.vcloud.org+xml"
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.

```xml
<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 Link element that specifies the MIME type of the resource and includes an href (URL) that can be used to retrieve the resource. The rules include a MIME type, a URL pattern, and a template for creating an id 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.

**Service Resource**

A member of a resource class distinguished by a specific id. If an extension service needs to define a resource class action or an ACL rule that applies to a specific resource, the service must create it as a ServiceResource and give it a UUID or other unique identifier.
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 {baseUrl} and {resourceId}. When constructing the href value, vCloud Director replaces {baseUrl} 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 369 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(?<id>[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>
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. The UrlPattern can be any of the following URL forms:

- An explicit URL like /backup/restore/vm/27. A UrlPattern of this form defines an action for a resource that has a specific URL.
- A URL that contains a regular expression, like /backup/restore/vm[-,a-z,0-9]*. A UrlPattern of this form defines an action for any resource in the class that matches the regular expression.
- A URL that contains a service resource id, which is expressed as <id>. In a UrlPattern, the delimiters must be written as the XML entities &lt; and &gt; . The id can stand alone, as in /backup/restore/vm/&lt;id&gt;, or appear as part of a regular expression like /backup/restore/vm/? &lt;id&gt;[-,a-z,0-9]*. A UrlPattern of this form targets a specific ServiceResource, which must the service must define and register.

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" />

   
   For information about the contents of this element, see the request portion of “Example: Define an Action for a Resource Class,” on page 371.

4. POST the ResourceClassAction element to the URL described in Step 2.

Example: Define an Action for a Resource Class

This example defines a resource class action for a GET request to a UrlPattern that could match URLs including https://vcloud.example.com/api/backup/27 or https://vcloud.example.com/api/backup/vm-27-backup--2013-04-25T01:17:00.000Z.

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
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

<table>
<thead>
<tr>
<th>Container Element</th>
<th>Access</th>
<th>Comments</th>
</tr>
</thead>
<tbody>
<tr>
<td>ServiceResourceAccess</td>
<td>Shared</td>
<td>The action is authorized for all resources in this resource class</td>
</tr>
<tr>
<td></td>
<td>Entity</td>
<td>The action is authorized for the service resource referenced in the Entity element in this container.</td>
</tr>
<tr>
<td>OrganizationAccess</td>
<td>Shared</td>
<td>The action is authorized for all members of the organization that owns the resource.</td>
</tr>
<tr>
<td></td>
<td>Published</td>
<td>The action is authorized for all members of any organization in the cloud.</td>
</tr>
<tr>
<td></td>
<td>Entity</td>
<td>The action is authorized for members of the organization referenced in the Entity element in this container.</td>
</tr>
<tr>
<td>PrincipalAccess</td>
<td>Shared</td>
<td>The action is authorized for all users</td>
</tr>
<tr>
<td></td>
<td>Entity</td>
<td>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.</td>
</tr>
</tbody>
</table>

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 371:

   ```
   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 374 for information about the contents of this element.

4. POST the **AclRule** 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 371. 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
<?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

```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
  rel="remove"
  href="https://vcloud.example.com/api/admin/extension/service/resourceclassaction/268"/>
...</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 375.

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.
Localization Framework for Extension Services

Extension service developers can provide localized content for service-specific tasks and events 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 file in the bundle has a name that indicates the locale to which it applies, and contains an arbitrary number of key=value pairs, where the key is the value of an attribute of a service-specific task operation or event, and the value is a localized string to display in log messages posted by the service.

Upload or Update a Localization Bundle

Each service provides a link that an administrator can use to upload a new or modified localization bundle for the service.

A localization bundle is a file in ZIP format that contains one or more localized message files that your service uses. Each line in one of these files provides localized text that replaces the text that your service posts in the values of certain attributes for service-specific events and tasks. See “Message File Content,” on page 377.

Prerequisites

- Verify that you are logged in to the vCloud API as a system administrator.
- Create a localization bundle.

Procedure

1. Create a localization bundle.
2. Find the localizationbundles URL for your service.
   a. Retrieve the XML representation of the service.
   b. Examine the representation for a Link of the following form:

   ```xml
   <vcloud:Link
       rel="bundle:upload"
       type="application/vnd.vmware.admin.bundleUploadParams+xml"
       href="https://vcloud.example.com/api/admin/extension/service/45/localizationbundles" />
   ```

   The localizationbundles URL is the value of the href attribute of this link.
3. Create a BundleUploadParams element that specifies the size of the bundle and the service namespace of the service.

4. POST the BundleUploadParams element to the localizationbundles URL of your service.

**Example: Upload a Localization Bundle**

This example uploads a localization bundle for the service created in “Example: Register an Extension Service,” on page 359. The initial request specifies the size of the ZIP file in bytes and the name of the service namespace.

Request:

```plaintext
POST https://vcloud.example.com/api/admin/extension/service/45/localizationbundles
Content-type: application/vnd.vmware.admin.bundleUploadParams+xml
...
<?xml version="1.0" encoding="UTF-8"?>
<vmext:BundleUploadParams
  xmlns:vmext="http://www.vmware.com/vcloud/extension/v1.5"
  xmlns:vcloud="http://www.vmware.com/vcloud/v1.5"
  fileSize="537"
  serviceNamespace="org.example.vcd.backup">
</vmext:BundleUploadParams>
```

The response body includes an uploadLocation URL. You can use a procedure similar to the one in “Example: Uploading File Data,” on page 68 to upload the ZIP file to this location.

Response:

```plaintext
<vmext:BundleUploadSocket
  uploadLocation="https://vcloud.example.com/transfer/6f3b.../file"
  xmlns:vmext="http://www.vmware.com/vcloud/extension/v1.5"
  xmlns:vcloud="http://www.vmware.com/vcloud/v1.5">`
</vmext:BundleUploadSocket>
```

To update a bundle, follow the same procedure using a new bundle that contains updated message files. When you upload a new localization bundle for a service that already has one, new key=value pairs are added, and the values of existing keys are updated.

**Message File Content**

Message files provide localized text for certain attribute values in Task and Event objects.

Each file in a localization bundle can include an arbitrary number of lines. Each line must have the following form, where key is any of the values that the service might assign to one of its service-specific Task or Event elements and value is the localized string to display.

```
key=value
```

The value string can contain parameters that provide the resourceId, resourceName, and resourceType of the subject of the Task or Event.

The following restrictions apply to each message file in a localization bundle:

- File contents must be encoded in the UTF-8 character set.
- key length cannot exceed 2000 UTF-8 characters.
- value length cannot exceed 2000 UTF-8 characters.
- File size cannot exceed 10MB.
- File name must be the locale code for the language used in the value strings. For example, the file containing English text must be named en_US. The file containing French text must be named fr_FR.
Whenever a localizable attribute appears in a log message, vCloud Director takes the following steps to find text to display:

1. If the service has a localization bundle, open the file in that bundle whose name corresponds to the current client locale and display the value as it appears in the file.

2. If the service has a localization bundle but no file exists in that bundle whose name corresponds to the current client locale, open the file in that bundle named `en_US` and display the value as it appears in the file. The default locale for vCloud Director is `en_US`.

3. Otherwise, display a predefined string.

A Task or Event that a service posts can also include a passthrough `key=value` pair that is always displayed as posted by the service, regardless of the current client locale or the presence or absence of a localization bundle.

**Message File Keys and Parameters**

Message file contents (`key=value` pairs) apply to all Task and Event objects that the service creates. If a Task and an Event both use the same `key`, the same message appears for both.

### Table 11-3. Localization Keys for Service-Specific Tasks

<table>
<thead>
<tr>
<th>Attribute Values Matched for <code>key</code></th>
<th>Available Parameters</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>operationName</code></td>
<td><code>resourceId</code></td>
</tr>
<tr>
<td></td>
<td>The value of the <code>id</code> attribute of the Owner of the Task.</td>
</tr>
<tr>
<td></td>
<td><code>resourceName</code></td>
</tr>
<tr>
<td></td>
<td>The value of the <code>name</code> attribute of the Owner of the Task.</td>
</tr>
<tr>
<td></td>
<td><code>resourceType</code></td>
</tr>
<tr>
<td></td>
<td>The value of the <code>type</code> attribute of the Owner of the Task.</td>
</tr>
</tbody>
</table>

- `Owner:type` None
- `serviceNamespace` None
- `operation` (passthrough) None

To create a message that appears only when a Task has status="running", append the string `_PROGRESS` to the `key`.

### Table 11-4. Localization Keys for Service-Specific Events

<table>
<thead>
<tr>
<th>Attribute Values Matched for <code>key</code></th>
<th>Available Parameters</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>type</code></td>
<td><code>resourceId</code></td>
</tr>
<tr>
<td></td>
<td>The value of the <code>id</code> attribute of the Owner of the Event.</td>
</tr>
<tr>
<td></td>
<td><code>resourceName</code></td>
</tr>
<tr>
<td></td>
<td>The value of the <code>name</code> attribute of the Owner of the Event.</td>
</tr>
<tr>
<td></td>
<td><code>resourceType</code></td>
</tr>
<tr>
<td></td>
<td>The value of the <code>type</code> attribute of the Owner of the Event.</td>
</tr>
</tbody>
</table>

- `Owner:type` None
- `serviceNamespace` None
- `typeFull` (passthrough) None

To create a message that appears only for a failed Event (one where `success="false"`), prepend the string `FAILED` to the `key`.

---

**vCloud API Programming Guide**

378 VMware, Inc.
**Example: Example Message File Content**

The following lines are appropriate in the en_US message file for a service that meets the following conditions:

- Its Namespace is registered as org.example.vcd.backup.
- It defined a Task whose operationName attribute can have a value of backupInProgress.
- It defined an Event whose type attribute can have a value of backupComplete.

```
org.example.vcd.backup=vCloud Backup Service
backupInProgress=Backup in progress for ${resourceName} ({resourceType}) with id: {resourceId}
backupInProgress_PROGRESS=Backup in progress for ${resourceName} ({resourceType}) with id: {resourceId}
backupComplete=Backup complete for entity {resourceName} ({resourceType}) with id: {resourceId}
FAILED.backupComplete=Backup failed for entity {resourceName} ({resourceType}) with id: {resourceId}
```

If the localization bundle for this service contained a file named fr_FR that included the following line, the Task posted in “Example: Add a Task to an Organization’s Tasks List,” on page 366 returns this localized value for the operationName attribute when the client locale is set to fr_FR. The passthrough value for operation is not localized.

```
backupInProgress_PROGRESS=Sauvegarde en cours pour entity {resourceName} ({resourceType}) avec id: {resourceId}
```

Request:

GET https://vcloud.example.com/api/task/604

Response:

200 OK
Content-Type: application/vnd.vmware.vcloud.task+xml
...
<Task
...
operation="Backup in progress for virtual machine with id 7b91b053-2b..."
operationName="Sauvegarde en cours pour entity Finance (application/vnd.vmware.vcloud.org+xml) avec id 26 "
...
>

**REST APIs for Extension Services**

A simple extension service does not need a REST API. You can define a service-specific REST API entrypoint 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 \texttt{UrlPattern} are sent to the service that has registered the filter. An API filter that specifies \texttt{ResponseContentType} is applied only to responses whose \texttt{Content-type} attribute has a value that matches the specified \texttt{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 359:
   
   \texttt{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:
   
   \texttt{<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 380 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 \texttt{UrlPattern} to set of patterns defined in the request portion of “Example: Register an Extension Service,” on page 359. The request replaces the existing set of filter expressions with a new one that includes the original set and one additional expression.

Request:

\texttt{POST https://vcloud.example.com/api/admin/extension/service/45/apifilters}
\texttt{Content-type:application/vnd.vmware.admin.apiFilter+xml}
\texttt{<?xml version="1.0" encoding="UTF-8"?>}
\texttt{<vmext:ApiFilter>}
\texttt{   <vmext:UrlPattern>(/api/org/.*/backups)|(/api/vApp/vapp-.*/backups)|
(\api/vApp/vapp-.*/action/backup)|(/api/backup/.*)) |
(\api/vApp/vapp-.*/action/recoverBackup)<vmext:UrlPattern>}
\texttt{</vmext:ApiFilter>
Response:
201 Created
Content-Type: application/vnd.vmware.admin.resourceClass+xml

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 \{baseUri\}, which represents the vCloud Director REST API base URL.

**Namespace**
The service namespace. See “Register an Extension Service,” on page 358

**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 382.

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 entrypoint 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>

Extension Service AMQP Message Format

vCloud Director uses the system AMQP service to communicate with extension services. Messages to and from an extension service are formatted as JSON objects.

When it receives an extension request, the vCloud Director REST service creates a message and sends it to the system AMQP service, specifying the exchange and routing key registered by the extension service. The extension service retrieves the message from a queue bound to the exchange it registered, processes the request, and returns a response to the common reply exchange.

AMQP Message Headers

Each message from vCloud Director to a service includes both standard and custom AMQP headers.
Table 11-5. Extension Service AMQP Message Headers

<table>
<thead>
<tr>
<th>Header</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>correlationId</td>
<td>A standard AMQP header that provides a unique identifier for the message. The extension must supply the same correlationId in the corresponding response.</td>
</tr>
<tr>
<td>reply-to</td>
<td>A standard AMQP header specifying the value that the extension must use as the routingKey in the response.</td>
</tr>
<tr>
<td>messageType</td>
<td>A custom AMQP header. One of:</td>
</tr>
<tr>
<td></td>
<td>ProcessHttpRequest</td>
</tr>
<tr>
<td></td>
<td>Indicates that this message is a forwarded request.</td>
</tr>
<tr>
<td></td>
<td>ProcessHttpResponse</td>
</tr>
<tr>
<td></td>
<td>Indicates that this message is a forwarded response.</td>
</tr>
<tr>
<td>replyToExchange</td>
<td>A custom AMQP header. The name of the AMQP exchange to which the extension should publish its response.</td>
</tr>
</tbody>
</table>

Property Names and Values

A request message contains all of the following name=value pairs.

Table 11-6. Extension Service AMQP Request Message Property Names and Values

<table>
<thead>
<tr>
<th>Name</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>method</td>
<td>The HTTP method (GET, PUT, POST, DELETE) used to make the request</td>
</tr>
<tr>
<td>id</td>
<td>The unique id of this message</td>
</tr>
<tr>
<td>scheme</td>
<td>The scheme (HTTP or HTTPS) specified in the request URL</td>
</tr>
<tr>
<td>protocol</td>
<td>The protocol used to make the request</td>
</tr>
<tr>
<td>headers</td>
<td>The request headers represented as a map of name:value pairs encoded as a JSON object in the form: name:value, name:value,...</td>
</tr>
<tr>
<td>queryString</td>
<td>The entire query string, or null if the request did not include a query string.</td>
</tr>
<tr>
<td>localPort</td>
<td>The local port to which the request was sent</td>
</tr>
<tr>
<td>remoteAddr</td>
<td>The IP address of the requesting machine</td>
</tr>
<tr>
<td>remotePort</td>
<td>The remote port from which the request was sent</td>
</tr>
<tr>
<td>localAddr</td>
<td>The IP address to which the request was sent</td>
</tr>
<tr>
<td>request</td>
<td>Always true in request messages</td>
</tr>
<tr>
<td>requestURI</td>
<td>The request URL, without any query string it might have included</td>
</tr>
<tr>
<td>parameters</td>
<td>always null</td>
</tr>
<tr>
<td>user</td>
<td>The id of the vCloud Director user who made the request</td>
</tr>
<tr>
<td>org</td>
<td>The id of the vCloud Director organization to which the requesting user belongs</td>
</tr>
<tr>
<td>rights</td>
<td>A comma-separated list of id values for the vCloud Director rights assigned to the requesting user.</td>
</tr>
</tbody>
</table>

The parameters, user, org, and rights properties provide the security context for the request, and are formatted as a separate JSON object, as shown in “Example: AMQP Message Format,” on page 384
A response message contains all of the following name=value pairs.

### Table 11-7. Extension Service AMQP Response Message Property Names and Values

<table>
<thead>
<tr>
<th>Name</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>id</td>
<td>The unique id of this message</td>
</tr>
<tr>
<td>headers</td>
<td>A comma-separated list of request headers in the form: name=value,name=value,...</td>
</tr>
<tr>
<td>statusCode</td>
<td>The HTTP status code to return to the requester</td>
</tr>
<tr>
<td>body</td>
<td>A base64-encoded response body</td>
</tr>
<tr>
<td>request</td>
<td>Always false in response messages</td>
</tr>
</tbody>
</table>

**Example: AMQP Message Format**

Assume an extension service that includes an API filter of the following form:

```xml
<vmext:ApiFilter>
  <vmext:UrlPattern>/api/org/.*</vmext:UrlPattern>
</vmext:ApiFilter>
```

When vCloud Director receives a request like this one:

```plaintext
GET https://10.23.6.168:8443/api/org/a93c9db9-7471-3192-8d09-a8f7eeda85f9
```

it creates the following message and places it on the service’s exchange.

```json
[
  {
    "method":"GET",
    "id":"32d5b9ec-5ee5-4aa3-9375-b054018b0e30",
    "scheme":"https",
    "protocol":"HTTP/1.1",
    "headers":{"Cookie":"...", "User-Agent":"...", ...},
    "queryString":null,
    "localPort":8443,
    "remoteAddr":"10.23.6.168",
    "remotePort":60576,
    "localAddr":"10.100.1.40",
    "request":true,
    "requestUri":"/api/org/a93c9db9-7471-3192-8d09-a8f7eeda85f9"
  },
  {
    "parameters":null,
    "user":"urn:vcloud:user:8cdd352f-f831-4712-a1a3-9e061687c5c6",
    "org":"urn:vcloud:org:a93c9db9-7471-3192-8d09-a8f7eeda85f9",
    "rights":null
  }
]
```

The service returns a response containing a base64-encoded body.

```json
[
  {
    "id":"32d5b9ec-5ee5-4aa3-9375-b054018b0e30",
    "headers":{"Date":"...", "Content-Type":"application/vnd.vmware.vcloud.org+xml;version=2.0"},
    "statusCode":200,
  }
]
"body":"base64-encoded-body",
"request":false,
}
]
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. 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.

**XML Schemas**

Each vCloud API object is defined in an XML schema document. Schema files and reference information about all elements, types, operations, and queries is included in the vCloud API Schema Reference. See “About the Schema Reference,” on page 25.

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 389.

All vCloud API requests are processed in the `http://www.vmware.com/vcloud/v1.5` XML namespace. vCloud API XML namespace information appears in the values of the `xsi:schemaLocation` and `xmlns` attributes in a response document.

```xml
xmlns="http://www.vmware.com/vcloud/v1.5"
xsi:schemaLocation="https://vcloud.example.com/api/v1.5/schema/master.xsd"
```

Other XML namespace identifiers may also be required in request bodies. See “XML Namespace Identifiers,” on page 389.

**API Versions**

The vCloud XML namespace (`http://www.vmware.com/vcloud/v1.5`) defines elements and attributes for all supported versions of the vCloud API. 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 5.6 and a version 5.6 response is expected:

```
Accept: application/*+xml;version=5.6
```
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. In general, client requests can access objects defined by any version of the vCloud API that is less than or equal to the version specified in the `Accept` header. Exceptions to this rule are mentioned in the vCloud Director Release Notes. 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 25.

To discover the API versions that a server supports, a client can make an unauthenticated GET request to a well-known URL on the server:

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

See “Example: Retrieve the Login URL and List of Supported API Versions,” on page 23.

## 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 and the `Description` and `ComputerName` elements have length limitations that depend on the object to which they are attached.

### Table 12-1. Length Limits on Element and Attribute String Values

<table>
<thead>
<tr>
<th>Object</th>
<th>Element or Attribute Name</th>
<th>Maximum Length in Characters</th>
</tr>
</thead>
<tbody>
<tr>
<td>Catalog</td>
<td>name</td>
<td>128</td>
</tr>
<tr>
<td>Catalog</td>
<td>Description</td>
<td>256</td>
</tr>
<tr>
<td>EdgeGateway</td>
<td>name</td>
<td>35</td>
</tr>
<tr>
<td>Media</td>
<td>name</td>
<td>128</td>
</tr>
<tr>
<td>Media</td>
<td>Description</td>
<td>256</td>
</tr>
<tr>
<td>VApp</td>
<td>name</td>
<td>128</td>
</tr>
<tr>
<td>VApp</td>
<td>Description</td>
<td>256</td>
</tr>
<tr>
<td>VAppTemplate</td>
<td>name</td>
<td>128</td>
</tr>
<tr>
<td>VAppTemplate</td>
<td>Description</td>
<td>256</td>
</tr>
<tr>
<td>Vdc</td>
<td>name</td>
<td>256</td>
</tr>
<tr>
<td>Vdc</td>
<td>Description</td>
<td>256</td>
</tr>
<tr>
<td>Vm</td>
<td>name</td>
<td>128</td>
</tr>
<tr>
<td>Vm</td>
<td>ComputerName</td>
<td>15 on Windows, 63 on all other platforms</td>
</tr>
</tbody>
</table>

## Extensibility

The vCloud API provides complete programmatic access to the vCloud Director Extension Services facility. See Chapter 11, “vCloud Director Extension Services,” on page 355.

In addition, there is a more general extensibility mechanism, `VCloudExtension`, that clients are free to use. `VCloudExtensibleType` 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 389
- “Common vCloud API Attributes,” on page 390
- “Retrieve an Object as an Entity,” on page 392

**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>
<thead>
<tr>
<th>Name</th>
<th>Value</th>
<th>Requirement</th>
</tr>
</thead>
<tbody>
<tr>
<td>xmlns</td>
<td><a href="http://www.vmware.com/vcloud/v1.5">http://www.vmware.com/vcloud/v1.5</a></td>
<td>Required in all request bodies.</td>
</tr>
<tr>
<td>xmlns:vmext</td>
<td><a href="http://www.vmware.com/vcloud/extension/v1.5">http://www.vmware.com/vcloud/extension/v1.5</a></td>
<td>Required in request bodies that include elements from the vSphere platform extensions.</td>
</tr>
<tr>
<td>xmlns:ovf</td>
<td><a href="http://schemas.dmtf.org/ovf/envelope/1">http://schemas.dmtf.org/ovf/envelope/1</a></td>
<td>Required in request bodies that include elements defined in OVF schema <a href="http://schemas.dmtf.org/ovf/envelope/1/dsp8023.xsd">http://schemas.dmtf.org/ovf/envelope/1/dsp8023.xsd</a>.</td>
</tr>
<tr>
<td>xmlns:rasd</td>
<td><a href="http://schemas.dmtf.org/wbem/wscim/1/cim-schema/2/CIM_ResourceAllocationSettingData">http://schemas.dmtf.org/wbem/wscim/1/cim-schema/2/CIM_ResourceAllocationSettingData</a></td>
<td>Required in request bodies that include elements defined in OVF schema CIM_ResourceAllocationSettingData.xsd.</td>
</tr>
<tr>
<td>xmlns:oe</td>
<td><a href="http://schemas.dmtf.org/ovf/environment/1">http://schemas.dmtf.org/ovf/environment/1</a></td>
<td>Required in request bodies that include elements defined in OVF schema dsp8027_1.1.0.xsd.</td>
</tr>
<tr>
<td>xsi:schemaLocation</td>
<td>An installation-dependent schema location search path. See <a href="http://www.w3.org/TR/xmlschema-0/">http://www.w3.org/TR/xmlschema-0/</a>.</td>
<td>Not required in request bodies.</td>
</tr>
</tbody>
</table>
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.

```xml
<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>
<thead>
<tr>
<th>Object Type</th>
<th>Name Scope</th>
</tr>
</thead>
<tbody>
<tr>
<td>ProviderVdc</td>
<td>Cloud</td>
</tr>
<tr>
<td>Org</td>
<td>Cloud</td>
</tr>
<tr>
<td>Vdc</td>
<td>Organization</td>
</tr>
<tr>
<td>Catalog</td>
<td>Organization</td>
</tr>
<tr>
<td>CatalogItem</td>
<td>Catalog</td>
</tr>
<tr>
<td>vAppTemplate</td>
<td>None</td>
</tr>
<tr>
<td>vApp</td>
<td>Organization</td>
</tr>
<tr>
<td>Vm</td>
<td>vApp</td>
</tr>
<tr>
<td>Media</td>
<td>Catalog</td>
</tr>
<tr>
<td>Disk</td>
<td>None</td>
</tr>
<tr>
<td>Network</td>
<td>Container (Organization VDC, vApp, or cloud)</td>
</tr>
</tbody>
</table>
Object Identifier, Type, and Reference

These attributes are common to all object representations.

**id**
The object identifier, expressed in URN format. The value of the id attribute uniquely identifies the object, persists for the life of the object, and is never reused. The id attribute value is intended to provide a context-free identifier that can be used with the vCloud API entityResolver (see “Retrieve an Object as an Entity,” on page 392).

**type**
The object type, specified as a MIME content type.

**href**
An object reference, expressed in URL format. This reference includes the object identifier portion of the id attribute value, and supplies additional information, including the current location of the object when accessed in a specific view. Although URLs have a well-known syntax and a well-understood interpretation, a client should treat each href as an opaque string. The rules that govern how the server constructs href strings might change in future releases.

Object Creation Status

Objects such as VAppTemplate, VApp, and Vm, that extend the ResourceEntity type have a status attribute whose value indicates the state of the object. In this table, YES indicates that a status value is allowed for the object listed in the column header. The status value for a VAppTemplate or VApp, which contain Vm objects that each have a status attribute of their own, is computed from the status of the contained objects. When returned in an XML representation, status has a numeric value. When returned by the query service, it has a string value.

<table>
<thead>
<tr>
<th>Numerical Value</th>
<th>String Value</th>
<th>Description</th>
<th>vAppTemplate</th>
<th>VApp</th>
<th>Vm</th>
<th>Media</th>
</tr>
</thead>
<tbody>
<tr>
<td>-1</td>
<td>FAILED_CREATION</td>
<td>The object could not be created.</td>
<td>YES</td>
<td>YES</td>
<td>YES</td>
<td>YES</td>
</tr>
<tr>
<td>0</td>
<td>UNRESOLVED</td>
<td>The object is unresolved.</td>
<td>YES</td>
<td>YES</td>
<td>YES</td>
<td>YES</td>
</tr>
<tr>
<td>1</td>
<td>RESOLVED</td>
<td>The object is resolved.</td>
<td>YES</td>
<td>YES</td>
<td>YES</td>
<td>YES</td>
</tr>
<tr>
<td>2</td>
<td>DEPLOYED</td>
<td>The object is deployed.</td>
<td>No</td>
<td>No</td>
<td>No</td>
<td>No</td>
</tr>
<tr>
<td>3</td>
<td>SUSPENDED</td>
<td>The object is suspended.</td>
<td>No</td>
<td>YES</td>
<td>YES</td>
<td>No</td>
</tr>
<tr>
<td>4</td>
<td>POWERED_ON</td>
<td>The object is powered on.</td>
<td>No</td>
<td>YES</td>
<td>YES</td>
<td>No</td>
</tr>
<tr>
<td>5</td>
<td>WAITING_FOR_INPUT</td>
<td>The object is waiting for user input.</td>
<td>No</td>
<td>YES</td>
<td>YES</td>
<td>No</td>
</tr>
<tr>
<td>6</td>
<td>UNKNOWN</td>
<td>The object is in an unknown state.</td>
<td>YES</td>
<td>YES</td>
<td>YES</td>
<td>No</td>
</tr>
<tr>
<td>7</td>
<td>UNRECOGNIZED</td>
<td>The object is in an unrecognized state.</td>
<td>YES</td>
<td>YES</td>
<td>YES</td>
<td>No</td>
</tr>
<tr>
<td>8</td>
<td>POWERED_OFF</td>
<td>The object is resolved and powered off.</td>
<td>YES</td>
<td>YES</td>
<td>YES</td>
<td>No</td>
</tr>
<tr>
<td>9</td>
<td>INCONSISTENT_STATE</td>
<td>The object is in an inconsistent state.</td>
<td>No</td>
<td>YES</td>
<td>YES</td>
<td>No</td>
</tr>
</tbody>
</table>
Table 12-4. status Attribute Values for VAppTemplate, VApp, Vm, and Media Objects (Continued)

<table>
<thead>
<tr>
<th>Numeric Value</th>
<th>String Value</th>
<th>Description</th>
<th>vAppTemplate</th>
<th>vApp</th>
<th>Vm</th>
<th>Media</th>
</tr>
</thead>
<tbody>
<tr>
<td>10</td>
<td>MIXED</td>
<td>Children do not all have the same status.</td>
<td>YES</td>
<td>YES</td>
<td>No</td>
<td>No</td>
</tr>
<tr>
<td>11</td>
<td>DESCRIPTOR.Pending</td>
<td>Upload initiated, OVF descriptor pending.</td>
<td>YES</td>
<td>No</td>
<td>No</td>
<td>No</td>
</tr>
<tr>
<td>12</td>
<td>COPYING_CONTENTS</td>
<td>Upload initiated, copying contents.</td>
<td>YES</td>
<td>No</td>
<td>No</td>
<td>No</td>
</tr>
<tr>
<td>13</td>
<td>DISK_CONTENTS_PENDING</td>
<td>Upload initiated, disk contents pending.</td>
<td>YES</td>
<td>No</td>
<td>No</td>
<td>No</td>
</tr>
<tr>
<td>14</td>
<td>QUARANTINED</td>
<td>Upload has been quarantined.</td>
<td>YES</td>
<td>No</td>
<td>No</td>
<td>No</td>
</tr>
<tr>
<td>15</td>
<td>QUARANTINE_EXPIRED</td>
<td>Upload quarantine period has expired.</td>
<td>YES</td>
<td>No</td>
<td>No</td>
<td>No</td>
</tr>
<tr>
<td>16</td>
<td>REJECTED</td>
<td>Upload has been rejected.</td>
<td>YES</td>
<td>No</td>
<td>No</td>
<td>No</td>
</tr>
<tr>
<td>17</td>
<td>TRANSFER_TIMEOUT</td>
<td>Upload transfer session timed out.</td>
<td>YES</td>
<td>No</td>
<td>No</td>
<td>YES</td>
</tr>
<tr>
<td>18</td>
<td>VAPP_UNDEPLOYED</td>
<td>The vApp is resolved and undeployed.</td>
<td>YES</td>
<td>No</td>
<td>No</td>
<td>No</td>
</tr>
<tr>
<td>19</td>
<td>VAPP_PARTIALLY_DEPLOYED</td>
<td>The vApp is resolved and partially deployed.</td>
<td>YES</td>
<td>No</td>
<td>No</td>
<td>No</td>
</tr>
</tbody>
</table>

VDC objects have their own set of status values and mappings.

Table 12-5. status Attribute Values for VDC Objects

<table>
<thead>
<tr>
<th>Numeric Value</th>
<th>String Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>-1</td>
<td>FAILED_CREATION</td>
<td>The VDC could not be created.</td>
</tr>
<tr>
<td>0</td>
<td>NOT_READY</td>
<td>The VDC is not ready for use</td>
</tr>
<tr>
<td>1</td>
<td>READY</td>
<td>The VDC is ready for use</td>
</tr>
<tr>
<td>2</td>
<td>UNKNOWN</td>
<td>The VDC status could not be retrieved</td>
</tr>
<tr>
<td>3</td>
<td>UNRECOGNIZED</td>
<td>The VDC status cannot be mapped to a known state.</td>
</tr>
</tbody>
</table>

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 entity Resolver URL to retrieve a context-free representation of the underlying object as an Entity element. The Entity includes a Link element for each currently valid reference to the object identified by the id specified in the request.

Prerequisites

Verify that you are logged in to the vCloud API.
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, which includes a link to the entityResolver.
   
   ```xml
   <Session ... >
   ...
   <link
      rel="entityResolver"
      type="application/vnd.vmware.vcloud.entity+xml"
      href="https://vcloud.example.com/api/entity/"/>
   </Session>
   ```

2. Retrieve the object whose id you want to resolve and find the value of its id attribute.
   
   See the request portion of “Example: Using the entityResolver URL,” on page 393.

3. Append the value of the object's id attribute to the entityResolver URL.

4. Make a GET request to the URL you created in Step 3.
   
   See the request portion of “Example: Using the entityResolver URL,” on page 393.

Example: Using the entityResolver URL

This example retrieves the organization object shown in “Example: Object id, type, and href Attributes,” on page 12 as an Entity.

Request:

GET https://vcloud.example.com/api/entity/urn:vcloud:org:72fe715c-5f6a-407f-bbb2-bf465915b5f4

This response includes two Link elements, each of which provides a valid href to the object identified by the id specified in the request.

Response:

```xml
<entity
   xmlns="http://www.vmware.com/vcloud/v1.5"
   id="urn:vcloud:org:72fe715c-5f6a-407f-bbb2-bf465915b5f4"
   name="urn:vcloud:org:72fe715c-5f6a-407f-bbb2-bf465915b5f4"
   type="application/vnd.vmware.vcloud.entity+xml"
   href="https://vcloud.example.com/api/entity/urn:vcloud:org:72fe715c-5f6a-407f-bbb2-bf465915b5f4"
   ...>
   ...
   <link
      rel="alternate"
      type="application/vnd.vmware.vcloud.organization+xml"
      href="https://vcloud.example.com/api/org/72fe715c-5f6a-407f-bbb2-bf465915b5f4"/>
   <link
      rel="alternate"
      type="application/vnd.vmware.vcloud.admin.organization+xml"
      href="https://vcloud.example.com/api/admin/org/72fe715c-5f6a-407f-bbb2-bf465915b5f4"/>
</entity>
```
Index

A
access control 86
access rights 88
administrative tasks, about 169
administrator, system 172
AMQP
  about 333
  certificate and truststore 301
AMQP settings
  to test 336
  to configure 334
API client, to develop 23
attributes
  custom 141
  name 390
  status 390

B
blocking task, to configure 333
blocking task requests, to monitor 349
blocking task settings, to configure 344
browsing 45

C
catalog
  change owner 85
  controlling access to 86
  for external publication 225
  removing items 81
  storage profile for 229
  subscription endpoint 236
  to change owner 169
  to delete 169
  to retrieve 31, 45
  to administer 221
  to create 169, 222
  to find 30
  to publish externally 235
  to share 232
  to share with specific organizations 233
  to synchronize with external source 58, 81, 227
  with external subscription 227
catalog item
  about 78
  to copy 78
to move 78
to change name or description 79
to copy or move 58
to remove 81
to retrieve 32, 45
to synchronize with external source 58, 81
client, REST 25
cloud, administrative view of 53
certificate and truststore 301
to test 336
to configure 334
download URL 73

datatools
  to delete 261
  to enable or disable 261
  to retrieve or update 261

dhcp service, in vApp network 148
dhcp service, in Edge Gateway 211
disk, independent 82

download URL 73

download URL 73

download URL 73

download URL 73

download URL 73

download URL 73

download URL 73

download URL 73

download URL 73

D
datastore
to delete 261
to enable or disable 261
to retrieve or update 261
DHCP 205, 211
dhcp service, in vApp network 148
DHCP service, in Edge Gateway 211
disk, independent 82
download URL 73

download URL 73

download URL 73

download URL 73

download URL 73

download URL 73

download URL 73

download URL 73

download URL 73

download URL 73

download URL 73

download URL 73

E
Edge Gateway
  about 193
  interface capacity 215
  syslog server settings 220
  to create 196
  to create or update 169
  entity, retrieve object as 12
  entity resolver, about 392
  event, service-specific 367
  examples, conventions for 25
  extension framework 355
  extension services
    API definition 381
    API filters for 380
    authorization frameworks for 368
    custom rights for 356
    links in 356
    localization bundles 376
    localization of 376
    message files 377
    message format 382
resource class actions 371
resource classes 369
REST APIs for 379
to register 356, 358
to retrieve list of 356
to unregister 356
URL patterns for 380
external network
to retrieve or update 261
to create 287

F
firewall service
and syslog 220
in vApp network 148
in Edge Gateway 203

G
group
to create, update, or remove 169
to import 247
to import from SAML 249
groups, to administer 243

H
host
to update or repair 261
to enable or disable 261

I
id attribute, and entity resolver 392
independent disk
about 82
list virtual machines using 58
to create 82
to attach or detach 94, 118
to create or update 58
to remove 84
instantiation parameters
in instantiateVAppTemplate request 96
in instantiateOvf request 104
sections allowed in 133
IP addresses, allocated to network 45
IP address, of virtual machine NIC 151

K
keytab, to upload 180, 301

L
LDAP, certificate and keystore 180, 301
Link element, rel attribute 13
load balancer service 208
logging, of firewall actions 203
login
integrated identity provider 47
SAML identity provider 49
login URL, to obtain 45
logout 44
M
maintenance mode, vApp 133
media
downloading 76
retrieve owner 85
supported formats 74
to insert or remove virtual 94
to copy, move, or delete 58
to upload 57
to upload or download 58
uploading 74
metadata
about 307
to retrieve or modify 309
to search for 329
metadata value, to retrieve or modify 312
N
NAT service
in Edge Gateway 205
in vApp network 148
network
list IP addresses allocated to 45
to create, update, or remove 169
to reset 169
to retrieve 45
network pool
and organization VDC network 193
isolation-backed 293
list of 272
portgroup-backed 295
to create 290
to retrieve or update 261
VLAN-backed 291
VxlanPoolType 277, 290, 293
network services
in vApp network 148
in Edge Gateway 201
list of 193
networks
about 193
DHCP service 211
NAT services 205
to retrieve list from vCenter 271
VPN 210
notifications
format of 338
to enable or disable 334
O
object identifiers 12
object references, about 12
object hierarchy, diagram of 10
organization
federation settings 182
retrieve or update settings 177
system 172
to create, update, or remove 169
to retrieve administrative view of 169
to administer 173
to create 173
to enable or disable 169, 179
to remove 179
organization VDC network
about 193
direct 212
isolated 218
routed 215
to administer 193
to configure static routes 206
to create 211
to create, update, or remove 169
organizations
federation of 177
to list 45, 52
OVF, to instantiate 104
OVF specification 91
OVF descriptor
to download 72
upload URL for 63
OVF environment 156
OVF package
manifest file 61, 67
to upload or download 57
uploading 60
OVF upload
initiating 61
to monitor progress of 68
using ranged PUTs 69

Q
queries
packaged 320
typed 316
query service
about 315
and metadata 329
attribute names 325
query parameters 325
query encoding rules 325
query types 316

R
RASD 91
requests
about 19
headers 19, 387
login 28
resource pool
adding 282
list of 261
removing 282
to enable or disable 261
to migrate virtual machines to or from 261
resource pool set, Provider VDC 281
resource pools, to retrieve list from vCenter 269
responses, about 21
right
reference format 257
to create 375
role, to create 251, 257
roles and rights 252

S
SAML
identity provider 49, 243
organization settings for 177
schema files, accessing 25
schema reference 25
SectionType element, to retrieve or update 144
service
DHCP 211
NAT 205
service links
required and optional elements 362
to add 362
to add or remove 362
to delete 363
Session object, to delete 44
single sign-on, SAML metadata 182
snapshot, of vApp 94
SSL certificate, to upload 180, 301
SSPI 180, 301

P
ProductSection element, to retrieve or update 94
Provider VDC
resource pool set 281
to retrieve or update 261
to add or remove storage profiles 284
to create 277
to merge 261, 286
static routing service
  in vApp network 148
  in Edge Gateway 206
status attribute
  of vApp or vApp template 60
  values 390
storage profile, disk 164
storage profiles
  and storage policies 190, 284
  compliance check for virtual machine 94
  to retrieve list from vCenter 275
  to refresh list of 275
system settings, to retrieve or update 265

T
  task
    blocking 350
    service-specific 365
    to cancel 169
    to retrieve 169
    update progress 352
    task list, to retrieve 169
    task operations 346
    timezone, in dateTime values 387
    truststore, to upload 180, 301

U
  user
    admin take ownership of a user's objects 169, 251
    list of rights granted 169
    to import from SAML 249
    to retrieve 169
    to update or remove 169
    to create 169, 244
    to import 246
users, to administer 243

V
  vApp
    access control for 116, 231
    add virtual machines 110
    capturing 77, 114
    changing owner 85
    cloning 112
    composing 107
    configuration links in 135
    controlling access to 86
    datacenter operations 91
    importing 77, 297
    importing from vCenter 296
    list of power operations 94
    maintenance mode 133
    recompose 110
remove virtual machines 110
  to change name or description 94
  to change owner 169
  to compose or recompose 94
  to deploy or undeploy 94
  to download 58
  to import from vCenter as template 261
  to import from vSphere 261
  to instantiate 96
  to undeploy 42
  to capture or clone 94
  to configure 133
  to create from OVF 94
  to delete 42
  to download as OVF package 70
  to enable for download 71
  to enter or exit maintenance mode 261
  to modify vApp network configuration 145
  to operate 132
  to power off 42
  to retrieve 38
  to view or modify lease settings 136
  to view or modify network settings 136
  to view or modify startup settings 136
  Vm snapshots 94
  vApp network
    about 193
    to retrieve 193
    to modify 145
    to view or modify configuration 136
  vApp snapshot 120
  vApp template
    captured from vApp 114
    hardware customization 98, 101
    retrieve owner 85
    to instantiate 35, 94
    to update 133
    to change name or description 58
    to copy, move, or delete 58
    to create from OVF 58
    to download 58
    to download as OVF package 70
    to enable for download 71
    to enable or disable for download 58
    to import virtual machine as 298
    to relocate virtual machine from 94
    to retrieve 45
    to upload or download 57
    upload URL 66
    uploading vmdk files 67
  vCenter server
    to attach 266
to register or unregister 261

to update settings 261

vCenter resources, to discover 268

vCloud API

and RESTful programming style 9

revision history 7

VDC

allocation models 185

InstantiateVAppTemplate action 34

networks in 34

SupportedHardwareVersion elements 185

to create, update, or remove 169

to retrieve 45

to add or remove storage profiles 190

to administer 185

to create 185

to enable or disable 169, 193

to find 30

to remove 193

virtual machine

available for import 273

CPU configuration 153

disk configuration 164

disk storage profile 164

disks 159

guest customization for 154

hard disk configuration 160

importing 297

importing from vCenter 296

metrics 121, 122, 126, 130

network cards 159

question from 117

to consolidate 94

to install VMware tools 94

to update NetworkConnectionSection 151

to update storage profile 163

to upgrade hardware version 94

to attach or detach independent disk 94

to import into existing vApp 261

to relocate 94, 299

to relocate to a different storage profile 163

to update multiple sections 142

to view or modify CPU properties 139

to view or modify guest customization properties 139

to view or modify memory settings 139

to view or modify network cards 139

to view or modify network connection 139

to view or modify operating system properties 139

to view or modify virtual disks 139

virtual hardware, vmx versions supported by

Provider VDC 277

virtual machine metrics

about 121

collection intervals 126, 130

negative values for 121

retrieve all 122

retrieve subset 126

wildcards 130

Vm

configuration links in 137

list of power operations 94

pending question 94

to retrieve 45

to reboot or reset 94

vmdk file, to download from template 73

VMware Tools

to install 94

to retrieve installed version 94

vSphere

operations 55

retrieve object URL 304

vSphere client, to generate URL for use by 261

vSphere object, retrieve URL to access directly 261

vSphere object map 305

vSphere platform, to manage 261

W

workflow, example of 27

X

XML

compressed responses 19

validation of 19

XML namespaces 389

XML schemas, reference information 387