Using VMware vRealize Orchestrator Plug-Ins

vRealize Orchestrator 7.0
You can find the most up-to-date technical documentation on the VMware website at:
https://docs.vmware.com/
If you have comments about this documentation, submit your feedback to
docfeedback@vmware.com
Contents

Using VMware vRealize Orchestrator Plug-Ins 9

1 Introduction to Orchestrator Plug-Ins 10
   Orchestrator Architecture 10
   Plug-Ins Installed with the Orchestrator Server 11
   Access the Orchestrator API Explorer 14

2 Configure the Orchestrator Plug-Ins 15
   Manage the Orchestrator Plug-Ins 15
   Uninstall a Plug-In 16
   Reinstall Plug-Ins 16

3 Using the vCenter Server Plug-In 18
   Configuring the vCenter Server Plug-In 18
      Configuration Workflows 19
      Configure the Connection to a vCenter Server Instance 19
   vCenter Server Plug-In Scripting API 20
   Using the vCenter Server Plug-In Inventory 21
   Using XPath Expressions with the vCenter Server Plug-In 21
      Using XPath Expressions with the vCenter Server Plug-In 21
   Access the vCenter Server Plug-In Workflow Library 22
   vCenter Server Plug-In Workflow Library 23
      Batch Workflows 25
      Cluster and Compute Resource Workflows 26
      Configuration Workflows 26
      Custom Attributes Workflows 27
      Datacenter Workflows 27
      Datastore and Files Workflows 28
      Datacenter Folder Management Workflows 28
      Host Folder Management Workflows 28
      Virtual Machine Folder Management Workflows 29
      Guest Operation Files Workflows 29
      Guest Operation Processes Workflows 29
      Power Host Management Workflows 30
      Basic Host Management Workflows 30
      Host Registration Management Workflows 31
      Networking Workflows 31
      Distributed Virtual Port Group Workflows 31
Run a Complete JDBC Cycle 54
Using the SQL Plug-In Standard Workflows 54
SQL Plug-In Workflow Library 55
Generate CRUD Workflows for a Table 55

7 Using the SSH Plug-In 57
Configure the SSH Plug-In 57
Configuration Workflows 58
Running the SSH Plug-In Sample Workflows 58
Generate a Key Pair 59
Change the Key Pair Passphrase 59
Register an Orchestrator Public Key on an SSH Host 60
Run an SSH Command 60
Copy a File from an SSH Host 61
Copy a File to an SSH Host 62

8 Using the XML Plug-In 63
Running the XML Plug-In Sample Workflows 63
Create a Simple XML Document 64
Find an Element in an XML Document 64
Modify an XML Document 65
Create an Example Address Book from XML 66

9 Using the Mail Plug-In 67
Define the Default SMTP Connection 67
Using the Mail Plug-In Sample Workflows 68
Access the Mail Plug-In Sample Workflows 68
Mail Plug-In Sample Workflows 68

10 Using the Net Plug-In 69

11 Using the Enumeration Plug-In 70
Time Zone Codes 70

12 Using the Workflow Documentation Plug-In 73
Workflow Library for the Workflow Documentation Plug-In 73
Generate Workflow Documentation 73

13 Using the HTTP-REST Plug-In 75
Configuring the HTTP-REST Plug-In 75
Configuration Workflows 75
Configure Kerberos Authentication 76
Add a REST Host 76
Add a REST Operation 78
Add a Schema to a REST Host 79
Generate a New Workflow from a REST Operation 80
Invoking a REST Operation 80
Invoke a REST Operation 81

14 Using the SOAP Plug-In 82
Configuring the SOAP Plug-In 82
Configuration Workflows 82
Add a SOAP Host 83
Configure Kerberos Authentication 84
Generate a New Workflow from a SOAP Operation 85
Test a Custom-Generated Workflow 85
Invoke a SOAP Operation 86

15 Using the AMQP Plug-In 87
Configuring the AMQP Plug-In 87
Configuration Workflows 87
Add a Broker 87
Subscribe to Queues 88
Update a Broker 89
Using the AMQP Plug-In Standard Workflows 89
Declare a Binding 90
Declare a Queue 90
Declare an Exchange 91
Send a Text Message 92
Delete a Binding 93

16 Using the SNMP Plug-In 94
Managing SNMP Devices 94
Device Management Workflows 94
Register an SNMP Device 95
Managing SNMP Queries 96
Query Management Workflows 96
Add a Query to an SNMP Device 96
Managing the SNMP Trap Host 97
Trap Host Management Workflows 97
Set the SNMP Trap Port 97
Receiving SNMP Traps 98
Wait for a Trap on an SNMP Device 98
Set a Trap Policy 98
Edit a Trap Policy 99
Generic SNMP Request Workflows 100

17 Using the Active Directory Plug-In 101
Configuring the Active Directory Plug-In 101
   Active Directory Configuration Workflows 101
Using the Active Directory Plug-In Workflow Library 102
Using the Active Directory Plug-In Inventory 102
Access the Active Directory Plug-In Workflow Library 102
Active Directory Plug-In Workflows 102

18 Using the Dynamic Types Plug-In 105
Dynamic Types Configuration Workflows 105

19 Using the PowerShell Plug-In 107
Using the PowerShell Plug-In Inventory 107
Configuring the PowerShell Plug-In 108
   Configuration Workflows 108
   Configure Kerberos Authentication 108
   Add a PowerShell Host 109
Running PowerShell Scripts 110
   Invoke a PowerShell Script 110
   Invoke an External Script 110
Generating Actions 111
   Generate an Action from a PowerShell Script 111
   Generate an Action for a PowerShell Cmdlet 112
Passing Invocation Results Between Actions 113
PowerCLI Integration with the PowerShell Plug-In 113
   Converter Workflows 114
Sample Workflows 114
Access the PowerShell Plug-In API 114
Working with PowerShell Results 115
Examples of Scripts for Common PowerShell Tasks 116
Troubleshooting 118
   Servers Not Found in Kerberos Database 118
   Unable to Obtain a Kerberos Ticket 118
   Kerberos Authentication Fails Due to Time Differences 119
   Kerberos Authentication Session Mode Fails 119
   Unable to Reach a Key Distribution Center for a Realm 119

20 Using the Multi-Node Plug-In 121
Introduction to the vRealize Orchestrator Multi-Node Plug-In 121
Configuring the Multi-Node Plug-In 121
  Servers Configuration Workflows 122
  Add an Orchestrator Server 122
Using Proxy Workflows 122
  Synchronous Proxy Workflows 123
  Asynchronous Proxy Workflows 123
  Remote Execution Workflows 124
Using the Multi-Node Plug-In Inventory 125
  Remote Management Workflows 125
Access the Multi-Node Plug-In API 126
Multi-Node Plug-In Use Cases 126
  Create a Multi-Proxy Action 127
  Maintenance of Remote and Proxy Workflows 128
  Deploy a Package from a Local Server 129

21 Using the vCloud Suite API (vAPI) Plug-In 130
  Configuring the vCloud Suite API Plug-In 130
  Import a vCloud Suite API Metamodel 130
  Add a vCloud Suite API Endpoint 131
  Access the vCloud Suite API Plug-In API 132
Using VMware vRealize Orchestrator Plug-Ins

*Using VMware vRealize Orchestrator Plug-Ins* provides information and instructions about configuring and using the standard set of plug-ins installed with VMware® vRealize Orchestrator.

**Intended Audience**

This information is intended for advanced vSphere administrators and experienced system administrators who are familiar with virtual machine technology and datacenter operations.
Introduction to Orchestrator Plug-Ins

With the Orchestrator plug-ins, you can access and control external technologies and applications. Exposing an external technology in an Orchestrator plug-in lets you incorporate objects and functions in workflows and run workflows on the objects of that external technology.

The external technologies that you access by using plug-ins include virtualization management tools, email systems, databases, directory services, and remote control interfaces.

Orchestrator provides a standard set of preinstalled plug-ins, which expose the VMware vCenter Server API, email and authentication capabilities, and other technologies. In addition, the Orchestrator open plug-in architecture lets you to develop plug-ins to access other applications. Orchestrator implements open standards to simplify integration with external systems. For information about developing custom content, see Developing with VMware vRealize Orchestrator.

The standard set of plug-ins is automatically installed with the Orchestrator server. You might need to configure some of the plug-ins, for example the vCenter Server plug-in, before start using them.

Plug-ins extend the Orchestrator scripting engine with new object types and methods, and plug-ins publish notification events from the external system that triggers events in Orchestrator and in the plugged-in technology. Plug-ins provide an inventory of JavaScript objects that you can access on the Inventory tab of the Orchestrator client. Each plug-in contains packages of workflows and actions that you can run on the objects in the inventory to automate the typical use cases of the integrated product.

This chapter includes the following topics:

- Orchestrator Architecture
- Plug-Ins Installed with the Orchestrator Server
- Access the Orchestrator API Explorer

Orchestrator Architecture

Orchestrator contains a workflow library and a workflow engine to allow you to create and run workflows that automate orchestration processes. You run workflows on the objects of different technologies that Orchestrator accesses through a series of plug-ins.

Orchestrator provides a standard set of plug-ins, including a plug-in for vCenter Server, to allow you to orchestrate tasks in the different environments that the plug-ins expose.
Orchestrator also presents an open architecture to allow you to plug in external third-party applications to the orchestration platform. You can run workflows on the objects of the plugged-in technologies that you define yourself. Orchestrator connects to an authentication provider to manage user accounts, and to a database to store information from the workflows that it runs. You can access Orchestrator, the Orchestrator workflows, and the objects it exposes through the Orchestrator client interface, or through Web services.

Figure 1.1. VMware vRealize Orchestrator Architecture

Plug-Ins Installed with the Orchestrator Server

Orchestrator includes a collection of standard plug-ins. Each plug-in exposes an external product API to the Orchestrator platform. Plug-ins provide inventory classes, additional object types for the scripting engine, and publish notification events from the external system. Each plug-in also provides a library of workflows for automating the typical use cases of the integrated external products.

You can see the list of the installed plug-ins on the Manage Plug-ins page in Control Center. For the plug-ins that require configuration, there are separate tabs in the interface.

Table 1.1. Plug-ins Installed With Orchestrator

<table>
<thead>
<tr>
<th>Plug-In</th>
<th>Purpose</th>
<th>Configuration</th>
</tr>
</thead>
<tbody>
<tr>
<td>vCenter Server</td>
<td>Provides access to the vCenter Server API so that you can incorporate all the vCenter Server objects and functions into the management processes that you automate by using Orchestrator.</td>
<td>See Configuring the vCenter Server Plug-In.</td>
</tr>
<tr>
<td>Configuration</td>
<td>Provides workflows for configuring the Orchestrator authentication, database connection, SSL certificates, and so on.</td>
<td>None</td>
</tr>
<tr>
<td>Plug-In</td>
<td>Purpose</td>
<td>Configuration</td>
</tr>
<tr>
<td>-------------</td>
<td>---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
<td>---------------</td>
</tr>
<tr>
<td>Library</td>
<td>Provides workflows that act as basic building blocks for customization and automation of client processes. The workflow library includes templates for life cycle management, provisioning, disaster recovery, hot backup, and other standard system management processes. You can copy and edit the templates to modify them according to your needs.</td>
<td>None</td>
</tr>
<tr>
<td>SQL</td>
<td>Provides the Java Database Connectivity (JDBC) API, which is the industry standard for database-independent connectivity between the Java programming language and a wide range of databases. The databases include SQL databases and other tabular data sources, such as spreadsheets or flat files. The JDBC API provides a call-level API for SQL-based database access from workflows.</td>
<td>None</td>
</tr>
<tr>
<td>SSH</td>
<td>Provides an implementation of the Secure Shell v2 (SSH-2) protocol. Allows remote command and file transfer sessions with password and public key-based authentication in workflows. Supports keyboard-interactive authentication. Optionally, the SSH plug-in can provide remote file system browsing directly in the Orchestrator client inventory.</td>
<td>See Configure the SSH Plug-In.</td>
</tr>
<tr>
<td>XML</td>
<td>A complete Document Object Model (DOM) XML parser that you can implement in workflows. Alternatively, you can use the ECMAScript for XML (E4X) implementation in the Orchestrator JavaScript API.</td>
<td>None</td>
</tr>
<tr>
<td>Mail</td>
<td>Uses Simple Mail Transfer Protocol (SMTP) to send email from workflows.</td>
<td>Set the default values for the EmailMessage object to use. See Define the Default SMTP Connection.</td>
</tr>
<tr>
<td>Net</td>
<td>Uses the Jakarta Apache Commons Net Library. Provides implementations of the Telnet, FTP, POP3, and IMAP protocols. The POP3 and IMAP protocols is used for reading email. In combination with the Mail plug-in, the Net plug-in provides complete email sending and receiving capabilities in workflows.</td>
<td>None</td>
</tr>
<tr>
<td>Workflow documentation</td>
<td>Provides workflows that you can use to generate information in PDF format about a workflow or a workflow category.</td>
<td>None</td>
</tr>
<tr>
<td>Enumeration</td>
<td>Provides common Enumerated Types that can be used in workflows by other plug-ins.</td>
<td>None</td>
</tr>
<tr>
<td>HTTP-REST</td>
<td>Enables management of REST Web services through an interaction between vRealize Orchestrator and REST hosts.</td>
<td>See Configuring the HTTP-REST Plug-In.</td>
</tr>
<tr>
<td>SOAP</td>
<td>Lets you manage SOAP Web services by providing interaction between vRealize Orchestrator and SOAP hosts.</td>
<td>See Configuring the SOAP Plug-In.</td>
</tr>
</tbody>
</table>
## Table 1-1. Plug-ins Installed With Orchestrator (Continued)

<table>
<thead>
<tr>
<th>Plug-In</th>
<th>Purpose</th>
<th>Configuration</th>
</tr>
</thead>
<tbody>
<tr>
<td>AMQP</td>
<td>Lets you interact with Advanced Message Queuing Protocol (AMQP) servers also known as brokers.</td>
<td>See Configuring the AMQP Plug-In.</td>
</tr>
<tr>
<td>SNMP</td>
<td>Enables vRealize Orchestrator to connect and receive information from SNMP-enabled systems and devices.</td>
<td>None</td>
</tr>
<tr>
<td>Active Directory</td>
<td>Provides interaction between vRealize Orchestrator and Microsoft Active Directory.</td>
<td>See Configuring the Active Directory Plug-In.</td>
</tr>
<tr>
<td>Dynamic Types</td>
<td>Lets you define dynamic types and create and use objects of these dynamic types.</td>
<td>See Chapter 18 Using the Dynamic Types Plug-In.</td>
</tr>
<tr>
<td>Multi-Node</td>
<td>Contains workflows for hierarchical management, management of Orchestrator instances, and scale-out of Orchestrator activities.</td>
<td>See Chapter 20 Using the Multi-Node Plug-In.</td>
</tr>
<tr>
<td>PowerShell</td>
<td>Lets you manage PowerShell hosts and run custom PowerShell operations.</td>
<td>See Chapter 19 Using the PowerShell Plug-In.</td>
</tr>
</tbody>
</table>

## Plug-In Components

Each plug-in is a DAR file package. The DAR files are stored in `/var/lib/vco/app-server/plugins` on the Orchestrator Appliance. The components of each plug-in, such as workflow categories and API modules, use different naming conventions.

## Table 1-2. Names of Plug-In Components

<table>
<thead>
<tr>
<th>Plug-In Name in the Configuration UI</th>
<th>DAR File</th>
<th>Workflow Categories</th>
<th>API Module</th>
</tr>
</thead>
<tbody>
<tr>
<td>vCenter Server</td>
<td>o11nplugin-vsphere.dar</td>
<td>vCenter</td>
<td>VC</td>
</tr>
<tr>
<td>vRO Configuration</td>
<td>o11nplugin-configurator.dar</td>
<td>Configuration</td>
<td>Configurator</td>
</tr>
<tr>
<td>Library</td>
<td>o11nplugin-library.dar</td>
<td>Locking Orchestrator Troubleshooting</td>
<td>Not applicable.</td>
</tr>
<tr>
<td>SQL</td>
<td>o11nplugin-database.dar</td>
<td>JDBC</td>
<td>SQL</td>
</tr>
<tr>
<td>SSH</td>
<td>o11nplugin-ssh.dar</td>
<td>SSH</td>
<td>SSH</td>
</tr>
<tr>
<td>XML</td>
<td>o11nplugin-xml.dar</td>
<td>XML</td>
<td>XML</td>
</tr>
<tr>
<td>Mail</td>
<td>o11nplugin-mail.dar</td>
<td>Mail</td>
<td>Mail</td>
</tr>
<tr>
<td>Net</td>
<td>o11nplugin-jakartocommonsnet.dar</td>
<td>None</td>
<td>Net</td>
</tr>
<tr>
<td>Workflow documentation</td>
<td>o11nplugin-wfdocs.dar</td>
<td>Workflow documentation</td>
<td>Workflow documentation</td>
</tr>
<tr>
<td>Common enumerated types</td>
<td>o11nplugin-enums.dar</td>
<td>None</td>
<td>Enums</td>
</tr>
<tr>
<td>Dynamic Types</td>
<td>o11n-plugin-dynamictypes.dar</td>
<td>Configuration</td>
<td>DynamicTypes</td>
</tr>
<tr>
<td>HTTP-REST</td>
<td>o11nplugin-rest.dar</td>
<td>Configuration</td>
<td>REST</td>
</tr>
</tbody>
</table>
Table 1-2. Names of Plug-In Components (Continued)

<table>
<thead>
<tr>
<th>Plug-In Name in the Configuration UI</th>
<th>DAR File</th>
<th>Workflow Categories</th>
<th>API Module</th>
</tr>
</thead>
<tbody>
<tr>
<td>SOAP</td>
<td>o11n-plugin-soap.dar</td>
<td>Configuration</td>
<td>SOAP</td>
</tr>
<tr>
<td>AMQP</td>
<td>o11n-plugin-amqp.dar</td>
<td>Configuration</td>
<td>AMQP</td>
</tr>
<tr>
<td>SNMP</td>
<td>o11n-plugin-snmp.dar</td>
<td>Device Management</td>
<td>SNMP</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Query Management</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Trap Host</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Management</td>
<td></td>
</tr>
<tr>
<td>Active Directory</td>
<td>o11nplugin-ad.dar</td>
<td>Computer</td>
<td>AD</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Configuration</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Organizational Unit</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>User</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>User Group</td>
<td></td>
</tr>
<tr>
<td>Orchestator</td>
<td>o11nplugin-multi-node.dar</td>
<td>Servers</td>
<td>VCO</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Configuration</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Remote Execution</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Remote Management</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Management</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Tasks</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Workflows</td>
<td></td>
</tr>
<tr>
<td>PowerShell</td>
<td>o11nplugin-powershell.dar</td>
<td>Configuration</td>
<td>PowerShell</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Generate</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Templates</td>
<td></td>
</tr>
</tbody>
</table>

Access the Orchestrator API Explorer

Orchestrator provides an API Explorer that you can use to search the Orchestrator API and see the documentation for JavaScript objects that you can use in scripted elements.

You can consult an online version of the Scripting API for the vCenter Server plug-in on the Orchestrator documentation home page.

Procedure

1. Log in to the Orchestrator client.
2. Select Tools > API Explorer.

The API Explorer appears. You can use it to search all the objects and functions of the Orchestrator API.

What to do next

Use the API Explorer to write scripts for scriptable elements.
Configure the Orchestrator Plug-Ins

The default Orchestrator plug-ins are configured only through workflows.

If you want to configure any of the default Orchestrator plug-ins, you need to use the specific workflow from the Orchestrator client.

This chapter includes the following topics:

- Manage the Orchestrator Plug-Ins
- Uninstall a Plug-In
- Reinstall Plug-Ins

Manage the Orchestrator Plug-Ins

In the Manage Plug-Ins page of Control Center, you can view a list of all plug-ins that are installed in Orchestrator and perform basic management actions.

Change Plug-Ins Logging Level

Instead of changing the logging level for Orchestrator, you can change it only for specific plug-ins.

Install a New Plug-In

With the Orchestrator plug-ins, the Orchestrator server can integrate with other software products. The Orchestrator Appliance includes a set of preinstalled plug-ins and you can also install custom plug-ins.

All Orchestrator plug-ins are installed from Control Center. The file extensions that can be used are .vmoapp and .dar. A .vmoapp file can contain a collection of several .dar files and can be installed as an application, while a .dar file contains all the resources associated with one plug-in.

Disable a Plug-In

You can disable a plug-in by deselecting the Enable check box next to the name of the plug-in.

This action does not remove the plug-in file. For more information on uninstalling a plug-in in Orchestrator, see Uninstall a Plug-In.
Uninstall a Plug-In

You can use Control Center to disable a plug-in, but this action does not remove the plug-in file from the Orchestrator Appliance file system. To remove the plug-in file, you must log in to the Orchestrator appliance and remove the plug-in file manually.

Procedure

1. Log in to the Orchestrator Appliance as root over SSH.
2. Open the `/etc/vco/app-server/plugins/_VSOPluginInstallationVersion.xml` file with a text editor.
   a. Delete the line of code that corresponds to the plug-in that you want to remove.
4. Delete the `.dar` archives that contain the plug-in that you want to remove.
5. Restart the vRealize Orchestrator services.
   The plug-in is removed from Control Center.
7. Select Administer from the drop-down menu in the upper-left corner.
8. Click the Packages view.
9. Right-click the package that you want to delete, and select Delete element with content.

   Note Orchestrator elements that are locked in the read-only state, for example workflows in the standard library, are not deleted.
10. Click Delete all.
11. Restart the vRealize Orchestrator services.

You removed all custom workflows, actions, policies, configurations, settings, and resources related to the plug-in.

Reinstall Plug-Ins

You must reinstall all Orchestrator plug-ins when you change the database server.

Prerequisites

Stop the Orchestrator server service from the Startup Options page in Control Center.

Procedure

1. Log in to Control Center as root.
2. Click Troubleshooting.
3 Click **Force Plug-ins Reinstall**.

The installed plug-ins are forced to reinstall the next time the Orchestrator server service starts.
Using the vCenter Server Plug-In

You can use the vCenter Server plug-in to manage multiple vCenter Server instances. You can create workflows that use the vCenter Server plug-in API to automate tasks in your vCenter Server environment.

The vCenter Server plug-in maps the vCenter Server API to the JavaScript that you can use in workflows. The plug-in also provides actions that perform individual vCenter Server tasks that you can include in workflows.

The vCenter Server plug-in provides a library of standard workflows that automate vCenter Server operations. For example, you can run workflows that create, clone, migrate, or delete virtual machines.

This chapter includes the following topics:

- Configuring the vCenter Server Plug-In
- vCenter Server Plug-In Scripting API
- Using the vCenter Server Plug-In Inventory
- Using XPath Expressions with the vCenter Server Plug-In
- Access the vCenter Server Plug-In Workflow Library
- vCenter Server Plug-In Workflow Library

Configuring the vCenter Server Plug-In

Before managing the objects in your vSphere inventory by using Orchestrator and to run workflows on the objects, you must configure the vCenter Server plug-in and define the connection parameters between Orchestrator and the vCenter Server instances you want to orchestrate.

You can configure the vCenter Server plug-in by running the vCenter Server configuration workflows from the Orchestrator client.

To manage the objects in your vSphere inventory by using the vSphere Web Client, make sure that you configure the Orchestrator server to work with the same vCenter Single Sign-On instance to which both vCenter Server and vSphere Web Client are pointing. You must also ensure that Orchestrator is registered as a vCenter Server extension. You register Orchestrator as a vCenter Server extension when you specify a user (by providing the user name and password), who has the privileges to manage vCenter Server extensions.
Configuration Workflows

The Configuration workflow category of the vCenter Server plug-in contains workflows that let you manage the connections to vCenter Server instances.

You can access these workflows from Library > vCenter > Configuration in the Workflows view of the Orchestrator client.

<table>
<thead>
<tr>
<th>Workflow Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Add a vCenter Server instance</td>
<td>Configures Orchestrator to connect to a new vCenter Server instance so that you can run workflows over the objects in the vSphere infrastructure.</td>
</tr>
<tr>
<td>List the vRealize Orchestrator extensions of vCenter Server</td>
<td>Lists all vRealize Orchestrator extensions of vCenter Server.</td>
</tr>
<tr>
<td>Register Orchestrator as a vCenter Server extension</td>
<td>Registers the Orchestrator instance as a vCenter Server extension.</td>
</tr>
<tr>
<td>Remove a vCenter Server instance</td>
<td>Removes a vCenter Server instance from the Orchestrator inventory. You will no longer be able to orchestrate this vCenter Server instance.</td>
</tr>
<tr>
<td>Update a vCenter Server instance</td>
<td>Updates the connection to a vCenter Server instance. For example, if the IP address of your vCenter Server system changes, you must update the connection parameters to the vCenter Server instance so that you can manage your vSphere inventory with Orchestrator.</td>
</tr>
<tr>
<td>Unregister a vCenter Server extension</td>
<td>Unregisters a vSphere Web Client extension.</td>
</tr>
</tbody>
</table>

Configure the Connection to a vCenter Server Instance

You can configure the connections to vCenter Server instances by running the vCenter Server configuration workflows in the Orchestrator client.

Procedure

1. Log in to the Orchestrator client as an administrator.
2. Click the Workflows view in the Orchestrator client.
3. In the workflows hierarchical list, expand Library > vCenter > Configuration and navigate to the Add a vCenter Server instance workflow.
4. Right-click the Add a vCenter Server instance workflow and select Start workflow.
5. Enter the IP address or the DNS name of the machine on which the vCenter Server instance you want to add is installed.
   
   **Note**   The hostname that you enter is case-sensitive.

6. Retain the default port value, 443.
7. Retain the default location of the SDK to use to connect to your vCenter Server instance.
8. Select whether you want to manage the vCenter Server instance through Orchestrator, and click Next.
9 Select whether you want to ignore certificate warnings for the vCenter Server instances that you want to add.

If you choose to ignore certificate warnings, the vCenter Server instance certificate is accepted silently and the certificate is added to the trusted store.

10 Select the method that you want to use to manage user access on the vCenter Server system.

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
</table>
| Share a unique session | Allows Orchestrator to create only one connection to vCenter Server.  
In the **User name** and **Password** text boxes, enter the credentials for Orchestrator to use to establish the connection to the vCenter Server host.  
The user that you select must be a valid user with privileges to manage vCenter Server extensions and a set of custom defined privileges. Orchestrator uses these credentials to monitor the VirtualCenter Web service, typically to run Orchestrator system workflows. |
| Session per user     | Creates a new session to vCenter Server. This action might rapidly use CPU, memory, and bandwidth.  
Select this option only if your vCenter Server is in an Active Directory domain or if vCenter Server Single Sign-On is enabled.  
The user that you select must be a valid user with privileges to manage vCenter Server extensions. |

The user account that you select is also used by the policy engine to collect statistical and other data. If the user that you select does not have enough privileges, the policy engine cannot access the necessary parts of the vCenter Server inventory and cannot collect the necessary data.

11 (Optional) Enter the user domain.

You must specify the user domain name only when you select to use a shared session.

**Note** Fill this text box if you are using LDAP authentication and session per user is selected.

12 Click **Submit** to run the workflow.

After the workflow runs successfully, the vCenter Server instance and all vSphere objects that belong to it appear in the **Inventory** view.

**vCenter Server Plug-In Scripting API**

The vCenter Server scripting API contains classes, with their respective attributes, methods, and constructors that allow interaction between vRealize Orchestrator and vCenter Server. You can use the API to develop custom workflows.

Using the vCenter Server Plug-In Inventory

The vCenter Server plug-in exposes all objects of the connected vCenter Server instances in the Inventory view. You can use the Inventory tab to add authorization elements or to run workflows on vCenter Server objects.

If you enable the Use contextual menu in inventory option from the Inventory tab of the User preferences tool, all of the workflows that you can run on the selected inventory object appear in a pop-up menu.

Using XPath Expressions with the vCenter Server Plug-In

You can use the finder methods in the vCenter Server plug-in to query for vCenter Server inventory objects. You can use XPath expressions to define search parameters.

The vCenter Server plug-in includes a set of object finder methods such as getAllDatastores(), getAllResourcePools(), findAllForType(). You can use these methods to access the inventories of the vCenter Server instances that are connected to your Orchestrator server and search for objects by ID, name, or other properties.

For performance reasons, the finder returns only a set of default properties for the queried object.

You can consult an online version of the Scripting API for the vCenter Server plug-in on the Orchestrator documentation home page.

**Important** The queries based on XPath expressions might impact the Orchestrator performance because the finder method returns all objects of a given type on the vCenter Server side and the query filters are applied on the vCenter Server plug-in side.

Using XPath Expressions with the vCenter Server Plug-In

When you invoke a finder method, you can use expressions based on the XPath query language. The search returns all the inventory objects that match the XPath expressions and the default properties of these objects. If you want to query for any non-default properties, you can include them to the search script in the form of a string array.

The following JavaScript example uses the VcPlugin scripting object and an XPath expression to return the names of all datastore objects that are part of the vCenter Server managed objects and contain the string ds in their names.

```javascript
var datastores = VcPlugin.getAllDatastores(null, "xpath:name[contains(.,'ds')]");
for each (datastore in datastores){
    System.log(datastore.name);
}
```
The same XPath expression can be invoked by using the Server scripting object and the findAllForType finder method.

```javascript
var datastores = Server.findAllForType("VC:Datastore", "xpath:name[contains(.,'ds')]textContent");
for each (datastore in datastores){
   System.log(datastore.name);
}
```

The following script example returns the names of all host system objects whose ID starts with the digit 1.

```javascript
var hosts = VcPlugin.getAllHostSystems(null, "xpath:id[starts-with(.,'1')]textContent");
for each (host in hosts){
   System.log(host.name);
}
```

The following script returns the names and IDs of all data center objects that contain the string DC, in upper- or lower-case letters, in their names. In addition to the default set of object properties, the script also retrieves the tag property.

```javascript
var datacenters = VcPlugin.getAllDatacenters(['tag'], "xpath:name[contains(translate(.,,'DC','dc'),'dc')]textContent");
for each (datacenter in datacenters){
   System.log(datacenter.name + " " + datacenter.id);
}
```

**Access the vCenter Server Plug-In Workflow Library**

You must use the Orchestrator client or the vSphere Web Client to access the elements from the vCenter Server plug-in workflow library.

**Prerequisites**

- Configure a connection to a vCenter Server instance.
- Verify that the user account you are logged in with has the necessary permissions to run vCenter Server workflows.

**Procedure**

1. In the Orchestrator client, select **Design** or **Run** from the drop-down menu in the left upper corner.
2. Click the **Workflows** view in the Orchestrator client left pane.
3. Expand the hierarchical list to **Library > vCenter**.

**What to do next**

Review the workflow library.
vCenter Server Plug-In Workflow Library

The vCenter Server plug-in workflow library contains workflows that you can use to run automated processes related to the management of vCenter Server.

- **Batch Workflows**
  Batch workflows populate configuration elements or run workflows on a selected vCenter Server object.

- **Cluster and Compute Resource Workflows**
  With the cluster and compute resource workflows, you can create, rename, or delete a cluster. You can also enable or disable high availability, Distributed Resource Scheduler, and vCloud Distributed Storage on a cluster.

- **Configuration Workflows**
  The Configuration workflow category of the vCenter Server plug-in contains workflows that let you manage the connections to vCenter Server instances.

- **Custom Attributes Workflows**
  With custom attributes workflows, you can add custom attributes to virtual machines or get a custom attribute for a virtual machine.

- **Datacenter Workflows**
  With datacenter workflows, you can create, delete, reload, rename, or rescan a datacenter.

- **Datastore and Files Workflows**
  With the datastore and files workflows, you can delete a list of files, find unused files in a datastore, and so on.

- **Datacenter Folder Management Workflows**
  With datacenter folder management workflows, you can create, delete, or rename a datacenter folder.

- **Host Folder Management Workflows**
  With host folder management workflows, you can create, delete, or rename a host folder.

- **Virtual Machine Folder Management Workflows**
  With virtual machine folder management workflows, you can create, delete, or rename a virtual machine folder.

- **Guest Operation Files Workflows**
  With the guest operation files workflows, you can manage files in a guest operating system.

- **Guest Operation Processes Workflows**
  With guest operation processes workflows, you can get information and control the running processes in a guest operating system.
- **Power Host Management Workflows**
  With power host management workflows you can reboot or shut down a host.

- **Basic Host Management Workflows**
  With the basic host management workflows, you can put a host into maintenance mode and make a host exit maintenance mode. You can also move a host to a folder or a cluster, and reload data from a host.

- **Host Registration Management Workflows**
  With the host registration management workflows, you can add a host to a cluster, disconnect, or reconnect a host from a cluster, and so on.

- **Networking Workflows**
  With networking workflows you can add a port group to distributed virtual switch, create a distributed virtual switch with a port group, and so on.

- **Distributed Virtual Port Group Workflows**
  With the distributed virtual port group workflows, you can update or delete a port group, and reconfigure the port group.

- **Distributed Virtual Switch Workflows**
  With distributed virtual switch workflows, you can create, update or delete a distributed virtual switch, and create, delete, or update a private VLAN.

- **Standard Virtual Switch Workflows**
  With the standard virtual switch workflows you can create, update, or delete a standard virtual switch, and create, delete, or update port groups in standard virtual switches.

- **Networking Virtual SAN Workflows**
  With Virtual SAN workflows, you can configure Virtual SAN network traffic.

- **Resource Pool Workflows**
  With the resource pool workflows you can create, rename, reconfigure or delete a resource pool, and get resource pool information.

- **Storage Workflows**
  With the storage workflows, you can perform storage-related operations.

- **Storage DRS Workflows**
  With the storage DRS workflows, you perform storage-related operations, such as creating and configuring a datastore cluster, removing a datastore from cluster, adding storage to a cluster, and others.

- **Storage VSAN Workflows**
  With the Virtual SAN workflows, you can manage non-SSD disks and disk groups in a Virtual SAN cluster.
Basic Virtual Machine Management Workflows
With the basic virtual machine management workflows, you can perform basic operations on virtual machines, for example, create, rename or delete a virtual machine, upgrade virtual hardware, and others.

Clone Workflows
With clone workflows, you can clone virtual machines with or without customizing the virtual machine properties.

Linked Clone Workflows
With the linked clone workflows, you can perform linked clone operations such as restoring a virtual machine from a linked clone, creating a linked clone, or others.

Linux Customization Clone Workflows
With Linux customization workflows, you can clone a Linux virtual machine and customize the guest operating system.

Tools Clone Workflows
With the tools clone workflows, you can obtain customization information about the operating system of the virtual machine, information required to update a virtual device, and others.

Windows Customization Clone Workflows
With the Windows customization clone workflows, you can clone Windows virtual machines and customize the guest operating system.

Device Management Workflows
With the device management workflows, you can manage the devices that are connected to a virtual machine or to a host datastore.

Move and Migrate Workflows
With the move and migrate workflows, you can migrate virtual machines.

Other Workflows
With the workflows from the Others category, you can enable and disable Fault Tolerance (FT), extract virtual machine information, and find orphaned virtual machines.

Power Management Workflows
With the power management workflows, you can power on and off virtual machines, reboot the guest operating system of a virtual machine, suspend a virtual machine, and others.

Snapshot Workflows
With snapshot workflows, you can perform snapshot-related operations.

VMware Tools Workflows
With VMware Tools workflows, you can perform VMware Tools-related tasks on virtual machines.

Batch Workflows
Batch workflows populate configuration elements or run workflows on a selected vCenter Server object.
You can access the batch workflows from **Library > vCenter > Batch** in the **Workflows** view of the Orchestrator client.

<table>
<thead>
<tr>
<th>Workflow Name</th>
<th>Description</th>
</tr>
</thead>
</table>
| Fill batch configuration elements | Populates the configuration elements that the Run a workflow on a selection of objects workflow uses. Performs the following tasks:  
  - Resets the BatchObject and BatchAction configuration elements.  
  - Fills the BatchObject configuration element with all the workflows that have only one input parameter.  
  - Fills the BatchAction configuration element with all the actions that have no input parameters or one input parameter and that have an array as the returnType. |
| Run a workflow on a selection of objects | Runs a workflow on a selection of vCenter Server objects, taking one action as input. This is the action that retrieves the list of objects on which to run the workflow. To return the objects without running the selected workflow, run the workflow in simulation mode. |

**Cluster and Compute Resource Workflows**

With the cluster and compute resource workflows, you can create, rename, or delete a cluster. You can also enable or disable high availability, Distributed Resource Scheduler, and vCloud Distributed Storage on a cluster.

You can access the cluster and compute resource workflows from **Library > vCenter > Cluster and Compute Resource** in the **Workflows** view of the Orchestrator client.

<table>
<thead>
<tr>
<th>Workflow Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Add DRS virtual machine group to cluster</td>
<td>Adds a DRS virtual machine group to a cluster.</td>
</tr>
<tr>
<td>Add virtual machines to DRS group</td>
<td>Adds a virtual machine list to an existing DRS virtual machine group.</td>
</tr>
<tr>
<td>Create cluster</td>
<td>Creates a cluster in a host folder.</td>
</tr>
<tr>
<td>Delete cluster</td>
<td>Deletes a cluster.</td>
</tr>
<tr>
<td>Disable DRS on cluster</td>
<td>Disables DRS on a cluster.</td>
</tr>
<tr>
<td>Disable HA on cluster</td>
<td>Disables high availability on a cluster.</td>
</tr>
<tr>
<td>Disable vCloud Distributed Storage on cluster</td>
<td>Disables vCloud Distributed Storage on a cluster.</td>
</tr>
<tr>
<td>Enable DRS on cluster</td>
<td>Enables DRS on a cluster.</td>
</tr>
<tr>
<td>Enable HA on cluster</td>
<td>Enables high availability on a cluster.</td>
</tr>
<tr>
<td>Enable vCloud Distributed Storage on cluster</td>
<td>Enables vCloud Distributed Storage on a cluster.</td>
</tr>
<tr>
<td>Remove virtual machine DRS group from cluster</td>
<td>Removes a DRS virtual machine group from a cluster.</td>
</tr>
<tr>
<td>Remove virtual machines from DRS group</td>
<td>Removes virtual machines from a cluster DRS group.</td>
</tr>
<tr>
<td>Rename cluster</td>
<td>Renames a cluster.</td>
</tr>
</tbody>
</table>

**Configuration Workflows**

The Configuration workflow category of the vCenter Server plug-in contains workflows that let you manage the connections to vCenter Server instances.
You can access these workflows from **Library > vCenter > Configuration** in the **Workflows** view of the Orchestrator client.

<table>
<thead>
<tr>
<th>Workflow Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Add a vCenter Server instance</td>
<td>Configures Orchestrator to connect to a new vCenter Server instance so that you can run workflows over the objects in the vSphere infrastructure.</td>
</tr>
<tr>
<td>List the Orchestrator extensions of vCenter Server</td>
<td>Lists all Orchestrator extensions of vCenter Server.</td>
</tr>
<tr>
<td>Register Orchestrator as a vCenter Server extension</td>
<td>Registers the Orchestrator instance as a vCenter Server extension.</td>
</tr>
<tr>
<td>Remove a vCenter Server instance</td>
<td>Removes a vCenter Server instance from the Orchestrator inventory. You cannot orchestrate this vCenter Server instance any longer.</td>
</tr>
<tr>
<td>Update a vCenter Server instance</td>
<td>Updates the connection to a vCenter Server instance. For example, if the IP address of your vCenter Server system changes, you must update the connection parameters to the vCenter Server instance so that you can manage your vSphere inventory with Orchestrator.</td>
</tr>
<tr>
<td>Unregister a vCenter Server extension</td>
<td>Unregisters a vCenter Server extension.</td>
</tr>
</tbody>
</table>

### Custom Attributes Workflows

With custom attributes workflows, you can add custom attributes to virtual machines or get a custom attribute for a virtual machine.

You can access the custom attributes workflows from **Library > vCenter > Custom Attributes** in the **Workflows** view of the Orchestrator client.

<table>
<thead>
<tr>
<th>Workflow Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Add custom attribute to a virtual machine</td>
<td>Adds a custom attribute to a virtual machine.</td>
</tr>
<tr>
<td>Add custom attribute to multiple virtual machines</td>
<td>Adds a custom attribute to a selection of virtual machines.</td>
</tr>
<tr>
<td>Get custom attribute</td>
<td>Gets a custom attribute for a virtual machine in vCenter Server.</td>
</tr>
</tbody>
</table>

### Datacenter Workflows

With datacenter workflows, you can create, delete, reload, rename, or rescan a datacenter.

You can access the datacenter workflows from **Library > vCenter > Datacenter** in the **Workflows** view of the Orchestrator client.

<table>
<thead>
<tr>
<th>Workflow Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Create datacenter</td>
<td>Creates a data center in a data center folder.</td>
</tr>
<tr>
<td>Delete datacenter</td>
<td>Deletes a data center.</td>
</tr>
<tr>
<td>Reload datacenter</td>
<td>Forces vCenter Server to reload data from a data center.</td>
</tr>
<tr>
<td>Rename datacenter</td>
<td>Renames a data center and waits for the task to complete.</td>
</tr>
<tr>
<td>Rescan datacenter HBAs</td>
<td>Scans the hosts in a data center and initiates a rescan on the host bus adapters to discover new storage.</td>
</tr>
</tbody>
</table>
Datastore and Files Workflows

With the datastore and files workflows, you can delete a list of files, find unused files in a datastore, and so on.

You can access the datastore and files workflows from **Library > vCenter > Datastore and Files** in the **Workflows** view of the Orchestrator client.

<table>
<thead>
<tr>
<th>Workflow Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Delete all files</td>
<td>Deletes a list of files.</td>
</tr>
<tr>
<td>Delete all unused datastore files</td>
<td>Searches all datastores in the vCenter Server environment and deletes all unused files.</td>
</tr>
<tr>
<td>Export unused datastore files</td>
<td>Searches all datastores and creates an XML descriptor file that lists all unused files.</td>
</tr>
<tr>
<td>Find unused files in datastores</td>
<td>Searches the vCenter Server environment for all unused disks (<em>.vmdk), virtual machines (</em>.vmx), and template (*.vmtx) files that are not associated with any vCenter Server instances registered with Orchestrator.</td>
</tr>
<tr>
<td>Get all configuration, template, and disk files from virtual machines</td>
<td>Creates a list of all virtual machine descriptor files and a list of all virtual machine disk files, for all datastores.</td>
</tr>
<tr>
<td>Log all datastore files</td>
<td>Creates a log for every virtual machine configuration file and every virtual machine file found in all datastores.</td>
</tr>
<tr>
<td>Log unused datastore files</td>
<td>Searches the vCenter Server environment for unused files that are registered on virtual machines and exports a log of the files in a text file.</td>
</tr>
<tr>
<td>Upload file to datastore</td>
<td>Uploads a file to an existing folder on a specific datastore. The uploaded file overwrites any existing file with the same name in the same destination folder.</td>
</tr>
</tbody>
</table>

Datacenter Folder Management Workflows

With datacenter folder management workflows, you can create, delete, or rename a datacenter folder.

You can access the datacenter folder management workflows from **Library > vCenter > Folder management > Datacenter folder** in the **Workflows** view of the Orchestrator client.

<table>
<thead>
<tr>
<th>Workflow Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Create datacenter folder</td>
<td>Creates a data center folder.</td>
</tr>
<tr>
<td>Delete datacenter folder</td>
<td>Deletes a data center folder and waits for the task to complete.</td>
</tr>
<tr>
<td>Rename datacenter folder</td>
<td>Renames a data center folder and waits for the task to complete.</td>
</tr>
</tbody>
</table>

Host Folder Management Workflows

With host folder management workflows, you can create, delete, or rename a host folder.

You can access the host folder management workflows from **Library > vCenter > Folder management > Host folder** in the **Workflows** view of the Orchestrator client.
Virtual Machine Folder Management Workflows

With virtual machine folder management workflows, you can create, delete, or rename a virtual machine folder.

You can access the virtual machine folder management workflows from Library > vCenter > Folder management > VM folder in the Workflow view of the Orchestrator client.

<table>
<thead>
<tr>
<th>Workflow Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Create virtual machine folder</td>
<td>Creates a virtual machine folder.</td>
</tr>
<tr>
<td>Delete virtual machine folder</td>
<td>Deletes a virtual machine folder and waits for the task to complete.</td>
</tr>
<tr>
<td>Rename virtual machine folder</td>
<td>Renames a virtual machine folder and waits for the task to complete.</td>
</tr>
</tbody>
</table>

Guest Operation Files Workflows

With the guest operation files workflows, you can manage files in a guest operating system.

You can access the guest operation files workflows from Library > vCenter > Guest operations > Files in the Workflows view of the Orchestrator client.

<table>
<thead>
<tr>
<th>Workflow Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Check for directory in guest</td>
<td>Verifies that a directory exists in a guest virtual machine.</td>
</tr>
<tr>
<td>Check for file in guest</td>
<td>Verifies that a file exists in a guest virtual machine.</td>
</tr>
<tr>
<td>Copy file from guest to Orchestrator</td>
<td>Copies a specified file from a guest file system to an Orchestrator server.</td>
</tr>
<tr>
<td>Copy file from Orchestrator to guest</td>
<td>Copies a specified file from an Orchestrator server to a guest file system.</td>
</tr>
<tr>
<td>Create directory in guest</td>
<td>Creates a directory in a guest virtual machine.</td>
</tr>
<tr>
<td>Create temporary directory in guest</td>
<td>Creates a temporary directory in a guest virtual machine.</td>
</tr>
<tr>
<td>Create temporary file in guest</td>
<td>Creates a temporary file in a guest virtual machine.</td>
</tr>
<tr>
<td>Delete directory in guest</td>
<td>Deletes a directory from a guest virtual machine.</td>
</tr>
<tr>
<td>Delete file in guest</td>
<td>Deletes a file from a guest virtual machine.</td>
</tr>
<tr>
<td>List path in guest</td>
<td>Shows a path in a guest virtual machine.</td>
</tr>
<tr>
<td>Move directory in guest</td>
<td>Moves a directory in a guest virtual machine.</td>
</tr>
<tr>
<td>Move file in guest</td>
<td>Moves a file in a guest virtual machine.</td>
</tr>
</tbody>
</table>

Guest Operation Processes Workflows

With guest operation processes workflows, you can get information and control the running processes in a guest operating system.
You can access the guest operation files workflows from **Library > vCenter > Guest operations > Processes** in the **Workflows** view of the Orchestrator client.

<table>
<thead>
<tr>
<th>Workflow Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Get environment variables from guest</td>
<td>Returns a list with environmental variables from a guest. An interactive session returns the variables of the user who is currently logged in.</td>
</tr>
<tr>
<td>Get processes from guest</td>
<td>Returns a list with the processes running in the guest operating system and the recently completed processes started by the API.</td>
</tr>
<tr>
<td>Run program in guest</td>
<td>Starts a program in a guest operating system.</td>
</tr>
<tr>
<td>Kill process in guest</td>
<td>Terminates a process in a guest operating system.</td>
</tr>
</tbody>
</table>

### Power Host Management Workflows

With **power host management workflows** you can reboot or shut down a host.

You can access the power host management workflows from **Library > vCenter > Host management > Power** in the **Workflows** view of the Orchestrator client.

- **Reboot host**
  Reboots a host. If the Orchestrator client is connected directly to the host, it does not receive an indication of success in the returned task, but rather loses the connection to the host if the operation succeeds.

- **Shut down host**
  Shuts down a host. If the Orchestrator client is connected directly to the host, it does not receive an indication of success in the returned task, but rather loses the connection to the host if the operation succeeds.

### Basic Host Management Workflows

With the **basic host management workflows**, you can put a host into maintenance mode and make a host exit maintenance mode. You can also move a host to a folder or a cluster, and reload data from a host.

You can access the basic host management workflows from **Library > vCenter > Host management > Basic** in the **Workflows** view of the Orchestrator client.

<table>
<thead>
<tr>
<th>Workflow Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Enter maintenance mode</td>
<td>Puts the host into maintenance mode. You can cancel the task.</td>
</tr>
<tr>
<td>Exit maintenance mode</td>
<td>Exits maintenance mode. You can cancel the task.</td>
</tr>
<tr>
<td>Move host to cluster</td>
<td>Moves an existing host to a cluster. The host must be part of the same data center, and if the host is part of a cluster, the host must be in maintenance mode.</td>
</tr>
<tr>
<td>Move host to folder</td>
<td>Moves a host into a folder as a standalone host. The host must be part of a ClusterComputeResource in the same data center and the host must be in maintenance mode.</td>
</tr>
<tr>
<td>Reload host</td>
<td>Forces vCenter Server to reload data from a host.</td>
</tr>
</tbody>
</table>
Host Registration Management Workflows

With the host registration management workflows, you can add a host to a cluster, disconnect, or reconnect a host from a cluster, and so on.

You can access the host management registration workflows from Library > vCenter > Host management > Registration in the Workflows view of the Orchestrator client.

<table>
<thead>
<tr>
<th>Workflow Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Add host to cluster</td>
<td>Adds a host to the cluster. This workflow fails if it cannot authenticate the SSL certificate of the host.</td>
</tr>
<tr>
<td>Add standalone host</td>
<td>Registers a host as a standalone host.</td>
</tr>
<tr>
<td>Disconnect host</td>
<td>Disconnects a host from the vCenter Server instance.</td>
</tr>
<tr>
<td>Reconnect host</td>
<td>Reconnects a disconnected host by providing only the host information.</td>
</tr>
<tr>
<td>Reconnect host with all information</td>
<td>Reconnects a disconnected host by providing all information about the host.</td>
</tr>
<tr>
<td>Remove host</td>
<td>Removes a host and unregisters it from the vCenter Server instance. If the host is part of a cluster, you must put it in maintenance mode before attempting to remove it.</td>
</tr>
</tbody>
</table>

Networking Workflows

With networking workflows you can add a port group to distributed virtual switch, create a distributed virtual switch with a port group, and so on.

You can access the networking workflows from Library > vCenter > Networking in the Workflows view of the Orchestrator client.

<table>
<thead>
<tr>
<th>Workflow Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Add port group to distributed virtual switch</td>
<td>Adds a new distributed virtual port group to a specified distributed virtual switch.</td>
</tr>
<tr>
<td>Attach host system to distributed virtual switch</td>
<td>Adds a host to a distributed virtual switch.</td>
</tr>
<tr>
<td>Create distributed virtual switch with port group</td>
<td>Creates a new distributed virtual switch with a distributed virtual port group.</td>
</tr>
</tbody>
</table>

Distributed Virtual Port Group Workflows

With the distributed virtual port group workflows, you can update or delete a port group, and reconfigure the port group.

You can access the distributed virtual port group workflows from Library > vCenter > Networking > Distributed virtual port group in the Workflows view of the Orchestrator client.

<table>
<thead>
<tr>
<th>Workflow Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Connect virtual machine NIC number to distributed virtual port group</td>
<td>Reconfigures the network connection of the specified virtual machine NIC number to connect to the specified distributed virtual port group. If no NIC number is specified, the number zero is used.</td>
</tr>
<tr>
<td>Delete distributed virtual port group</td>
<td>Deletes a specified distributed virtual port group.</td>
</tr>
</tbody>
</table>
Distributed Virtual Switch Workflows

With distributed virtual switch workflows, you can create, update or delete a distributed virtual switch, and create, delete, or update a private VLAN.

You can access the distributed virtual switch workflows from Library > vCenter > Networking > Distributed virtual switch in the Workflows view of the Orchestrator client.

<table>
<thead>
<tr>
<th>Workflow Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Set teaming options</td>
<td>Provides an interface to manage the teaming options for a distributed virtual port group.</td>
</tr>
<tr>
<td>Update distributed virtual port group</td>
<td>Updates the configuration of a specified distributed virtual port group.</td>
</tr>
<tr>
<td>Create distributed virtual switch</td>
<td>Creates a distributed virtual switch in the specified network folder with a name and uplink port names that you specify. You must specify at least one uplink port name.</td>
</tr>
<tr>
<td>Create private VLAN</td>
<td>Creates a VLAN on the specified distributed virtual switch.</td>
</tr>
<tr>
<td>Delete distributed virtual switch</td>
<td>Deletes a distributed virtual switch and all associated elements.</td>
</tr>
<tr>
<td>Delete private VLAN</td>
<td>Deletes a VLAN from a specified distributed virtual switch. If a secondary VLAN exists, you must first delete the secondary VLAN.</td>
</tr>
<tr>
<td>Update distributed virtual switch</td>
<td>Updates the properties of a distributed virtual switch.</td>
</tr>
<tr>
<td>Update private VLAN</td>
<td>Updates a VLAN on the specified distributed virtual switch.</td>
</tr>
</tbody>
</table>

Standard Virtual Switch Workflows

With the standard virtual switch workflows you can create, update, or delete a standard virtual switch, and create, delete, or update port groups in standard virtual switches.

You can access the standard virtual switch workflows from Library > vCenter > Networking > Standard virtual switch in the Workflows view of the Orchestrator client.

<table>
<thead>
<tr>
<th>Workflow Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Add port group in standard virtual switch</td>
<td>Adds a port group in a standard virtual switch.</td>
</tr>
<tr>
<td>Create standard virtual switch</td>
<td>Creates a standard virtual switch.</td>
</tr>
<tr>
<td>Delete port group from standard virtual switch</td>
<td>Deletes a port group from a standard virtual switch.</td>
</tr>
<tr>
<td>Delete standard virtual switch</td>
<td>Deletes a standard virtual switch from a host network configuration.</td>
</tr>
<tr>
<td>Retrieve all standard virtual switches</td>
<td>Retrieves all standard virtual switches from a host.</td>
</tr>
<tr>
<td>Update port group in standard virtual switch</td>
<td>Updates the properties of a port group in a standard virtual switch.</td>
</tr>
<tr>
<td>Update standard virtual switch</td>
<td>Updates the properties of a standard virtual switch.</td>
</tr>
<tr>
<td>Update VNIC for port group in standard virtual switch</td>
<td>Updates a virtual NIC associated with a port group in a standard virtual switch.</td>
</tr>
</tbody>
</table>

Networking Virtual SAN Workflows

With Virtual SAN workflows, you can configure Virtual SAN network traffic.
You can access the networking workflows from Library > vCenter > Networking > VSAN in the Workflows view of the Orchestrator client.

<table>
<thead>
<tr>
<th>Workflow Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Set a cluster's VSAN traffic network</td>
<td>Sets a Virtual SAN traffic network of the cluster.</td>
</tr>
<tr>
<td>Set a host's VSAN traffic network</td>
<td>Sets a Virtual SAN traffic network of the host.</td>
</tr>
</tbody>
</table>

**Resource Pool Workflows**

With the resource pool workflows you can create, rename, reconfigure or delete a resource pool, and get resource pool information.

You can access the resource pool workflows from Library > vCenter > Resource Pool in the Workflows view of the Orchestrator client.

<table>
<thead>
<tr>
<th>Workflow Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Create resource pool</td>
<td>Creates a resource pool with the default CPU and memory allocation values. To create a resource pool in a cluster, the cluster must have VMware DRS enabled.</td>
</tr>
<tr>
<td>Create resource pool with specified values</td>
<td>Creates a resource pool with CPU and memory allocation values that you specify. To create a resource pool in a cluster, the cluster must have VMware DRS enabled.</td>
</tr>
<tr>
<td>Delete resource pool</td>
<td>Deletes a resource pool and waits for the task to complete.</td>
</tr>
<tr>
<td>Get resource pool information</td>
<td>Returns CPU and memory information about a given resource pool.</td>
</tr>
<tr>
<td>Reconfigure resource pool</td>
<td>Reconfigures CPU and memory allocation configuration for a given resource pool.</td>
</tr>
<tr>
<td>Rename resource pool</td>
<td>Renames a resource pool and waits for the task to complete.</td>
</tr>
</tbody>
</table>

**Storage Workflows**

With the storage workflows, you can perform storage-related operations.

You can access the storage workflows from Library > vCenter > Storage in the Workflows view of the Orchestrator client.

<table>
<thead>
<tr>
<th>Workflow Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Add datastore on iSCSI/FC/local SCSI</td>
<td>Creates a datastore on a Fibre Channel, iSCSI or local SCSI disk. Only disks that are not currently in use by an existing VMFS are applicable to new datastore creation. The new datastore allocates the maximum available space of the specified disk.</td>
</tr>
<tr>
<td>Add datastore on NFS</td>
<td>Adds a datastore on an NFS server.</td>
</tr>
<tr>
<td>Add iSCSI target</td>
<td>Adds iSCSI targets to a vCenter Server host. The targets can be of the type Send or Static.</td>
</tr>
<tr>
<td>Create VMFS for all available disks</td>
<td>Creates a VMFS volume for all available disks of a specified host.</td>
</tr>
<tr>
<td>Delete datastore</td>
<td>Deletes datastores from a vCenter Server host.</td>
</tr>
<tr>
<td>Delete iSCSI target</td>
<td>Deletes already configured iSCSI targets. The targets can be of type Send or Static.</td>
</tr>
<tr>
<td>Disable iSCSI adapter</td>
<td>Disables the software iSCSI adapter of a specified host.</td>
</tr>
<tr>
<td>Display all datastores and disks</td>
<td>Displays the existing datastores and available disks on a specified host.</td>
</tr>
<tr>
<td>Workflow Name</td>
<td>Description</td>
</tr>
<tr>
<td>-------------------------------</td>
<td>-----------------------------------------------------------------------------</td>
</tr>
<tr>
<td>Enable iSCSI adapter</td>
<td>Enables an iSCSI adapter.</td>
</tr>
<tr>
<td>List all storage adapters</td>
<td>Lists all storage adapters of a specified host.</td>
</tr>
</tbody>
</table>

**Storage DRS Workflows**

With the storage DRS workflows, you perform storage-related operations, such as creating and configuring a datastore cluster, removing a datastore from cluster, adding storage to a cluster, and others.

You can access the storage DRS workflows from **Library > vCenter > Storage > Storage DRS** in the **Workflows** view of the Orchestrator client.

<table>
<thead>
<tr>
<th>Workflow Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Add datastore to cluster</td>
<td>Adds datastores to a datastore cluster. Datastores must be able to connect to all hosts to be included in the datastore cluster. Datastores must have the same connection type to reside within a datastore cluster.</td>
</tr>
<tr>
<td>Change Storage DRS per virtual machine configuration</td>
<td>Sets Storage DRS settings for each virtual machine.</td>
</tr>
<tr>
<td>Configure datastore cluster</td>
<td>Configures datastore cluster setting values for automation and runtime rules.</td>
</tr>
<tr>
<td>Create simple datastore cluster</td>
<td>Creates a simple datastore cluster with default configuration. The new datastore cluster contains no datastores.</td>
</tr>
<tr>
<td>Create Storage DRS scheduled task</td>
<td>Creates a scheduled task for reconfiguring a datastore cluster. Only automation and runtime rules can be set.</td>
</tr>
<tr>
<td>Create virtual machine anti-affinity rule</td>
<td>Creates an anti-affinity rule to indicate that all virtual disks of certain virtual machines must be kept on different datastores.</td>
</tr>
<tr>
<td>Create VMDK anti-affinity rule</td>
<td>Creates a VMDK anti-affinity rule for a virtual machine that indicates which of its virtual disks must be kept on different datastores. The rule applies to the virtual disks of the selected virtual machine.</td>
</tr>
<tr>
<td>Remove datastore cluster</td>
<td>Removes a datastore cluster. Removing a datastore cluster also removes all the settings and the alarms for the cluster from the vCenter Server system.</td>
</tr>
<tr>
<td>Remove datastore from cluster</td>
<td>Removes a datastore from a datastore cluster and puts the datastore in a datastore folder.</td>
</tr>
<tr>
<td>Remove Storage DRS scheduled task</td>
<td>Removes a scheduled Storage DRS task.</td>
</tr>
<tr>
<td>Remove virtual machine anti-affinity rule</td>
<td>Removes a virtual machine anti-affinity rule for a given datastore cluster.</td>
</tr>
<tr>
<td>Remove VMDK anti-affinity rule</td>
<td>Removes a VMDK anti-affinity rule for a given datastore cluster.</td>
</tr>
</tbody>
</table>

**Storage VSAN Workflows**

With the Virtual SAN workflows, you can manage non-SSD disks and disk groups in a Virtual SAN cluster.

You can access the networking workflows from **Library > vCenter > Storage > VSAN** in the **Workflows** view of the Orchestrator client.
<table>
<thead>
<tr>
<th>Workflow Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Add disks to a disk group</td>
<td>Adds non-SSD disks to a Virtual SAN disk group.</td>
</tr>
<tr>
<td>Claim disks into disk groups</td>
<td>Claims disks for use by the Virtual SAN system and automatically creates disk groups and distributes the disks into existing disk groups.</td>
</tr>
<tr>
<td>Create a disk group</td>
<td>Creates a Virtual SAN disk group.</td>
</tr>
<tr>
<td>List hosts, disk groups and disks</td>
<td>Lists all hosts in a cluster, their disk groups and disks, used or eligible for use by the Virtual SAN system.</td>
</tr>
<tr>
<td>Remove disk groups</td>
<td>Removes Virtual SAN disk groups.</td>
</tr>
<tr>
<td>Remove disks from disk groups</td>
<td>Removes non-SSD disks from Virtual SAN disk groups.</td>
</tr>
</tbody>
</table>

### Basic Virtual Machine Management Workflows

With the basic virtual machine management workflows, you can perform basic operations on virtual machines, for example, create, rename or delete a virtual machine, upgrade virtual hardware, and others.

You can access the basic virtual machine management workflows from **Library > vCenter > Virtual Machine management > Basic** in the **Workflows** view of the Orchestrator client.

<table>
<thead>
<tr>
<th>Workflow Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Create custom virtual machine</td>
<td>Creates a virtual machine with the specified configuration options and additional devices.</td>
</tr>
<tr>
<td>Create simple dvPortGroup virtual machine</td>
<td>Creates a simple virtual machine. The network used is a Distributed Virtual Port Group.</td>
</tr>
<tr>
<td>Create simple virtual machine</td>
<td>Creates a virtual machine with the most common devices and configuration options.</td>
</tr>
<tr>
<td>Delete virtual machine</td>
<td>Removes a virtual machine from the inventory and datastore.</td>
</tr>
<tr>
<td>Get virtual machines by name</td>
<td>Returns a list of virtual machines from all registered vCenter Server instances that match the provided expression.</td>
</tr>
<tr>
<td>Mark as template</td>
<td>Converts an existing virtual machine to a template, not allowing it to start. You can use templates to create virtual machines.</td>
</tr>
<tr>
<td>Mark as virtual machine</td>
<td>Converts an existing template to a virtual machine, allowing it to start.</td>
</tr>
<tr>
<td>Move virtual machine to folder</td>
<td>Moves a virtual machine to a specified virtual machine folder.</td>
</tr>
<tr>
<td>Move virtual machine to resource pool</td>
<td>Moves a virtual machine to a resource pool. If the target resource pool is not in the same cluster, you must use the migrate or relocate workflows.</td>
</tr>
<tr>
<td>Move virtual machines to folder</td>
<td>Moves several virtual machines to a specified virtual machine folder.</td>
</tr>
<tr>
<td>Move virtual machines to resource pool</td>
<td>Moves several virtual machines to a resource pool.</td>
</tr>
<tr>
<td>Register virtual machine</td>
<td>Registers a virtual machine. The virtual machine files must be placed in an existing datastore and must not be already registered.</td>
</tr>
<tr>
<td>Reload virtual machine</td>
<td>Forces vCenter Server to reload a virtual machine.</td>
</tr>
<tr>
<td>Rename virtual machine</td>
<td>Renames an existing virtual machine on the vCenter Server system or host and not on the datastore.</td>
</tr>
<tr>
<td>Set virtual machine performance</td>
<td>Changes performance settings such as shares, minimum and maximum values, shaping for network, and disk access of a virtual machine.</td>
</tr>
</tbody>
</table>
### Workflow Name | Description
--- | ---
Unregister virtual machine | Removes an existing virtual machine from the inventory.
Upgrade virtual machine hardware (force if required) | Upgrades the virtual machine hardware to the latest revision that the host supports. This workflow forces the upgrade to continue, even if VMware Tools is out of date. If the VMware Tools is out of date, forcing the upgrade to continue reverts the guest network settings to the default settings. To avoid this situation, upgrade VMware Tools before running the workflow.
Upgrade virtual machine | Upgrades the virtual hardware to the latest revision that the host supports. An input parameter allows a forced upgrade even if VMware Tools is out of date.
Wait for task and answer virtual machine question | Waits for a vCenter Server task to complete or for the virtual machine to ask a question. If the virtual machine requires an answer, accepts user input and answers the question.

### Clone Workflows

With clone workflows, you can clone virtual machines with or without customizing the virtual machine properties.

You can access the clone workflows from **Library > vCenter > Virtual Machine management > Clone** in the **Workflows** view of the Orchestrator client.

| Workflow Name | Description |
--- | ---|
Clone virtual machine from properties | Clones virtual machines by using properties as input parameters. |
Clone virtual machine, no customization | Clones a virtual machine without changing anything except the virtual machine UUID. |
Customize virtual machine from properties | Customizes a virtual machine by using properties as input parameters. |

### Linked Clone Workflows

With the linked clone workflows, you can perform linked clone operations such as restoring a virtual machine from a linked clone, creating a linked clone, or others.

You can access the linked clone workflows from **Library > vCenter > Virtual Machine management > Clone > Linked Clone** folder and its subfolders in the **Workflows** view of the Orchestrator client.

| Workflow Name | Description |
--- | ---|
Restore virtual machine from linked clone | Removes a virtual machine from a linked clone setup. |
Set up virtual machine for linked clone | Prepares a virtual machine to be link cloned. |
Create a linked clone of a Linux machine with multiple NICs | Creates a linked clone of a Linux virtual machine, performs the guest operating system customization, and configures up to four virtual network cards. |
Create a linked clone of a Linux machine with a single NIC | Creates a linked clone of a Linux virtual machine, performs the guest operating system customization, and configures one virtual network card. |
Create a linked clone of a Windows machine with multiple NICs and credential | Creates a linked clone of a Windows virtual machine and performs the guest operating system customization. Configures up to four virtual network cards and a local administrator user account. |
<table>
<thead>
<tr>
<th>Workflow Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Create a linked clone of a Windows machine with a single NIC and credential</td>
<td>Creates a linked clone of a Windows virtual machine and performs the guest operating system customization. Configures one virtual network card and a local administrator user account.</td>
</tr>
<tr>
<td>Create a linked clone with no customization</td>
<td>Creates the specified number of linked clones of a virtual machine.</td>
</tr>
</tbody>
</table>

**Linux Customization Clone Workflows**

With Linux customization workflows, you can clone a Linux virtual machine and customize the guest operating system.

You can access the Linux customization clone workflows from **Library > vCenter > Virtual Machine management > Clone > Linux Customization** in the **Workflows** view of the Orchestrator client.

<table>
<thead>
<tr>
<th>Workflow Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Clone a Linux machine with multiple NICs</td>
<td>Clones a Linux virtual machine, performs the guest operating system customization, and configures up to four virtual network cards.</td>
</tr>
<tr>
<td>Clone a Linux machine with a single NIC</td>
<td>Clones a Linux virtual machine, performs the guest operating system customization, and configures one virtual network card.</td>
</tr>
</tbody>
</table>

**Tools Clone Workflows**

With the tools clone workflows, you can obtain customization information about the operating system of the virtual machine, information required to update a virtual device, and others.

You can access the tools clone workflows from **Library > vCenter > Virtual Machine management > Clone > Tools** in the **Workflows** view of the Orchestrator client.

<table>
<thead>
<tr>
<th>Workflow Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Get a virtual Ethernet card to change the network</td>
<td>Returns a new ethernet card to update a virtual device. Contains only the device key of the given virtual device and the new network.</td>
</tr>
<tr>
<td>Get Linux customization</td>
<td>Returns the Linux customization preparation.</td>
</tr>
<tr>
<td>Get multiple virtual Ethernet card device changes</td>
<td>Returns an array of VirtualDeviceConfigSpec objects for add and remove operations on VirtualEthernetCard objects.</td>
</tr>
<tr>
<td>Get NIC setting map</td>
<td>Returns the setting map for a virtual network card by using VimAdapterMapping.</td>
</tr>
<tr>
<td>Get Windows customization for Sysprep with credentials</td>
<td>Returns customization information about the Microsoft Sysprep process, with credentials. Workflows for cloning Windows virtual machines use this workflow.</td>
</tr>
<tr>
<td>Get Windows customization for Sysprep with Unattended.txt</td>
<td>Returns customization information about the Microsoft Sysprep process by using an Unattended.txt file. Workflows for cloning Windows virtual machines use this workflow.</td>
</tr>
<tr>
<td>Get Windows customization for Sysprep</td>
<td>Returns customization information about the Microsoft Sysprep process. Workflows for cloning Windows virtual machines use this workflow.</td>
</tr>
</tbody>
</table>
Windows Customization Clone Workflows

With the Windows customization clone workflows, you can clone Windows virtual machines and customize the guest operating system.

You can access the Windows customization clone workflows from Library > vCenter > Virtual Machine management > Clone > Windows Customization folder and its subfolder in the Workflows view of the Orchestrator client.

<table>
<thead>
<tr>
<th>Workflow Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Customize a Windows machine with single NIC and credential</td>
<td>Performs guest operating system customization, configures one virtual network card and a local administrator user account on a Windows virtual machine.</td>
</tr>
<tr>
<td>Clone a thin provisioned Windows machine with single NIC and credential</td>
<td>Clones a Windows virtual machine performing the guest operating system customization. Configures one virtual network card and a local administrator user account. Sysprep tools must be available on vCenter Server.</td>
</tr>
<tr>
<td>Clone a Windows machine Sysprep with single NIC and credential</td>
<td>Clones a Windows virtual machine performing the guest operating system customization. Configures one virtual network card and a local administrator user account. Sysprep tools must be available on vCenter Server.</td>
</tr>
<tr>
<td>Clone a Windows machine with multiple NICs and credential</td>
<td>Clones a Windows virtual machine performing the guest operating system customization. Configures the local administrator user account and up to four virtual network cards. Sysprep tools must be available on the vCenter Server system.</td>
</tr>
<tr>
<td>Clone a Windows machine with single NIC</td>
<td>Clones a Windows virtual machine performing the guest operating system customization and configures one virtual network card. Sysprep tools must be available on the vCenter Server system.</td>
</tr>
<tr>
<td>Clone a Windows machine with single NIC and credential</td>
<td>Clones a Windows virtual machine performing the guest operating system customization. Configures one virtual network card and a local administrator user account. Sysprep tools must be available on the vCenter Server system.</td>
</tr>
</tbody>
</table>

Device Management Workflows

With the device management workflows, you can manage the devices that are connected to a virtual machine or to a host datastore.

You can access the device management workflows from Library > vCenter > Virtual Machine management > Device Management in the Workflows view of the Orchestrator client.

<table>
<thead>
<tr>
<th>Workflow Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Add CD-ROM</td>
<td>Adds a virtual CD-ROM to a virtual machine. If the virtual machine has no IDE controller, the workflow creates one.</td>
</tr>
<tr>
<td>Add disk</td>
<td>Adds a virtual disk to a virtual machine.</td>
</tr>
<tr>
<td>Change RAM</td>
<td>Changes the amount of RAM of a virtual machine.</td>
</tr>
<tr>
<td>Convert disks to thin provisioning</td>
<td>Converts thick-provisioned disks of virtual machines to thin-provisioned disks.</td>
</tr>
<tr>
<td>Convert independent disks</td>
<td>Converts all independent virtual machine disks to normal disks by removing the independent flag from the disks.</td>
</tr>
<tr>
<td>Disconnect all detachable devices from a running virtual machine</td>
<td>Disconnects floppy disks, CD-ROM drives, parallel ports, and serial ports from a running virtual machine.</td>
</tr>
</tbody>
</table>
### Move and Migrate Workflows

With the move and migrate workflows, you can migrate virtual machines.

You can access the move and migrate workflows from **Library > vCenter > Virtual Machine management > Move and Migrate** in the **Workflows** view of the Orchestrator client.

<table>
<thead>
<tr>
<th>Workflow Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Mass migrate virtual machines with storage vMotion</td>
<td>Uses Storage vMotion to migrate a single virtual machine, a selection of virtual machines, or all available virtual machines.</td>
</tr>
<tr>
<td>Mass migrate virtual machines with vMotion</td>
<td>Uses vMotion, Storage vMotion, or both vMotion and Storage vMotion to migrate a single virtual machine, a selection of virtual machines, or all available virtual machines.</td>
</tr>
<tr>
<td>Migrate virtual machine with vMotion</td>
<td>Migrates a virtual machine from one host to another by using the <code>MigrateVM_Task</code> operation from the vSphere API.</td>
</tr>
<tr>
<td>Move virtual machine to another vCenter Server system</td>
<td>Moves a list of virtual machines to another vCenter Server system.</td>
</tr>
<tr>
<td>Quick migrate multiple virtual machines</td>
<td>Suspends the virtual machines if they are powered on and migrates them to another host using the same storage.</td>
</tr>
<tr>
<td>Quick migrate virtual machine</td>
<td>Suspends the virtual machine if it is powered on and migrates it to another host using the same storage.</td>
</tr>
<tr>
<td>Relocate virtual machine disks</td>
<td>Relocates virtual machine disks to another host or datastore while the virtual machine is powered off by using the <code>RelocateVM_Task</code> operation from the vSphere API.</td>
</tr>
</tbody>
</table>

### Other Workflows

With the workflows from the Others category, you can enable and disable Fault Tolerance (FT), extract virtual machine information, and find orphaned virtual machines.

You can access these workflows from **Library > vCenter > Virtual Machine management > Others** in the **Workflows** view of the Orchestrator client.

<table>
<thead>
<tr>
<th>Workflow Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Disable FT</td>
<td>Disables Fault Tolerance for a specified virtual machine.</td>
</tr>
<tr>
<td>Enable FT</td>
<td>Enables Fault Tolerance for a specified virtual machine.</td>
</tr>
<tr>
<td>Extract virtual machine information</td>
<td>Returns the virtual machine folder, host system, resource pool, compute resource, datastore, hard drive sizes, CPU and memory, network, and IP address for a given virtual machine. Might require VMware Tools.</td>
</tr>
<tr>
<td>Find orphaned virtual machines</td>
<td>Lists all virtual machines in an orphaned state in the Orchestrator inventory. Lists the VMDK and VMTX files for all datastores in the Orchestrator inventory that have no association with any virtual machines in the Orchestrator inventory. Sends the lists by email (optional).</td>
</tr>
</tbody>
</table>
Workflow Name | Description
--- | ---
Get Virtual Machine by Name and BIOS UUID | Searches virtual machines by name and then filters the result with particular universally unique identifier (UUID) in order to identify a unique virtual machine.

**Note** This workflow is needed when DynamicOps calls vRealize Orchestrator workflows having input parameters of `VC:VirtualMachine` type in order to make the correspondence between a particular DynamicOps and vRealize Orchestrator virtual machine.

Get Virtual Machine by Name and UUID | Searches virtual machines by name and then filters the result with particular universally unique identifier (UUID) in order to identify a unique virtual machine.

**Note** This workflow is needed when DynamicOps calls vRealize Orchestrator workflows having input parameters of `VC:VirtualMachine` type in order to make the correspondence between a particular DynamicOps and vRealize Orchestrator virtual machine.

Get Virtual Machine UUID | Searches virtual machines by name and then filters the result with particular universally unique identifier (UUID) in order to identify a unique virtual machine.

**Note** This workflow is needed when DynamicOps calls vRealize Orchestrator workflows having input parameters of `VC:VirtualMachine` type in order to make the correspondence between a particular DynamicOps and vRealize Orchestrator virtual machine.

### Power Management Workflows

With the power management workflows, you can power on and off virtual machines, reboot the guest operating system of a virtual machine, suspend a virtual machine, and others.

You can access the power management workflows from **Library > vCenter > Virtual Machine management > Power Management** in the **Workflows** view of the Orchestrator client.

<table>
<thead>
<tr>
<th>Workflow Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Power off virtual machine and wait</td>
<td>Powers off a virtual machine and waits for the process to complete.</td>
</tr>
<tr>
<td>Reboot guest OS</td>
<td>Reboots the guest operating system of the virtual machine. Does not reset nonpersistent virtual machines. VMware Tools must be running.</td>
</tr>
<tr>
<td>Reset virtual machine and wait</td>
<td>Resets a virtual machine and waits for the process to complete.</td>
</tr>
<tr>
<td>Resume virtual machine and wait</td>
<td>Resumes a suspended virtual machine and waits for the process to complete.</td>
</tr>
<tr>
<td>Set guest OS to standby mode</td>
<td>Sets the guest operating system to standby mode. VMware Tools must be running.</td>
</tr>
<tr>
<td>Shut down and delete virtual machine</td>
<td>Shuts down a virtual machine and deletes it from the inventory and disk.</td>
</tr>
<tr>
<td>Shut down guest OS and wait</td>
<td>Shuts down a guest operating system and waits for the process to complete.</td>
</tr>
<tr>
<td>Start virtual machine and wait</td>
<td>Starts a virtual machine and waits for VMware Tools to start.</td>
</tr>
<tr>
<td>Suspend virtual machine and wait</td>
<td>Suspends a virtual machine and waits for the process to complete.</td>
</tr>
</tbody>
</table>

### Snapshot Workflows

With snapshot workflows, you can perform snapshot-related operations.

You can access the snapshot workflows from **Library > vCenter > Virtual Machine management > Snapshot** in the **Workflows** view of the Orchestrator client.
### VMware Tools Workflows

With VMware Tools workflows, you can perform VMware Tools-related tasks on virtual machines.

You can access the VMware Tools workflows from **Library > vCenter > Virtual Machine management > VMware Tools** in the **Workflows** view of the Orchestrator client.

<table>
<thead>
<tr>
<th>Workflow Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Mount VMware Tools installer</td>
<td>Mounts the VMware Tools installer on the virtual CD-ROM.</td>
</tr>
<tr>
<td>Set console screen resolution</td>
<td>Sets the resolution of the console window. The virtual machine must be powered on.</td>
</tr>
<tr>
<td>Turn on time synchronization</td>
<td>Turns on time synchronization between the virtual machine and the ESXi server in VMware Tools.</td>
</tr>
<tr>
<td>Unmount VMware Tools installer</td>
<td>Unmounts the VMware Tools CD-ROM.</td>
</tr>
<tr>
<td>Upgrade VMware Tools</td>
<td>Upgrades VMware Tools on a virtual machine.</td>
</tr>
<tr>
<td>Upgrade VMware Tools at next reboot</td>
<td>Upgrades VMware Tools on a virtual machine without performing an automatic reboot.</td>
</tr>
</tbody>
</table>
Using the Configuration Plug-In

In addition to configuring Orchestrator by using Control Center, you can modify the Orchestrator server configuration settings by running workflows from the Configuration plug-in.

With the Configuration plug-in, you can configure and manage the Orchestrator server keystores and trusted certificates.

This chapter includes the following topics:

- Access the Configuration Plug-In Workflow Library
- Configuration Plug-In Workflow Library

Access the Configuration Plug-In Workflow Library

You must use the Orchestrator client to access the elements from the Configuration plug-in workflow library.

Procedure

1. From the drop-down menu in the Orchestrator client, select Run or Design.
2. Click the Workflows view.
3. Expand the hierarchical list to Library > Configuration.

What to do next

Review the workflow library.

Configuration Plug-In Workflow Library

The Configuration plug-in workflow library contains workflows that you can use to run automated processes related to the configuration of vRealize Orchestrator.

SSL Trust Manager Workflows

The SSL Trust Manager category contains workflows that you can use for deleting and importing SSL certificates.

You access these workflows from Library > Configuration > SSL Trust Manager workflows in the Workflows view of the Orchestrator client.
### Workflow Name

<table>
<thead>
<tr>
<th>Workflow Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Delete a trusted certificate</td>
<td>Deletes an SSL certificate from the server trust store.</td>
</tr>
<tr>
<td>Import certificate from URL</td>
<td>Imports an SSL certificate from a URL into the server trust store.</td>
</tr>
<tr>
<td>Import certificate from URL using proxy server</td>
<td>Imports an SSL certificate from a URL into the server trust store.</td>
</tr>
<tr>
<td>Import certificate from URL with certificate alias</td>
<td>Imports an SSL certificate from a URL into the server trust store.</td>
</tr>
<tr>
<td>Import trusted certificate from a file</td>
<td>Imports an SSL certificate from a file into the server trust store.</td>
</tr>
</tbody>
</table>

## Keystore Workflows

You access the Keystore configuration workflows from **Library > Configuration > Keystores** in the **Workflows** view of the Orchestrator client.

<table>
<thead>
<tr>
<th>Workflow Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Add certificate</td>
<td>Adds a certificate to a keystore.</td>
</tr>
<tr>
<td>Add key</td>
<td>Adds a key.</td>
</tr>
<tr>
<td>Create a keystore</td>
<td>Creates a new keystore.</td>
</tr>
<tr>
<td>Delete a keystore</td>
<td>Deletes a keystore.</td>
</tr>
<tr>
<td>Delete certificate</td>
<td>Deletes a certificate from a keystore.</td>
</tr>
<tr>
<td>Delete entry</td>
<td>Deletes an entry.</td>
</tr>
<tr>
<td>Delete key</td>
<td>Deletes a key.</td>
</tr>
</tbody>
</table>
Using the vCO Library Plug-In

You can use the vCO Library plug-in workflows as templates for customization and automation of client processes, and to troubleshoot Orchestrator.

vCO Library Plug-In Workflows

The vCO Library plug-in provides workflows in the Locking, Orchestrator, and Troubleshooting workflow categories.

Locking Workflows

You access these workflows from Library > Locking in the Workflows view of the Orchestrator client.

<table>
<thead>
<tr>
<th>Workflow Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Display all locks</td>
<td>Shows all locks.</td>
</tr>
<tr>
<td>Locking test</td>
<td>A test workflow that creates a lock.</td>
</tr>
<tr>
<td>Locking test (x5)</td>
<td>A test workflow that creates five locks.</td>
</tr>
<tr>
<td>Release all locks</td>
<td>Releases all locks.</td>
</tr>
</tbody>
</table>

Orchestrator Task Workflows

You access these workflows from Library > Orchestrator > Tasks in the Workflows view of the Orchestrator client.

<table>
<thead>
<tr>
<th>Workflow Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Create recurrent task</td>
<td>Creates a recurrent task and returns the newly created task.</td>
</tr>
<tr>
<td>Create task</td>
<td>Schedules a workflow to run at a later time and date, as a task.</td>
</tr>
</tbody>
</table>

Orchestrator Workflows

You access these workflows from Library > Orchestrator > Workflows in the Workflows view of the Orchestrator client.
### Troubleshooting Workflows

You access these workflows from **Library > Troubleshooting** in the **Workflows** view of the Orchestrator client.

<table>
<thead>
<tr>
<th>Workflow Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Export logs and application settings</td>
<td>Generates a ZIP archive of troubleshooting information that contains configuration files and server, configuration, wrapper, and installation log files. The output directory must exist and Orchestrator must have writing permissions.</td>
</tr>
</tbody>
</table>

### Tagging Workflows

You access these workflows from **Library > Tagging** in the **Workflows** view of the Orchestrator client.

<table>
<thead>
<tr>
<th>Workflow Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Find objects by tag</td>
<td>Finds objects by the tags assigned to them. You provide the names and values of the tags and the workflow returns a list of the objects to which these tags apply.</td>
</tr>
<tr>
<td>List workflow tags</td>
<td>Lists the tags assigned to the workflow you specified as an input parameter.</td>
</tr>
<tr>
<td>Tagging example</td>
<td>Demonstrates workflow tagging.</td>
</tr>
<tr>
<td>Tag workflow</td>
<td>Assigns a tag to a workflow. You must specify the workflow you want to tag and the tag name and value.</td>
</tr>
<tr>
<td>Untag workflow</td>
<td>Removes a tag from a workflow. You must specify the workflow you want to untag and the tag you want to remove from the specified workflow.</td>
</tr>
</tbody>
</table>
Using the SQL Plug-In

You can use the API that the SQL plug-in provides to implement connectivity to SQL databases and other tabular data sources, such as spreadsheets or flat files.

The SQL plug-in API which is based on JDBC, provides a call-level API for SQL-based database access. The SQL plugin also provides sample workflows that demonstrate how to use the API in workflows.

This chapter includes the following topics:

- Configuring the SQL Plug-In
- Running the SQL Sample Workflows
- Using the SQL Plug-In Standard Workflows

Configuring the SQL Plug-In

You can use the workflows included in the SQL plug-in and run them from the Orchestrator client to configure the SQL plug-in and to add, update, or remove a database.

SQL Plug-In Configuration Workflows

The Configuration workflow category of the SQL plug-in contains workflows that allow you to manage databases and database tables.

You can access these workflows from Library > SQL > Configuration in the Workflows view of the Orchestrator client.

<table>
<thead>
<tr>
<th>Workflow Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Add a database</td>
<td>Adds a database object to the Database plug-in inventory.</td>
</tr>
<tr>
<td>Add tables to a database</td>
<td>Adds database tables to a database in the Database plug-in inventory.</td>
</tr>
<tr>
<td>Remove a database</td>
<td>Removes a database object from the Database plug-in inventory.</td>
</tr>
<tr>
<td>Remove a table from a database</td>
<td>Removes a database table from a database in the Database plug-in inventory.</td>
</tr>
<tr>
<td>Update a database</td>
<td>Updates the configuration of a database object in the Database plug-in inventory.</td>
</tr>
<tr>
<td>Validate a database</td>
<td>Validates a database in the Database plug-in inventory.</td>
</tr>
</tbody>
</table>
Add a Database

You can run a workflow to add a database to the Orchestrator server and configure the host connection parameters.

When you add a database that requires a secure connection, you must import the database SSL certificate. You can import the SSL certificate under the Trusted Certificates tab in Control Center.

Procedure

1. Log in to the Orchestrator client as an administrator.
2. Click the Workflows view in the Orchestrator client.
3. In the workflows hierarchical list, expand Library > SQL > Configuration and navigate to the Add a database workflow.
4. Right-click the Add a database workflow and select Start workflow.
5. In the Name text box, type the name of the database.
6. Select the type of the database.
7. In the Connection URL text box, type the address of the database.
8. Select the session mode that the plug-in uses to connect to the database.
9. Click Submit to run the workflow.

After the workflow runs successfully, the database and all tables that belong to it appear in the Inventory view.

<table>
<thead>
<tr>
<th>Database Type</th>
<th>Syntax</th>
</tr>
</thead>
<tbody>
<tr>
<td>Oracle</td>
<td>jdbc:oracle:thin:@database_url:port_number:SID</td>
</tr>
<tr>
<td>Microsoft SQL (with SQL authentication)</td>
<td>jdbc:jtds:sqlserver://database_url:port_number/database_name</td>
</tr>
<tr>
<td>Microsoft SQL (with Windows account authentication)</td>
<td>jdbc:jtds:sqlserver://database_url:port_number/database_name;useNTLMv2=true;domain=domain_name</td>
</tr>
<tr>
<td>PostgreSQL</td>
<td>jdbc:postgresql://database_url:port_number/database_name</td>
</tr>
<tr>
<td>MySQL</td>
<td>jdbc:mysql://database_url:port_number/database_name</td>
</tr>
</tbody>
</table>

Option | Description                                                                 |
--------|-----------------------------------------------------------------------------|
Shared Session | The plug-in uses shared credentials to connect to the database. You must provide the database credentials for the shared session. |
Session Per User | The Orchestrator client retrieves credentials from the user who is logged in. |

**Note** To use session per user mode, you must authenticate by using a user name only. You should not use domain\user or user@domain for authentication.
Add Tables to a Database
You can run a workflow to add tables to a database that is in the Database plug-in inventory.

Prerequisites
- Verify that you are logged in to the Orchestrator client as an administrator.
- Verify that you have a connection to a database from the Inventory view.

Procedure
1. Click the Workflows view in the Orchestrator client.
2. In the workflows hierarchical list, select Library > SQL > Configuration and navigate to the Add tables to a database workflow.
3. Right-click the Add tables to a database workflow and select Start workflow.
4. Select a database to which to add tables.
5. Select the tables that you want to add.
6. Click Submit to run the workflow.

After the workflow runs successfully, the added database tables appear in the Inventory view of the Orchestrator client.

Update a Database
You can run a workflow to update the configuration of a database that is in the plug-in's inventory.

Procedure
1. Log in to the Orchestrator client as an administrator.
2. Click the Workflows view in the Orchestrator client.
3. In the Workflows hierarchical list, select Library > SQL > Configuration and navigate to the Update a database workflow.
4. Right-click the Update a database workflow and select Start workflow.
5. Select a database that you want to update.
6. In the Name text box, type the new name of the database.
   The database appears in the Inventory view with the name that you specify.
7. Select the type of the database.
8. In the Connection URL text box, type the new address of the database.
9 Select the session mode that the plug-in uses to connect to the database.

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Shared Session</td>
<td>The plug-in uses shared credentials to connect to the database. You must provide the database credentials for the shared session.</td>
</tr>
<tr>
<td>Session Per User</td>
<td>The Orchestrator client retrieves credentials from the user who is logged in.</td>
</tr>
</tbody>
</table>

**Note** To use session per user mode, you must authenticate by using a user name only. You should not use domain\user or user@domain for authentication.

10 Click **Submit** to run the workflow.

## Running the SQL Sample Workflows

You can run the SQL plug-in workflows to perform JDBC operations such as generating a JDBC URL, testing a JDBC connection, and managing rows in JDBC tables. You can also run the SQL plug-in workflows to manage databases and database tables, as well as to run SQL operations.

### Generate a JDBC URL

You can run a workflow from the Orchestrator client to generate a JDBC connection URL.

**Prerequisites**

Verify that the user account you are logged in with has the necessary permissions to run JDBC workflows.

**Procedure**

1. Click the **Workflows** view in the Orchestrator client.
2. In the workflows hierarchical list, open **Library > JDBC** to navigate to the JDBC URL generator workflow.
3. Right-click the JDBC URL generator workflow and select **Start workflow**.
4. Select the type of database for which to generate a URL.
   **Note** If you use a Microsoft database, you might have to click **Next** and to provide the database instance name and database user domain name.
5. Provide the required information to generate a database URL.
   a. Type a database server name or IP address.
   b. Type a database name.
   c. (Optional) Type a database port number.
      If you do not specify a port number, the workflow uses a default port number.
   d. Type a user name to access the database.
   e. Type a password to access the database.
6 Click Submit to run the workflow.

**Test a JDBC Connection**

You can run a workflow from the Orchestrator client to test the connection to a database.

**Prerequisites**

Verify that the user account you are logged in with has the necessary permissions to run JDBC workflows.

**Procedure**

1 Click the Workflows view in the Orchestrator client.
2 In the workflows hierarchical list, open Library > JDBC > JDBC Examples to navigate to the JDBC connection example workflow.
3 Right-click the JDBC connection example workflow and select Start workflow.
4 Provide the required information to test a database connection.
   a Type a user name to access the database.
   b Type the URL to test.
   c Type a password to access the database.
5 Click Submit to run the workflow.

**Create a Table by Using JDBC**

You can run a workflow from the Orchestrator client to create a database.

**Prerequisites**

Verify that the user account you are logged in with has the necessary permissions to run JDBC workflows.

**Procedure**

1 Click the Workflows view in the Orchestrator client.
2 In the workflows hierarchical list, open Library > JDBC > JDBC Examples to navigate to the JDBC create table example workflow.
3 Right-click the JDBC create table example workflow and select Start workflow.
4 Provide the required information, and click Next.
   a Type a password to access the database.
   b Type a database connection URL.
   c Type a user name to access the database.
5 Type an SQL create statement.
   An example syntax is:
   
   ```sql
   CREATE TABLE "table_name"
   ("column1" "data_type_for_column1",
    "column2" "data_type_for_column2")
   ```

6 Click **Submit** to run the workflow.

**Insert a Row into a JDBC Table**

You can run a workflow from the Orchestrator client to test the insertion of a row into a JDBC table.

**Prerequisites**

Verify that the user account you are logged in with has the necessary permissions to run JDBC workflows.

**Procedure**

1 Click the **Workflows** view in the Orchestrator client.

2 In the workflows hierarchical list, open **Library > JDBC > JDBC Examples** to navigate to the JDBC insert into table example workflow.

3 Right-click the JDBC insert into table example workflow and select **Start workflow**.

4 Provide the required information, and click **Next**.
   a Type a database connection URL.
   b Type a user name to access the database.
   c Type a password to access the database.

5 Type an SQL insert statement, and click **Next**.
   An example syntax is:
   
   ```sql
   INSERT INTO "table_name" ("column1", "column2")
   VALUES ("value1", "value2")
   ```

6 Type the values to insert into the row.

7 Click **Submit** to run the workflow.

**Select Rows from a JDBC Table**

You can run a workflow from the Orchestrator client to select rows from a JDBC table.

**Prerequisites**

Verify that the user account you are logged in with has the necessary permissions to run JDBC workflows.
Procedure

1. Click the **Workflows** view in the Orchestrator client.

2. In the workflows hierarchical list, open **Library > JDBC > JDBC Examples** to navigate to the JDBC select from table example workflow.

3. Right-click the JDBC select from table example workflow and select **Start workflow**.

4. Provide the required information, and click **Next**.
   - Type a database connection URL.
   - Type a user name to access the database.
   - Type a password to access the database.

5. Type an SQL select statement.
   An example syntax is:
   
   ```sql
   SELECT * FROM "table_name"
   ```

6. Click **Submit** to run the workflow.

**Delete an Entry from a JDBC Table**

You can run a workflow from the Orchestrator client to test the deletion of an entry from a JDBC table.

**Prerequisites**

Verify that the user account you are logged in with has the necessary permissions to run JDBC workflows.

**Procedure**

1. Click the **Workflows** view in the Orchestrator client.

2. In the workflows hierarchical list, open **Library > JDBC > JDBC Examples** to navigate to the JDBC delete entry from table example workflow.

3. Right-click the JDBC delete entry from table example workflow and select **Start workflow**.

4. Provide the required information, and click **Next**.
   - Type the first name of the user entry to be deleted.
   - Type a user name to access the database.
   - Type a JDBC connection URL.
   - Type the last name of the user entry to be deleted.
   - Type a password to access the database.
5 Type an SQL delete statement.
   An example syntax is:

   ```sql
   DELETE FROM "table_name" where ("column1" = ?, "column2" = ?)
   ```

6 Click Submit to run the workflow.

**Delete All Entries from a JDBC Table**

You can run a workflow from the Orchestrator client to delete all entries from a JDBC table.

**Prerequisites**

Verify that the user account you are logged in with has the necessary permissions to run JDBC workflows.

**Procedure**

1 Click the **Workflows** view in the Orchestrator client.

2 In the workflows hierarchical list, open **Library > JDBC > JDBC Examples** to navigate to the JDBC delete all from table example workflow.

3 Right-click the JDBC delete all from table example workflow and select **Start workflow**.

4 Provide the required information, and click **Next**.
   a Type a database connection URL.
   b Type a user name to access the database.
   c Type a password to access the database.

5 Type an SQL delete statement.
   An example syntax is:

   ```sql
   DELETE FROM "table_name"
   ```

6 Click Submit to run the workflow.

**Drop a JDBC Table**

You can run a workflow from the Orchestrator client to test the dropping of a JDBC table.

**Prerequisites**

Verify that the user account you are logged in with has the necessary permissions to run JDBC workflows.

**Procedure**

1 Click the **Workflows** view in the Orchestrator client.
2 In the workflows hierarchical list, open **Library > JDBC > JDBC Examples** to navigate to the JDBC drop table example workflow.

3 Right-click the JDBC drop table example workflow and select **Start workflow**.

4 Provide the required information, and click **Next**.
   a Type a password to access the database.
   b Type a database connection URL.
   c Type a user name to access the database.

5 Type an SQL drop statement.
   An example syntax is:
   
   ```sql
   DROP TABLE "table_name"
   ```

6 Click **Submit** to run the workflow.

**Run a Complete JDBC Cycle**

You can run a workflow from the Orchestrator client to test all JDBC example workflows in one full cycle.

**Prerequisites**

Verify that the user account you are logged in with has the necessary permissions to run JDBC workflows.

**Procedure**

1 Click the **Workflows** view in the Orchestrator client.

2 In the workflows hierarchical list, open **Library > JDBC > JDBC Examples** to navigate to the Full JDBC cycle example workflow.

3 Right-click the Full JDBC cycle example workflow and select **Start workflow**.

4 Provide the required information, and click **Next**.
   a Type a database connection URL.
   b Type a user name to access the database.
   c Type a password to access the database.

5 Type the values to be used as entries in the database.

6 Click **Submit** to run the workflow.

**Using the SQL Plug-In Standard Workflows**

You can use the SQL workflows to run SQL operations.
SQL Plug-In Workflow Library

You can run the SQL plug-in workflows to manage databases and database tables and to run SQL operations.

You can access the database configuration workflows from Library > SQL > Configuration in the Workflows view of the Orchestrator client.

<table>
<thead>
<tr>
<th>Workflow Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Add a database</td>
<td>Adds a database object to the plug-in inventory.</td>
</tr>
<tr>
<td>Add tables to a database</td>
<td>Adds database tables to a database in the plug-in inventory.</td>
</tr>
<tr>
<td>Remove a database</td>
<td>Removes a database object from the plug-in inventory.</td>
</tr>
<tr>
<td>Remove a table from a database</td>
<td>Removes a database table from a database in the plug-in inventory.</td>
</tr>
<tr>
<td>Update a database</td>
<td>Updates the configuration of a database object in the plug-in inventory.</td>
</tr>
<tr>
<td>Validate a database</td>
<td>Validates a database in the plug-in inventory.</td>
</tr>
</tbody>
</table>

You can access the SQL operations workflows from Library > SQL in the Workflows view of the Orchestrator client.

<table>
<thead>
<tr>
<th>Workflow Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Execute a custom query on a database</td>
<td>Runs a custom query on a specified database and returns the number of affected rows. You can run the workflow to update, delete, insert, and write queries.</td>
</tr>
<tr>
<td>Generate CRUD workflows for a table</td>
<td>Generates Create, Read, Update, and Delete workflows for a particular table.</td>
</tr>
<tr>
<td>Read a custom query from a database</td>
<td>Runs a custom query on a specified database and returns the result in an array of properties. You can run the workflow to select and read queries.</td>
</tr>
</tbody>
</table>

Generate CRUD Workflows for a Table

You can run a workflow to generate Create, Read, Update, and Delete workflows for a particular table.

Prerequisites

- Verify that you are logged in to the Orchestrator client as an administrator.
- Verify that you have a connection to a database from the Inventory view.

Procedure

1. Click the Workflows view in the Orchestrator client.
2. In the workflows hierarchical list, select Library > SQL and navigate to the Generate CRUD workflows for a table workflow.
3. Right-click the Generate CRUD workflows for a table workflow and select Start workflow.
4. Select a table for which to generate the workflows.
5. Select the workflow folder in which to generate the workflows.
6 Select whether to overwrite any existing workflows.

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Yes</td>
<td>The generated workflows overwrite existing workflows with the same name.</td>
</tr>
<tr>
<td>No</td>
<td>New workflows are not generated if workflows with the same name exist in the folder.</td>
</tr>
</tbody>
</table>

7 (Optional) Select columns that should not be populated.

You cannot edit the selected columns with the generated CRUD workflows.

8 Click **Submit** to run the workflow.

After the workflow runs successfully, the CRUD workflows appear in the selected workflow folder.

**What to do next**

You can run the generated workflows on the selected database table.
Using the SSH Plug-In

You can use the SSH plug-in workflows to run SSH commands on a remote host that supports SSH and transfer files between an Orchestrator server and a remote host through a secure connection.

This chapter includes the following topics:
- Configure the SSH Plug-In
- Running the SSH Plug-In Sample Workflows

Configure the SSH Plug-In

You can set up the SSH plug-in to ensure encrypted connections.

Procedure

1. Log in to the Orchestrator client as an administrator.
2. Click the Workflows view in the Orchestrator client.
3. In the workflows hierarchical list, expand Library > SSH and navigate to the Add SSH host workflow.
4. Right-click the Configure mail workflow and select Start workflow.
5. In the Host name text box, enter the name of the host that you want to access with SSH through Orchestrator.
6. Enter the target port. The default SSH port is 22.
   The host is added to the list of SSH connections.
7. (Optional) Configure an entry path on the server.
   a. Click New root folder.
   b. Enter the new path and click Insert value.
8. Enter the user name for a user who has the necessary permissions to run SSH commands.
9. Select the authentication type.

<table>
<thead>
<tr>
<th>Option</th>
<th>Action</th>
</tr>
</thead>
<tbody>
<tr>
<td>Yes</td>
<td>Enter a password to use password authentication.</td>
</tr>
<tr>
<td>No</td>
<td>Enter the path to the private key and the private key passphrase to use key authentication.</td>
</tr>
</tbody>
</table>
10 Click Submit to run the workflow.

The SSH host is available in the Inventory view of the Orchestrator client.

**Configuration Workflows**

The Configuration category of the SSH plug-in contains workflows that let you manage the connections between Orchestrator and SSH hosts.

You can access these workflows from Library > SSH > Configuration in the Workflows view of the Orchestrator client.

<table>
<thead>
<tr>
<th>Workflow Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Add a Root Folder to SSH Host</td>
<td>Adds a root folder to an existing connection to an SSH host.</td>
</tr>
<tr>
<td>Add SSH Host</td>
<td>Adds a new connection to an SSH host to the existing configuration.</td>
</tr>
<tr>
<td>Remove a Root Folder from SSH Host</td>
<td>Removes a root folder from an existing connection to an SSH host.</td>
</tr>
<tr>
<td>Remove SSH Host</td>
<td>Removes an existing connection to an SSH host from the existing configuration.</td>
</tr>
<tr>
<td>Update SSH Host</td>
<td>Updates an existing connection to an SSH host.</td>
</tr>
</tbody>
</table>

**Running the SSH Plug-In Sample Workflows**

You can run the SSH plug-in sample workflows from the Orchestrator client to test the connection between the Orchestrator server and the SSH host.

- **Generate a Key Pair**
  You can run a workflow from the Orchestrator client to generate a key pair. You can use the key pair to connect to an SSH host without a password.

- **Change the Key Pair Passphrase**
  You can run a workflow from the Orchestrator client to change the passphrase for the key pair that you generated most recently.

- **Register an Orchestrator Public Key on an SSH Host**
  You can use a public key instead of a password. To register an Orchestrator public key on an SSH host, you can run a workflow from the Orchestrator client.

- **Run an SSH Command**
  You can run a workflow from the Orchestrator client to run SSH commands on a remote SSH server.

- **Copy a File from an SSH Host**
  You can run a workflow on the Orchestrator client to copy files from an SSH host to the Orchestrator server.

- **Copy a File to an SSH Host**
  You can run a workflow from the Orchestrator client to copy files from the Orchestrator server to an SSH host.
Generate a Key Pair

You can run a workflow from the Orchestrator client to generate a key pair. You can use the key pair to connect to an SSH host without a password.

A key pair consists of a public key and a private key. Orchestrator can use the private key to connect to the public key on an SSH host. You can use a passphrase to improve security.

**Caution**  All Orchestrator users with the right set of privileges can read, use, and overwrite your private key.

**Prerequisites**
Verify that the user account you are logged in with has the necessary permissions to run SSH workflows.

**Procedure**

1. Click the **Workflows** view in the Orchestrator client.
2. In the workflows hierarchical list, expand **Library > SSH** and navigate to the Generate key pair workflow.
3. Right-click the Generate key pair workflow and select **Start workflow**.
4. Provide the required information.
   a. Select the key type.
   b. Select the key size.
   c. (Optional) Enter a passphrase.
      
      **Note**  You can change the passphrase later.
   d. (Optional) Enter a comment.
5. Click **Submit** to run the workflow.

If a key pair exists, the new key pair overwrites it.

Change the Key Pair Passphrase

You can run a workflow from the Orchestrator client to change the passphrase for the key pair that you generated most recently.

**Prerequisites**
Verify that the user account you are logged in with has the necessary permissions to run SSH workflows.

**Procedure**

1. Click the **Workflows** view in the Orchestrator client.
2 In the workflows hierarchical list, expand **Library > SSH** and navigate to the Change key pair passphrase workflow.

3 Right-click the Change key pair passphrase workflow and select **Start workflow**.

4 Reset the key pair passphrase.
   a Enter the current passphrase.
   b Enter the new passphrase.

5 Click **Submit** to run the workflow.

**Register an Orchestrator Public Key on an SSH Host**

You can use a public key instead of a password. To register an Orchestrator public key on an SSH host, you can run a workflow from the Orchestrator client.

**Prerequisites**

Verify that the user account you are logged in with has the necessary permissions to run SSH workflows.

**Procedure**

1 Click the **Workflows** view in the Orchestrator client.

2 In the workflows hierarchical list, expand **Library > SSH** and navigate to the Register vCO public key on host workflow.

3 Right-click the Register vCO public key on host workflow and select **Start workflow**.

4 Provide the name of the SSH host, and the user name and password to log in to this host.

   **Note** You must provide credentials that are registered on the SSH host.

5 Click **Submit** to run the workflow.

You can use public key authentication instead of password authentication when you connect to the SSH host as the registered user.

**Run an SSH Command**

You can run a workflow from the Orchestrator client to run SSH commands on a remote SSH server.

**Prerequisites**

Verify that the user account you are logged in with has the necessary permissions to run SSH workflows.

**Procedure**

1 Click the **Workflows** view in the Orchestrator client.

2 In the workflows hierarchical list, expand **Library > SSH** and navigate to the Run SSH command workflow.
3 Right-click the Run SSH command workflow and select **Start workflow**.

4 Enter an SSH host name or IP address, and click **Next**.

5 Enter an SSH command to run, and click **Next**.

   **Note** The default SSH command is **uptime**. It shows how long the server has been active and the user load for that period.

6 Select **Yes** to use password authentication, and click **Next**.

   **Note** The default option is to use key file authentication.

7 Enter a user name, and click **Next**.

8 Enter a password if the authentication method requires a password. Otherwise, enter the path to the private key and enter the passphrase for the private key.

9 Click **Submit** to run the workflow.

---

**Copy a File from an SSH Host**

You can run a workflow on the Orchestrator client to copy files from an SSH host to the Orchestrator server.

The SSH plug-in uses the Java JCraft library, which implements SFTP. The SCP get command workflow transfers files by using SFTP.

**Prerequisites**

Verify that the user account you are logged in with has the necessary permissions to run SSH workflows.

   **Note** Orchestrator must have explicit write permissions in order to write in folders.

**Procedure**

1 Click the **Workflows** view in the Orchestrator client.

2 In the workflows hierarchical list, expand **Library > SSH** and navigate to the SCP get command workflow.

3 Right-click the SCP get command workflow and select **Start workflow**.

4 Provide the required information, and click **Next**.
   a Type an SSH host name or IP address.
   b Type the SSH authentication information.

5 Type the file information.
   a Type the path to the directory on the Orchestrator server into which to copy the file.
   b Type the path to the file to get from the remote SSH host.

6 Click **Submit** to run the workflow.
Copy a File to an SSH Host

You can run a workflow from the Orchestrator client to copy files from the Orchestrator server to an SSH host.

The SSH plug-in uses the Java JCraft library, which implements SFTP. The SCP put command workflow transfers files by using SFTP.

Prerequisites

Verify that the user account you are logged in with has the necessary permissions to run SSH workflows.

Procedure

1. Click the Workflows view in the Orchestrator client.
2. In the workflows hierarchical list, expand Library > SSH and navigate to the SCP put command workflow.
3. Right-click the SCP put command workflow and select Start workflow.
4. Provide the required information, and click Next.
   a. Type an SSH host name or IP address.
   b. Type the SSH authentication information.
5. Enter the file information.
   a. Enter the path to the file that you want to copy from the local Orchestrator server to the remote SSH host.
   b. Enter the path to the directory on the remote SSH host into which to copy the file.
6. Click Submit to run the workflow.
Using the XML Plug-In

You can use the XML plug-in to run workflows that create and modify XML documents.

The XML plug-in adds an implementation of a Document Object Model (DOM) XML parser to the Orchestrator JavaScript API. The XML plug-in also provides some sample workflows to demonstrate how you can create and modify XML documents from workflows.

Alternatively, you can use the ECMAScript for XML (E4X) implementation in the Orchestrator JavaScript API to process XML documents directly in JavaScript. For an E4X scripting example, see Developing with VMware vRealize Orchestrator.

For information about E4X, go to the Web site of the organization that maintains the ECMA-357 standard.

Running the XML Plug-In Sample Workflows

You can run the XML plug-in sample workflows from the Orchestrator client to create and modify XML documents for testing purposes.

Because the workflows can create, read, or modify files, you must have sufficient access rights to the working directory.

Orchestrator has read, write, and execute rights to a folder named orchestrator, at the root of the server system. Although workflows have permission to read, write, and execute in this folder, you must create the folder on the server system. If you use the Orchestrator Appliance, the folder is named vco and is located at /var/run/vco.

You can allow access to other folders by changing the settings for server file system access from workflows and JavaScript. See Installing and Configuring VMware vRealize Orchestrator, Setting Server File System Access from Workflows and Actions.

- **Create a Simple XML Document**
  You can run a workflow from the Orchestrator client to create a simple XML document for testing purposes.

- **Find an Element in an XML Document**
  You can run a workflow from the Orchestrator client to find an element in the XML created by the Create a simple XML document workflow.
Modify an XML Document
You can run a workflow from the Orchestrator client to modify the XML that the Create a simple XML document workflow creates.

Create an Example Address Book from XML
You can run a workflow from the Orchestrator client to create an address book for testing purposes.

Create a Simple XML Document
You can run a workflow from the Orchestrator client to create a simple XML document for testing purposes.

Prerequisites
- Verify that the user account you are logged in with has the necessary permissions to run XML workflows.
- Verify that you created the `c:/orchestrator` folder at the root of the Orchestrator server system or set access rights to another folder.

Procedure
1. Click the Workflows view in the Orchestrator client.
2. In the workflows hierarchical list, open Library > XML > Samples XML (Simple) to navigate to the Create a simple XML document workflow.
3. Right-click the Create a simple XML document workflow and select Start workflow.
4. Type the filepath to the XML document to create.
   - For example, `c:/orchestrator/filename.xml`.
5. Click Submit to run the workflow.

The workflow creates an XML document that contains a list of users. The attributes for each entry are user ID and name.

Find an Element in an XML Document
You can run a workflow from the Orchestrator client to find an element in the XML created by the Create a simple XML document workflow.

Prerequisites
- Verify that the user account you are logged in with has the necessary permissions to run XML workflows.
- Verify that you created the `c:/orchestrator` folder at the root of the Orchestrator server system or set access rights to another folder.
Procedure

1. Click the Workflows view in the Orchestrator client.
2. In the workflows hierarchical list, open Library > XML > Samples XML (Simple) to navigate to the Find element in document workflow.
3. Right-click the Find element in document workflow and select Start workflow.
4. Type the filepath to the XML document.
   For example, c:/orchestrator/filename.xml.
5. Click Submit to run the workflow.
   The workflow searches for an element and displays the result in the system log.

What to do next
To view the result, select the completed workflow run in the Orchestrator client and click Logs on the Schema tab.

Modify an XML Document
You can run a workflow from the Orchestrator client to modify the XML that the Create a simple XML document workflow creates.

Prerequisites
- Verify that the user account you are logged in with has the necessary permissions to run XML workflows.
- Verify that you created the c:/orchestrator folder at the root of the Orchestrator server system or set access rights to another folder.

Procedure

1. Click the Workflows view in the Orchestrator client.
2. In the workflows hierarchical list, open Library > XML > Samples XML (Simple) to navigate to the Modify XML document workflow.
3. Right-click the Modify XML document workflow and select Start workflow.
4 Provide the input and output filepaths.
   a Type the filepath to the XML document to modify.
      For example, c:/orchestrator/filename.xml.
   b Type the filepath to the modified XML document.
      For example, c:/orchestrator/filename.xml.

   **Note** If you type the same filepath in both fields, the workflow overwrites the original file with the modified file. If you type an output filepath to a file that does not exist, the workflow creates a modified file.

5 Click **Submit** to run the workflow.

The workflow searches for an element and modifies the entry where the element is found.

**Create an Example Address Book from XML**

You can run a workflow from the Orchestrator client to create an address book for testing purposes.

**Prerequisites**

- Verify that the user account you are logged in with has the necessary permissions to run XML workflows.
- Verify that you created the c:/orchestrator folder at the root of the Orchestrator server system or set access rights to another folder.

**Procedure**

1 Click the **Workflows** view in the Orchestrator client.

2 In the workflows hierarchical list, open **Library > XML > Samples XML (Address Book)** to navigate to the Full address book test workflow.

3 Right-click the Full address book test workflow and select **Start workflow**.

4 Type the path to the address book folder.
   For example, c:/orchestrator/foldername.
   The workflow automatically creates the folder if it does not exist.

5 Click **Submit** to run the workflow.

The workflow creates a DTD, an XML, and a CSS file, appends the stylesheet, and stores the files in the specified folder.
Using the Mail Plug-In

You can send email messages from workflows by using the Mail plug-in, which uses the Simple Mail Transfer Protocol (SMTP). For example, you can create a workflow to send an email to a given address if the workflow requires user interaction or when it completes its run.

This chapter includes the following topics:

- Define the Default SMTP Connection
- Using the Mail Plug-In Sample Workflows

Define the Default SMTP Connection

The Mail plug-in is installed together with the Orchestrator server and is used for sending and receiving email notifications. You can set the default email account that can authenticate against an SMTP server to send and receive email notifications.

**Note** Avoid load balancers when configuring mail in Orchestrator. You might receive SMTP_HOST_UNREACHABLE error.

**Procedure**

1. Log in to the Orchestrator client as an administrator.
2. Click the **Workflows** view in the Orchestrator client.
3. In the workflows hierarchical list, expand **Library > Mail** and navigate to the Configure mail workflow.
4. Right-click the Configure mail workflow and select **Start workflow**.
5. Enter the required information.

<table>
<thead>
<tr>
<th>Text Box</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>SMTP host</td>
<td>Enter the IP address or domain name of your SMTP server.</td>
</tr>
<tr>
<td>SMTP port</td>
<td>Enter a port number to match your SMTP configuration. The default SMTP port is 25.</td>
</tr>
<tr>
<td>User name</td>
<td>Enter a valid email account. This is the email account that Orchestrator uses to send emails.</td>
</tr>
<tr>
<td>Password</td>
<td>Enter the password associated with the user name.</td>
</tr>
<tr>
<td>From name and address</td>
<td>Enter the sender information to appear in all emails sent by Orchestrator.</td>
</tr>
</tbody>
</table>
Using the Mail Plug-In Sample Workflows

You can call the sample workflows of the Mail plug-in from custom workflows to implement the email functionality to the custom workflows. You can run an example workflow to test the interaction between Orchestrator and your SMTP server.

- **Access the Mail Plug-In Sample Workflows**
  You can access the Mail plug-in sample workflows through the Orchestrator client.

- **Mail Plug-In Sample Workflows**
  You can enhance your custom workflows by integrating the sample Mail plug-in workflows.

**Access the Mail Plug-In Sample Workflows**

You can access the Mail plug-in sample workflows through the Orchestrator client.

**Prerequisites**

Verify that the user account you are logged in with has the necessary permissions to run Mail workflows.

**Procedure**

1. Click the **Workflows** view in the Orchestrator client.
2. Expand the hierarchical list to **Library > Mail**.

**What to do next**

Review and run the sample workflows.

**Mail Plug-In Sample Workflows**

You can enhance your custom workflows by integrating the sample Mail plug-in workflows.

You can access the Mail workflows from **Library > Mail** in the **Workflows** view of the Orchestrator client.

<table>
<thead>
<tr>
<th>Workflow Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Configure mail</td>
<td>Defines the connection to the SMTP server, the SMTP authentication account, and the address and display name of the sender.</td>
</tr>
<tr>
<td>Retrieve messages</td>
<td>Retrieves the messages of a given email account by using the POP3 protocol.</td>
</tr>
<tr>
<td>Retrieve messages (via MailClient)</td>
<td>Retrieves the messages of a certain email account, without deleting them, by using the new scripting API provided by the MailClient class.</td>
</tr>
<tr>
<td>Send notification</td>
<td>Sends an email with specified content to a given email address. If optional parameters are not specified, the workflow uses the default values set through the Configure mail workflow.</td>
</tr>
<tr>
<td>Send notification to mailing list</td>
<td>Sends an email with specified content to a given email address list, CC list, and BCC list. If optional parameters are not specified, the workflow uses the default values set through the Configure mail workflow.</td>
</tr>
</tbody>
</table>
Using the Net Plug-In

You can use the Net plug-in to implement the Telnet, FTP, POP3, and IMAP protocols in workflows. The POP3 and IMAP implementations allow downloading and reading email. In combination with the Mail plug-in, the Net plug-in provides full email sending and receiving capabilities in workflows.
Using the Enumeration Plug-In

You can use the Enumeration plug-in to implement common enumerated types in workflows.

**Time Zone Codes**

You can use the time zone codes as possible values for the Enums:MSTimeZone enumeration.

<table>
<thead>
<tr>
<th>Time Zone Code</th>
<th>Time Zone Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>000</td>
<td>Dateline Standard Time</td>
<td>(GMT-12:00) International Date Line West</td>
</tr>
<tr>
<td>001</td>
<td>Samoa Standard Time</td>
<td>(GMT-11:00) Midway Island, Samoa</td>
</tr>
<tr>
<td>002</td>
<td>Hawaiian Standard Time</td>
<td>(GMT-10:00) Hawaii</td>
</tr>
<tr>
<td>003</td>
<td>Alaskan Standard Time</td>
<td>(GMT-09:00) Alaska</td>
</tr>
<tr>
<td>004</td>
<td>Pacific Standard Time</td>
<td>(GMT-08:00) Pacific Time (US and Canada); Tijuana</td>
</tr>
<tr>
<td>010</td>
<td>Mountain Standard Time</td>
<td>(GMT-07:00) Mountain Time (US and Canada)</td>
</tr>
<tr>
<td>013</td>
<td>Mexico Standard Time 2</td>
<td>(GMT-07:00) Chihuahua, La Paz, Mazatlan</td>
</tr>
<tr>
<td>015</td>
<td>U.S. Mountain Standard Time</td>
<td>(GMT-07:00) Arizona</td>
</tr>
<tr>
<td>020</td>
<td>Central Standard Time</td>
<td>(GMT-06:00) Central Time (US and Canada)</td>
</tr>
<tr>
<td>025</td>
<td>Canada Central Standard Time</td>
<td>(GMT-06:00) Saskatchewan</td>
</tr>
<tr>
<td>030</td>
<td>Mexico Standard Time</td>
<td>(GMT-06:00) Guadalajara, Mexico City, Monterrey</td>
</tr>
<tr>
<td>033</td>
<td>Central America Standard Time</td>
<td>(GMT-06:00) Central America</td>
</tr>
<tr>
<td>035</td>
<td>Eastern Standard Time</td>
<td>(GMT-05:00) Eastern Time (US and Canada)</td>
</tr>
<tr>
<td>040</td>
<td>U.S. Eastern Standard Time</td>
<td>(GMT-05:00) Indiana (East)</td>
</tr>
<tr>
<td>045</td>
<td>S.A. Pacific Standard Time</td>
<td>(GMT-05:00) Bogota, Lima, Quito</td>
</tr>
<tr>
<td>050</td>
<td>Atlantic Standard Time</td>
<td>(GMT-04:00) Atlantic Time (Canada)</td>
</tr>
<tr>
<td>055</td>
<td>S.A. Western Standard Time</td>
<td>(GMT-04:00) Caracas, La Paz</td>
</tr>
<tr>
<td>056</td>
<td>Pacific S.A. Standard Time</td>
<td>(GMT-04:00) Santiago</td>
</tr>
<tr>
<td>Time Zone Code</td>
<td>Time Zone Name</td>
<td>Description</td>
</tr>
<tr>
<td>---------------</td>
<td>---------------------------------------------------------</td>
<td>-----------------------------------------------</td>
</tr>
<tr>
<td>060</td>
<td>Newfoundland and Labrador Standard Time</td>
<td>(GMT-03:30) Newfoundland and Labrador Standard Time</td>
</tr>
<tr>
<td>065</td>
<td>E. South America Standard Time</td>
<td>(GMT-03:00) Brasilia</td>
</tr>
<tr>
<td>070</td>
<td>S.A. Eastern Standard Time</td>
<td>(GMT-03:00) Buenos Aires, Georgetown</td>
</tr>
<tr>
<td>073</td>
<td>Greenland Standard Time</td>
<td>(GMT-03:00) Greenland</td>
</tr>
<tr>
<td>075</td>
<td>Mid-Atlantic Standard Time</td>
<td>(GMT-02:00) Mid-Atlantic</td>
</tr>
<tr>
<td>080</td>
<td>Azores Standard Time</td>
<td>(GMT-01:00) Azores</td>
</tr>
<tr>
<td>083</td>
<td>Cape Verde Standard Time</td>
<td>(GMT-01:00) Cape Verde Islands</td>
</tr>
<tr>
<td>090</td>
<td>Greenwich Standard Time</td>
<td>(GMT) Casablanca, Monrovia</td>
</tr>
<tr>
<td>095</td>
<td>Central Europe Standard Time</td>
<td>(GMT+01:00) Belgrade, Bratislava, Budapest, Ljubljana, Prague</td>
</tr>
<tr>
<td>100</td>
<td>Central European Standard Time</td>
<td>(GMT+01:00) Sarajevo, Skopje, Warsaw, Zagreb</td>
</tr>
<tr>
<td>105</td>
<td>Romance Standard Time</td>
<td>(GMT+01:00) Brussels, Copenhagen, Madrid, Paris</td>
</tr>
<tr>
<td>110</td>
<td>W. Europe Standard Time</td>
<td>(GMT+01:00) Amsterdam, Berlin, Bern, Rome, Stockholm, Vienna</td>
</tr>
<tr>
<td>113</td>
<td>W. Central Africa Standard Time</td>
<td>(GMT+01:00) West Central Africa</td>
</tr>
<tr>
<td>115</td>
<td>E. Europe Standard Time</td>
<td>(GMT+02:00) Bucharest</td>
</tr>
<tr>
<td>120</td>
<td>Egypt Standard Time</td>
<td>(GMT+02:00) Cairo</td>
</tr>
<tr>
<td>125</td>
<td>FLE Standard Time</td>
<td>(GMT+02:00) Helsinki, Kyiv, Riga, Sofia, Tallinn, Vilnius</td>
</tr>
<tr>
<td>130</td>
<td>GTB Standard Time</td>
<td>(GMT+02:00) Athens, Istanbul, Minsk</td>
</tr>
<tr>
<td>135</td>
<td>Israel Standard Time</td>
<td>(GMT+02:00) Jerusalem</td>
</tr>
<tr>
<td>140</td>
<td>South Africa Standard Time</td>
<td>(GMT+02:00) Harare, Pretoria</td>
</tr>
<tr>
<td>145</td>
<td>Russian Standard Time</td>
<td>(GMT+03:00) Moscow, St. Petersburg, Volgograd</td>
</tr>
<tr>
<td>150</td>
<td>Arab Standard Time</td>
<td>(GMT+03:00) Kuwait, Riyadh</td>
</tr>
<tr>
<td>155</td>
<td>E. Africa Standard Time</td>
<td>(GMT+03:00) Nairobi</td>
</tr>
<tr>
<td>158</td>
<td>Arabic Standard Time</td>
<td>(GMT+03:00) Baghdad</td>
</tr>
<tr>
<td>160</td>
<td>Iran Standard Time</td>
<td>(GMT+03:30) Tehran</td>
</tr>
<tr>
<td>165</td>
<td>Arabian Standard Time</td>
<td>(GMT+04:00) Abu Dhabi, Muscat</td>
</tr>
<tr>
<td>170</td>
<td>Caucasus Standard Time</td>
<td>(GMT+04:00) Baku, Tbilisi, Yerevan</td>
</tr>
<tr>
<td>175</td>
<td>Transitional Islamic State of Afghanistan Standard Time</td>
<td>(GMT+04:30) Kabul</td>
</tr>
<tr>
<td>180</td>
<td>Ekaterinburg Standard Time</td>
<td>(GMT+05:00) Ekaterinburg</td>
</tr>
<tr>
<td>Time Zone Code</td>
<td>Time Zone Name</td>
<td>Description</td>
</tr>
<tr>
<td>---------------</td>
<td>--------------------------------------</td>
<td>--------------------------------------------------------------</td>
</tr>
<tr>
<td>185</td>
<td>West Asia Standard Time</td>
<td>(GMT+05:00) Islamabad, Karachi, Tashkent</td>
</tr>
<tr>
<td>190</td>
<td>India Standard Time</td>
<td>(GMT+05:30) Chennai, Kolkata, Mumbai, New Delhi</td>
</tr>
<tr>
<td>193</td>
<td>Nepal Standard Time</td>
<td>(GMT+05:45) Kathmandu</td>
</tr>
<tr>
<td>195</td>
<td>Central Asia Standard Time</td>
<td>(GMT+06:00) Astana, Dhaka</td>
</tr>
<tr>
<td>200</td>
<td>Sri Lanka Standard Time</td>
<td>(GMT+06:00) Sri Jayawardenepura</td>
</tr>
<tr>
<td>201</td>
<td>N. Central Asia Standard Time</td>
<td>(GMT+06:00) Almaty, Novosibirsk</td>
</tr>
<tr>
<td>203</td>
<td>Myanmar Standard Time</td>
<td>(GMT+06:30) Yangon (Rangoon)</td>
</tr>
<tr>
<td>205</td>
<td>S.E. Asia Standard Time</td>
<td>(GMT+07:00) Bangkok, Hanoi, Jakarta</td>
</tr>
<tr>
<td>207</td>
<td>North Asia Standard Time</td>
<td>(GMT+07:00) Krasnoyarsk</td>
</tr>
<tr>
<td>210</td>
<td>China Standard Time</td>
<td>(GMT+08:00) Beijing, Chongqing, Hong Kong SAR, Urumqi</td>
</tr>
<tr>
<td>215</td>
<td>Singapore Standard Time</td>
<td>(GMT+08:00) Kuala Lumpur, Singapore</td>
</tr>
<tr>
<td>220</td>
<td>Taipei Standard Time</td>
<td>(GMT+08:00) Taipei</td>
</tr>
<tr>
<td>225</td>
<td>W. Australia Standard Time</td>
<td>(GMT+08:00) Perth</td>
</tr>
<tr>
<td>227</td>
<td>North Asia East Standard Time</td>
<td>(GMT+08:00) Irkutsk, Ulaan Bataar</td>
</tr>
<tr>
<td>230</td>
<td>Korea Standard Time</td>
<td>(GMT+09:00) Seoul</td>
</tr>
<tr>
<td>235</td>
<td>Tokyo Standard Time</td>
<td>(GMT+09:00) Osaka, Sapporo, Tokyo</td>
</tr>
<tr>
<td>240</td>
<td>Yakutsk Standard Time</td>
<td>(GMT+09:00) Yakutsk</td>
</tr>
<tr>
<td>245</td>
<td>A.U.S. Central Standard Time</td>
<td>(GMT+09:30) Darwin</td>
</tr>
<tr>
<td>250</td>
<td>Cen. Australia Standard Time</td>
<td>(GMT+09:30) Adelaide</td>
</tr>
<tr>
<td>255</td>
<td>A.U.S. Eastern Standard Time</td>
<td>(GMT+10:00) Canberra, Melbourne, Sydney</td>
</tr>
<tr>
<td>260</td>
<td>E. Australia Standard Time</td>
<td>(GMT+10:00) Brisbane</td>
</tr>
<tr>
<td>265</td>
<td>Tasmania Standard Time</td>
<td>(GMT+10:00) Hobart</td>
</tr>
<tr>
<td>270</td>
<td>Vladivostok Standard Time</td>
<td>(GMT+10:00) Vladivostok</td>
</tr>
<tr>
<td>275</td>
<td>West Pacific Standard Time</td>
<td>(GMT+10:00) Guam, Port Moresby</td>
</tr>
<tr>
<td>280</td>
<td>Central Pacific Standard Time</td>
<td>(GMT+11:00) Magadan, Solomon Islands, New Caledonia</td>
</tr>
<tr>
<td>285</td>
<td>Fiji Islands Standard Time</td>
<td>(GMT+12:00) Fiji Islands, Kamchatka, Marshall Islands</td>
</tr>
<tr>
<td>290</td>
<td>New Zealand Standard Time</td>
<td>(GMT+12:00) Auckland, Wellington</td>
</tr>
<tr>
<td>300</td>
<td>Tonga Standard Time</td>
<td>(GMT+13:00) Nuku'alofa</td>
</tr>
</tbody>
</table>
Using the Workflow Documentation Plug-In

You can use the Workflow Documentation plug-in to generate PDF documentation about a specific workflow or workflow category.

This chapter includes the following topics:

- Workflow Library for the Workflow Documentation Plug-In
- Generate Workflow Documentation

Workflow Library for the Workflow Documentation Plug-In

With the Workflow Documentation plug-in workflows, you can generate PDF documentation about specific workflows or workflow categories.

You can access these workflows from Library > Workflow documentation in the Workflows view of the Orchestrator client.

<table>
<thead>
<tr>
<th>Workflow Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Get documentation for workflow</td>
<td>Generates information about a workflow that you select.</td>
</tr>
<tr>
<td>Get documentation for workflow category</td>
<td>Generates information about a workflow category that you select.</td>
</tr>
</tbody>
</table>

Generate Workflow Documentation

You can export documentation in PDF format about a workflow or a workflow folder that you select at any time.

The exported document contains detailed information about the selected workflow or the workflows in the folder. The information about each workflow includes name, version history of the workflow, attributes, parameter presentation, workflow schema, and workflow actions. In addition, the documentation also provides the source code for the used actions.

Procedure

1. From the drop-down menu in the Orchestrator client, select Run or Design.
2. Click the Workflows view.
3 Navigate to the workflow or workflow folder for which you want to generate documentation and right-click it.

4 Select **Generate documentation**.

5 Browse to locate the folder in which to save the PDF file, provide a file name, and click **Save**.

The PDF file containing the information about the selected workflow, or the workflows in the folder, is saved on your system.
Using the HTTP-REST Plug-In

The HTTP-REST plug-in allows you to manage REST Web services by providing interaction between vRealize Orchestrator and REST hosts. You can define REST services and their operations as inventory objects by running configuration workflows, and perform REST operations on the defined objects.

The plug-in contains a set of standard workflows related to managing REST hosts and invoking REST operations. You can also generate custom workflows to automate tasks in a REST environment.

This chapter includes the following topics:

- Configuring the HTTP-REST Plug-In
- Generate a New Workflow from a REST Operation
- Invoking a REST Operation

Configuring the HTTP-REST Plug-In

You must use the Orchestrator client to configure the HTTP-REST plug-in.

Configuration Workflows

The Configuration workflow category contains workflows that allow you to manage REST hosts.

You can access these workflows from Library > HTTP-REST > Configuration on the Workflows view in the Orchestrator client.

<table>
<thead>
<tr>
<th>Workflow Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Add a REST host</td>
<td>Adds a REST host to the plug-in's inventory.</td>
</tr>
<tr>
<td>Add a REST operation</td>
<td>Adds an operation to a REST host.</td>
</tr>
<tr>
<td>Add schema to a REST host</td>
<td>Adds an XSD schema to a REST host.</td>
</tr>
<tr>
<td>Clone a REST host</td>
<td>Creates a clone of a REST host.</td>
</tr>
<tr>
<td>Clone a REST operation</td>
<td>Creates a clone of a REST operation.</td>
</tr>
<tr>
<td>Reload plug-in configuration</td>
<td>Refreshes the list of REST hosts in the plug-in’s inventory.</td>
</tr>
<tr>
<td>Remove a REST host</td>
<td>Removes a REST host from the plug-in’s inventory.</td>
</tr>
<tr>
<td>Remove a REST operation</td>
<td>Removes an operation from a REST host.</td>
</tr>
<tr>
<td>Remove schemas form a REST host</td>
<td>Removes all associated XSD schemas from a REST host.</td>
</tr>
<tr>
<td>Workflow Name</td>
<td>Description</td>
</tr>
<tr>
<td>-------------------------------</td>
<td>-------------------------------------------------------</td>
</tr>
<tr>
<td>Update a REST host</td>
<td>Updates a REST host in the plug-in’s inventory.</td>
</tr>
<tr>
<td>Update a REST operation</td>
<td>Updates an operation on a REST host.</td>
</tr>
</tbody>
</table>

**Configure Kerberos Authentication**

You can use Kerberos authentication when you add a host.

The `krb5.conf` file contains the following information:

- Kerberos configuration information
- Locations of Key Distribution Centers (KDC) and administration servers for the Kerberos realms of interest
- Default values for the current realm and for Kerberos applications
- Mappings of host names onto Kerberos realms

**Procedure**

- Create a `krb5.conf` file and save it to the following location.

<table>
<thead>
<tr>
<th>Operating System</th>
<th>Path</th>
</tr>
</thead>
<tbody>
<tr>
<td>Windows</td>
<td>C:\Program Files\Common Files\VMware\VMware vCenter Server - Java</td>
</tr>
<tr>
<td></td>
<td>Components\lib\security\</td>
</tr>
<tr>
<td>Linux</td>
<td>/usr/java/jre-vmware/lib/security/</td>
</tr>
</tbody>
</table>

A `krb5.conf` file has the following structure:

```plaintext
[libdefaults]
default_realm = YOURDOMAIN.COM
udp_preference_limit = 1
[realms]
YOURDOMAIN.COM = {
kdc = kdc.yourdomain.com
default_domain = yourdomain.com
}
[domain_realm]
.yourdomain.com=YOURDOMAIN.COM
yourdomain.com=YOURDOMAIN.COM
```

**Note** The Kerberos authentication requires a Fully Qualified Domain Name (FQDN) host address.

**Important** When you add or modify the `krb5.conf` file, you must restart the Orchestrator server service.

**Add a REST Host**

You can run a workflow to add a REST host and configure the host connection parameters.
Procedure

1. Log in to the Orchestrator client as an administrator.
2. Click the **Workflows** view in the Orchestrator client.
3. In the workflows hierarchical list, expand **Library > HTTP-REST > Configuration** and navigate to the Add a REST host workflow.
4. Right-click the Add a REST host workflow and select **Start workflow**.
5. In the **Name** text box, enter the name of the host.
6. In the **URL** text box, enter the address of the host.
   
   **Note**  The Kerberos authentication requires a Fully Qualified Domain Name (FQDN) host address.
7. In the **Connection timeout** text box, enter the number of seconds before a connection times out.
8. In the **Operation timeout** text box, enter the number of seconds before an operation times out.
9. Select **Yes** to accept the REST host certificate.
   
   The certificate is added to the Orchestrator server trust store.
10. Select the authentication type.

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>None</td>
<td>No authentication is required.</td>
</tr>
<tr>
<td>OAuth 1.0</td>
<td>Provide the required authentication parameters.</td>
</tr>
<tr>
<td>OAuth 2.0</td>
<td>Provide the authentication token.</td>
</tr>
<tr>
<td>Basic</td>
<td>Provides basic access authentication. Select the session mode.</td>
</tr>
<tr>
<td>Digest</td>
<td>Provides digest access authentication that uses encryption. Select the session mode.</td>
</tr>
</tbody>
</table>

- If you select **Shared Session**, provide credentials for the shared session.
- If you select **Per User Session**, the Orchestrator client retrieves credentials from the user who is logged in.
<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
</table>
| NTLM   | Provides NT LAN Manager (NTLM) access authentication within the Window Security Support Provider (SSPI) framework. Select the session mode.  
- If you select Shared Session, provide credentials for the shared session.  
- If you select Per User Session, the Orchestrator client retrieves credentials from the user who is logged in.  
Provide the NTLM settings. |
| Kerberos | Provides Kerberos access authentication.  
Select the session mode.  
- If you select Shared Session, provide credentials for the shared session.  
- If you select Per User Session, the Orchestrator client retrieves credentials from the user who is logged in. |

11 To use a proxy, enter the address and the port of the proxy server.  
   a  (Optional) Select the proxy authentication type.  
<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>None</td>
<td>No authentication is required.</td>
</tr>
</tbody>
</table>
| Basic  | Provides basic access authentication.  
Select the session mode.  
- If you select Shared Session, provide credentials for the shared session.  
- If you select Per User Session, the Orchestrator client retrieves credentials from the user who is logged in. |

12 Select whether you want the target hostname to match the name stored in the server certificate.  
13 (Optional) Select a keystore entry to use to authenticate against the server. The keystore entry must be of the PrivateKeyEntry type.  
14 Click Submit to run the workflow.  

After the workflow runs successfully, the REST host appears in the Inventory view.  

What to do next  
You can add operations and XSD schema to the REST host, and run workflows from the Inventory view.  

**Add a REST Operation**  
You can run a workflow to add an operation to a REST host from the plug-in's inventory.  

**Prerequisites**  
- Verify that you are logged in to the Orchestrator client as an administrator.  
- Verify that you have a connection to a REST host from the Inventory view.  

**Procedure**  
1 Click the Workflows view in the Orchestrator client.
2. In the workflows hierarchical list, select **Library > HTTP-REST > Configuration** and navigate to the Add a REST operation workflow.

3. Right-click the Add a REST operation workflow and select **Start workflow**.

4. Select the host to which you want to add the operation.

5. In the **Name** text box, type the name of the operation.

6. In the **Template URL** text box, type only the operation part of the URL. You can include placeholders for parameters that are provided when you run the operation. The following is an example URL syntax. 

   `/customer/{id}/orders?date={date}`

7. Select the HTTP method that the operation uses. If you select **POST** or **PUT**, you can provide content type for the method.

8. Click **Submit** to run the workflow.

**What to do next**

You can run workflows on the operation from the **Inventory** view.

### Add a Schema to a REST Host

You can run a workflow to add an XSD schema to a REST host from the plug-in's inventory. The XSD schema describes the XML documents that are used as input and output content from Web services. By associating such a schema with a host, you can specify the XML element that is required as an input when you are generating a workflow from a REST operation.

**Prerequisites**

- Verify that you are logged in to the Orchestrator client as an administrator.
- Verify that you have a connection to a REST host from the **Inventory** view.

**Procedure**

1. Click the **Workflows** view in the Orchestrator client.

2. In the workflows hierarchical list, select **Library > HTTP-REST > Configuration** to navigate to the Add a schema to a REST host workflow.

3. Right-click the Add a schema to a REST host workflow and select **Start workflow**.

4. Select the host to which you want to add the XSD schema.

5. Select whether to load the schema from URL.

<table>
<thead>
<tr>
<th>Option</th>
<th>Action</th>
</tr>
</thead>
<tbody>
<tr>
<td>Yes</td>
<td>Type the URL of the schema.</td>
</tr>
<tr>
<td>No</td>
<td>Provide the schema content.</td>
</tr>
</tbody>
</table>
Click Submit to run the workflow.

Generate a New Workflow from a REST Operation

You can create a custom workflow from a REST operation.

You can integrate custom-generated workflows into high-level workflows. For more information about workflow development, see the vRealize Orchestrator Developer's Guide.

Prerequisites

- Verify that you are logged in to the Orchestrator client as an administrator.
- Verify that you have a connection to a REST host from the Inventory view.

Procedure

1. Click the Workflows view in the Orchestrator client.
2. In the workflows hierarchical list, expand Library > HTTP-REST and navigate to the Generate a new workflow from a REST operation workflow.
3. Right-click the Generate a new workflow from a REST operation workflow and select Start workflow.
4. Select the REST operation from the list of available operations.
   - If the operation takes input and XSD schemas are added to its host, you can specify the request input type.
5. In the Name text box, type the name of the workflow to generate.
6. Select the workflow folder in which to generate the new workflow.
   - You can select any existing folder from the workflow library.
7. Click Submit to run the workflow.

Invoking a REST Operation

To make REST requests, you can either invoke a configured REST operation or invoke a REST operation dynamically by using a configured REST operation as a template and replacing any of the parameters at runtime.

There are several ways to invoke a REST operation.

- Configure REST hosts and associate REST operations with them by running the Add a REST Host and Add a REST Operation workflows. The registered REST hosts and REST operations are persistent and can be found in the Inventory and Resources views.
- Invoke a REST operation without previously configuring REST hosts and adding REST operations by running the Invoke a dynamic REST operation workflow from Library > HTTP-REST Samples. With this workflow, you can provide REST host base URL and operation parameters. The data is not persistent and is not available in the Inventory and Resources views.
Configure REST hosts, associate REST operations with them, and use the configured REST hosts and REST operations as templates for further use, by running the **Invoke a REST host with dynamic params** and **Invoke a REST operation with dynamic params** workflows from Library > HTTP-REST Samples. You can replace some of the parameters of already configured REST hosts and REST operations when you run the workflows. The original REST hosts and REST operations are not affected.

**Invoke a REST Operation**

Call a REST operation directly

**Prerequisites**

- Verify that you are logged in to the Orchestrator client as an administrator.
- Verify that you have a connection to a REST host from the Inventory view.

**Procedure**

1. Click the Workflows view in the Orchestrator client.
2. In the workflows hierarchical list, expand Library > HTTP-REST and navigate to the **Invoke a REST operation** workflow.
3. Right-click the **Invoke a REST operation** workflow and select **Start workflow**.
4. Select the REST operation from the list of available operations.
5. Provide the input parameters and content that the operation requires.
6. Click **Submit** to run the workflow.
The SOAP plug-in allows you to manage SOAP Web services by providing interaction between vRealize Orchestrator and SOAP hosts. You can define SOAP services as inventory objects by running configuration workflows, and perform SOAP operations on the defined objects.

The plug-in contains a set of standard workflows related to managing SOAP hosts and invoking SOAP operations. You can also generate custom workflows to automate tasks in a SOAP environment.

This chapter includes the following topics:

- Configuring the SOAP Plug-In
- Generate a New Workflow from a SOAP Operation
- Invoke a SOAP Operation

Configuring the SOAP Plug-In

You must use the Orchestrator client to configure the SOAP plug-in.

Configuration Workflows

The Configuration workflow category contains workflows that allow you to manage SOAP hosts.

You can access these workflows from Library > SOAP > Configuration on the Workflows view in the Orchestrator client.

<table>
<thead>
<tr>
<th>Workflow Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Add a SOAP host</td>
<td>Adds a SOAP host to the plug-in inventory.</td>
</tr>
<tr>
<td>Reload plug-in configuration</td>
<td>Refreshes the list of SOAP hosts in the plug-in inventory.</td>
</tr>
<tr>
<td>Remove a SOAP host</td>
<td>Removes a SOAP host from the plug-in inventory.</td>
</tr>
<tr>
<td></td>
<td>Caution When you remove a host from the inventory, all workflows generated from it stop working.</td>
</tr>
<tr>
<td>Update a SOAP host</td>
<td>Updates a SOAP host in the plug-in inventory.</td>
</tr>
<tr>
<td>Update a SOAP host with an endpoint URL</td>
<td>Updates a SOAP host with a preferred endpoint address. The new endpoint address is used for sending and receiving SOAP messages, instead of the endpoint address defined within the WSDL.</td>
</tr>
</tbody>
</table>
Add a SOAP Host

You can run a workflow to add a SOAP host and configure the host connection parameters.

Procedure

1. Log in to the Orchestrator client as an administrator.
2. Click the Workflows view in the Orchestrator client.
3. In the workflows hierarchical list, expand Library > SOAP > Configuration and navigate to the Add a SOAP host workflow.
4. Right-click the Add a SOAP host workflow and select Start workflow.
5. In the Name text box, enter the name of the host.
6. Select whether to provide the WSDL content as text.

<table>
<thead>
<tr>
<th>Option</th>
<th>Action</th>
</tr>
</thead>
<tbody>
<tr>
<td>Yes</td>
<td>Copy the text in the WSDL content text box.</td>
</tr>
<tr>
<td>No</td>
<td>Enter the correct path in the WSDL URI text box.</td>
</tr>
</tbody>
</table>

7. In the Connection timeout text box, enter the number of seconds, within which Orchestrator must connect to the SOAP host, otherwise the connection times out.

8. In the Request timeout text box, specify the number of seconds, within which a SOAP request must succeed, before it times out.

9. Select whether to use a proxy.

<table>
<thead>
<tr>
<th>Option</th>
<th>Action</th>
</tr>
</thead>
<tbody>
<tr>
<td>Yes</td>
<td>Provide the proxy address and proxy port.</td>
</tr>
<tr>
<td>No</td>
<td>Continue to the next step.</td>
</tr>
</tbody>
</table>

10. Select the authentication type.

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>None</td>
<td>No authentication is required.</td>
</tr>
<tr>
<td>Basic</td>
<td>Provides basic access authentication. Select the session mode.</td>
</tr>
<tr>
<td></td>
<td>- If you select Shared Session, provide credentials for the shared session.</td>
</tr>
<tr>
<td></td>
<td>- If you select Per User Session, the Orchestrator client retrieves credentials from the user who is logged in.</td>
</tr>
<tr>
<td>Digest</td>
<td>Provides digest access authentication that uses encryption. Select the session mode.</td>
</tr>
<tr>
<td></td>
<td>- If you select Shared Session, provide credentials for the shared session.</td>
</tr>
<tr>
<td></td>
<td>- If you select Per User Session, the Orchestrator client retrieves credentials from the user who is logged in.</td>
</tr>
</tbody>
</table>
### Option | Description
--- | ---
**NTLM** | Provides NT LAN Manager (NTLM) access authentication within the Window Security Support Provider (SSPI) framework.

Select the session mode.
- If you select **Shared Session**, provide credentials for the shared session.
- If you select **Per User Session**, the Orchestrator client retrieves credentials from the user who is logged in.

Provide the NTLM settings.

**Kerberos** | Provides Kerberos access authentication.

Select the session mode.
- If you select **Shared Session**, provide credentials for the shared session.
- If you select **Per User Session**, the Orchestrator client retrieves credentials from the user who is logged in.

---

11. Click **Submit** to run the workflow.

After the workflow runs successfully, the SOAP host appears in the **Inventory** view.

**What to do next**

You can explore the SOAP host objects and run workflows on them from the **Inventory** view.

### Configure Kerberos Authentication

You can use Kerberos authentication when you add a host.

The `krb5.conf` file contains the following information:

- Kerberos configuration information
- Locations of Key Distribution Centers (KDC) and administration servers for the Kerberos realms of interest
- Default values for the current realm and for Kerberos applications
- Mappings of host names onto Kerberos realms

**Procedure**

- Create a `krb5.conf` file and save it to the following location.

<table>
<thead>
<tr>
<th>Operating System</th>
<th>Path</th>
</tr>
</thead>
<tbody>
<tr>
<td>Windows</td>
<td><code>C:\Program Files\Common Files\VMware\VMware vCenter Server - Java Components\lib\security\</code></td>
</tr>
<tr>
<td>Linux</td>
<td><code>/usr/java/jre-vmware/lib/security/</code></td>
</tr>
</tbody>
</table>

A `krb5.conf` file has the following structure:

```ini
[libdefaults]
default_realm = YOURDOMAIN.COM
udp_preference_limit = 1
[realms]
YOURDOMAIN.COM = {
```
kdc = kdc.yourdomain.com
default_domain = yourdomain.com
}
[domain_realm]
.yourdomain.com=YOURDOMAIN.COM
yourdomain.com=YOURDOMAIN.COM

---

**Note** The Kerberos authentication requires a Fully Qualified Domain Name (FQDN) host address.

**Important** When you add or modify the krb5.conf file, you must restart the Orchestrator server service.

---

## Generate a New Workflow from a SOAP Operation

You can create a custom workflow from a SOAP operation.

You can integrate custom-generated workflows into high-level workflows. For more information about workflow development, see the *vRealize Orchestrator Developer's Guide*.

### Prerequisites

- Verify that you are logged in to the Orchestrator client as an administrator.
- Verify that you have a connection to a SOAP host from the Inventory view.

### Procedure

1. Click the **Workflows** view in the Orchestrator client.
2. In the workflows hierarchical list, expand **Library > SOAP** to navigate to the Generate a new workflow from a SOAP operation workflow.
3. Right-click the Generate a new workflow from a SOAP operation workflow and select **Start workflow**.
4. Select the SOAP operation from the list of available operations.
5. In the **Name** text box, type the name of the workflow to generate.
6. Select the workflow folder in which to generate the new workflow.
   - You can select any existing folder from the workflow library.
7. Click **Submit** to run the workflow.

### What to do next

You can test the generated workflow.

---

## Test a Custom-Generated Workflow

You can run a custom workflow generated from a SOAP operation to get the output parameters of the operation.
Prerequisites

- Verify that you are logged in to the Orchestrator client as an administrator.
- Verify that you have a connection to a SOAP host from the Inventory view.

Procedure

1. Click the Workflows view in the Orchestrator client.
2. Navigate to the workflow location.
3. Right-click the custom workflow and select Start workflow.
4. Provide the input parameters that the SOAP operation requires.
5. Click Submit to run the workflow.
6. (Optional) In the Logs tab, review the list of available output parameters.

Invoke a SOAP Operation

You can call a SOAP operation directly, without generating a new workflow.

Prerequisites

- Verify that you are logged in to the Orchestrator client as an administrator.
- Verify that you have a connection to a SOAP host from the Inventory view.

Procedure

1. Click the Workflows view in the Orchestrator client.
2. In the workflows hierarchical list, select Library > SOAP and navigate to the Invoke a SOAP operation workflow.
3. Right-click the Invoke a SOAP operation workflow and select Start workflow.
4. Select the SOAP operation from the list of available operations.
5. Provide the input parameters that the SOAP operation requires.
6. Click Submit to run the workflow.
7. (Optional) In the Logs tab, review the list of available output parameters.
Using the AMQP Plug-In

The AMQP plug-in allows you to interact with Advanced Message Queuing Protocol (AMQP) servers also known as brokers. You can define AMQP brokers and queue subscriptions as inventory objects by running configuration workflows, and perform AMQP operations on defined objects.

The plug-in contains a set of standard workflows related to managing AMQP brokers and calling AMQP operations.

This chapter includes the following topics:

- Configuring the AMQP Plug-In
- Using the AMQP Plug-In Standard Workflows

Configuring the AMQP Plug-In

You must use the Orchestrator client to configure the AMQP plug-in.

Configuration Workflows

The Configuration workflow category contains workflows that allow you to manage AMQP brokers.

You can access these workflows from Library > AMQP > Configuration on the Workflows view in the Orchestrator client.

<table>
<thead>
<tr>
<th>Workflow Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Add a broker</td>
<td>Adds an AMQP broker.</td>
</tr>
<tr>
<td>Remove a broker</td>
<td>Removes an AMQP broker.</td>
</tr>
<tr>
<td>Remove a subscription</td>
<td>Removes an AMQP message subscription.</td>
</tr>
<tr>
<td>Subscribe to queues</td>
<td>Creates a new subscription element.</td>
</tr>
<tr>
<td>Update a broker</td>
<td>Updates broker properties.</td>
</tr>
<tr>
<td>Validate a broker</td>
<td>Validate a broker by attempting to start a connection.</td>
</tr>
</tbody>
</table>

Add a Broker

You can run a workflow to add an AMQP broker.
**Procedure**

1. Log in to the Orchestrator client as an administrator.
2. Click the *Workflows* view in the Orchestrator client.
3. In the hierarchical list of workflows, select *Library > AMQP > Configuration* and navigate to the Add a broker workflow.
4. Right-click the Add a broker workflow and select *Start workflow*.
5. Provide the information required for the Add a broker workflow.

<table>
<thead>
<tr>
<th>Option</th>
<th>Action</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name</td>
<td>Type the name of the broker.</td>
</tr>
<tr>
<td>Host</td>
<td>Type the address of the host.</td>
</tr>
<tr>
<td>Port</td>
<td>Type the port of the AMQP broker service. The default port is 5672.</td>
</tr>
<tr>
<td>Virtual host</td>
<td>Type the address of the virtual host. The default value provided is /.</td>
</tr>
<tr>
<td>Use SSL</td>
<td>Select whether to use SSL certificates.</td>
</tr>
<tr>
<td>Accept all certificates</td>
<td>Select whether to accept all SSL certificates without validation.</td>
</tr>
<tr>
<td>User name</td>
<td>Type the user name for the broker.</td>
</tr>
<tr>
<td>Password</td>
<td>Type the password for the broker.</td>
</tr>
</tbody>
</table>

6. Click *Submit* to run the workflow.

After the workflow runs successfully, the AMQP broker appears in the *Inventory* view.

**What to do next**

You can run a Validate a broker workflow. If an error occurs, use the Update a broker workflow to change the broker’s properties before validating again.

**Subscribe to Queues**

You can run a workflow to create a new subscription elements.

**Prerequisites**

- Verify that you are logged in to the Orchestrator client as an administrator.
- Verify that you have a connection to an AMQP broker from the *Inventory* view.
- Verify that the AMQP broker has all queues included in the subscription declared.

**Procedure**

1. Click the *Workflows* view in the Orchestrator client.
2. In the hierarchical list of workflows, select *Library > AMQP > Configuration* and navigate to the Subscribe to queues workflow.
3. Right-click the Subscribe to queues workflow and select *Start workflow*. 
4. In the Name text box, type the name of the queue to display.
5. Select the broker to which you want to add the subscription.
6. Select all the queues for message subscription.
7. Click Submit to run the workflow.

After the workflow runs successfully a child of the broker appears in the Inventory view.

What to do next
You can create a policy.

Update a Broker
You can run a workflow to update the broker properties.

Prerequisites
- Verify that you are logged in to the Orchestrator client as an administrator.
- Verify that you have a connection to an AMQP broker from the Inventory view.

Procedure
1. Click the Workflows view in the Orchestrator client.
2. In the hierarchical list of workflows, select Library > AMQP and navigate to the Update a broker workflow.
3. Right-click the Update a broker workflow and select Start workflow.
4. Select the broker that you want to update.
   Current properties of the broker appear.
5. Edit the properties that you want.
6. Click Submit to run the workflow.

Using the AMQP Plug-In Standard Workflows
The AMQP workflow category contains workflows that allow you to run AMQP operations.

You can access these workflows from Library > AMQP on the Workflows view in the Orchestrator client.

<table>
<thead>
<tr>
<th>Workflow Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Bind</td>
<td>Creates a binding in a specified broker.</td>
</tr>
<tr>
<td>Declare a queue</td>
<td>Adds a queue to a specified broker.</td>
</tr>
<tr>
<td>Declare an exchange</td>
<td>Adds an exchange to a specified broker.</td>
</tr>
<tr>
<td>Delete a queue</td>
<td>Deletes a queue from a specified broker.</td>
</tr>
<tr>
<td>Delete an exchange</td>
<td>Deletes an exchange from a specified broker.</td>
</tr>
</tbody>
</table>
### Declare a Binding

You can run a workflow to create a binding in a specified broker.

**Prerequisites**

- Verify that you are logged in to the Orchestrator client as an administrator.
- Verify that you have a connection to an AMQP broker from the **Inventory** view.

**Procedure**

1. Click the **Workflows** view in the Orchestrator client.
2. In the hierarchical list of workflows, select **Library > AMQP** and navigate to the Bind workflow.
3. Right-click the Bind workflow and select **Start workflow**.
4. Select a broker in which you want create a binding.
5. Provide information about the binding.

<table>
<thead>
<tr>
<th>Option</th>
<th>Action</th>
</tr>
</thead>
<tbody>
<tr>
<td>Queue name</td>
<td>Type the name of the queue.</td>
</tr>
<tr>
<td>Exchange name</td>
<td>Type the name of the exchange.</td>
</tr>
<tr>
<td>Routing key</td>
<td>Type the routing key.</td>
</tr>
</tbody>
</table>

6. Click **Submit** to run the workflow.

### Declare a Queue

You can run a workflow to add a queue to a specified broker.

**Prerequisites**

- Verify that you are logged in to the Orchestrator client as an administrator.
- Verify that you have a connection to an AMQP broker from the **Inventory** view.

**Procedure**

1. Click the **Workflows** view in the Orchestrator client.
2. In the hierarchical list of workflows, select **Library > AMQP** and navigate to the Declare a queue workflow.
3. Right-click the Declare a queue workflow and select **Start workflow**.
4 Select a broker to which you want to add the queue.

5 In the **Name** text box, type the name of the queue to display.

6 Select whether the queue is durable.

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Yes</td>
<td>The queue is removed after a broker restart.</td>
</tr>
<tr>
<td>No</td>
<td>The queue remains after a broker restart.</td>
</tr>
</tbody>
</table>

7 Select whether an exclusive client is set for the specific queue.

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Yes</td>
<td>Sets one client for this specific queue.</td>
</tr>
<tr>
<td>No</td>
<td>Sets more clients for this specific queue.</td>
</tr>
</tbody>
</table>

8 Select whether to automatically delete the queue with activated subscription.

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Yes</td>
<td>Automatically deletes the queue when no more clients are connected to it. The queue remains until at least one client subscribes to it.</td>
</tr>
<tr>
<td>No</td>
<td>Does not automatically delete the queue.</td>
</tr>
</tbody>
</table>

9 Click **Submit** to run the workflow.

**Declare an Exchange**

You can run a workflow to add an exchange in a specified broker.

**Prerequisites**

- Verify that you are logged in to the Orchestrator client as an administrator.
- Verify that you have a connection to an AMQP broker from the **Inventory** view.

**Procedure**

1 Click the **Workflows** view in the Orchestrator client.

2 In the hierarchical list of workflows, select **Library > AMQP** and navigate to the Declare an exchange workflow.

3 Right-click the Declare an exchange workflow and select **Start workflow**.

4 Select a broker to which you want to add the exchange.

5 In the **Name** text box, type a name for the exchange.
6  Select the exchange type.

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>direct</td>
<td>Makes a direct match between the routing key provided in the message and the routing criteria used when a queue is bound to this exchange.</td>
</tr>
<tr>
<td>fanout</td>
<td>Forwards any message sent to this exchange to all queues bound to it. Queues that are bound to this exchange contain no arguments.</td>
</tr>
<tr>
<td>headers</td>
<td>Queues are bound to this exchange with a table of arguments that can contain headers and values. A special argument named x-match determines the matching algorithm.</td>
</tr>
<tr>
<td>topic</td>
<td>Performs a wildcard match between the routing key and the routing pattern specified in the binding.</td>
</tr>
</tbody>
</table>

7  Select whether the exchange is durable.

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Yes</td>
<td>The exchange remains after a broker restart.</td>
</tr>
<tr>
<td>No</td>
<td>The exchange is removed after a broker restart.</td>
</tr>
</tbody>
</table>

8  Select whether to automatically delete the exchange with activated subscription.

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Yes</td>
<td>Automatically deletes the exchange when no more queues are bound to it. The exchange remains until at least one queue is bound to it.</td>
</tr>
<tr>
<td>No</td>
<td>Does not automatically delete the exchange.</td>
</tr>
</tbody>
</table>

9  Click Submit to run the workflow.

**Send a Text Message**

You can run a workflow to send a text message using a specified broker.

**Prerequisites**

- Verify that you are logged in to the Orchestrator client as an administrator.
- Verify that you have a connection to an AMQP broker from the Inventory view.

**Procedure**

1  Click the Workflows view in the Orchestrator client.

2  In the hierarchical list of workflows, select Library > AMQP and navigate to the Send a text message workflow.

3  Right-click the Send a text message workflow and select Start workflow.

4  Select a broker from which you want to send a message.

5  In the Exchange name text box, specify the name of the exchange.

6  In the Routing key text box, specify the routing key.
7 In the Content text box, type the message you want to send.
8 Click Submit to run the workflow.

Delete a Binding
You can run a workflow to delete a binding in a specified broker.

Prerequisites
- Verify that you are logged in to the Orchestrator client as an administrator.
- Verify that you have a connection to an AMQP broker from the Inventory view.

Procedure
1 Click the Workflows view in the Orchestrator client.
2 In the hierarchical list of workflows, select Library > AMQP and navigate to the Unbind workflow.
3 Right-click the Unbind workflow and select Start workflow.
4 Select a broker to remove the binding from.
5 Provide information about the binding.

<table>
<thead>
<tr>
<th>Option</th>
<th>Action</th>
</tr>
</thead>
<tbody>
<tr>
<td>Queue name</td>
<td>Specify the name of the queue.</td>
</tr>
<tr>
<td>Exchange name</td>
<td>Specify the name of the exchange.</td>
</tr>
<tr>
<td>Routing key</td>
<td>Specify the routing key.</td>
</tr>
</tbody>
</table>

6 Click Submit to run the workflow.
Using the SNMP Plug-In

The SNMP plug-in allows vRealize Orchestrator to connect and receive information from SNMP-enabled systems and devices. You can define SNMP devices as inventory objects by running workflows, and perform SNMP operations on the defined objects.

You can use the plug-in to connect to SNMP devices such as routers, switches, network printers, and UPS devices. The plug-in can also receive events from vCenter Server over the SNMP protocol.

The SNMP plug-in provides two methods of communication with the SNMP devices.

- Queries for the values of specific SNMP variables.
- Listening for events (SNMP traps) that are generated from the devices and pushed to the registered SNMP managers.

The plug-in contains a set of standard workflows related to managing SNMP devices, queries, the trap host, and performing SNMP operations. You can also create custom workflows to automate tasks in an SNMP environment.

This chapter includes the following topics:

- Managing SNMP Devices
- Managing SNMP Queries
- Managing the SNMP Trap Host
- Receiving SNMP Traps
- Generic SNMP Request Workflows

Managing SNMP Devices

You can run workflows to register SNMP devices with Orchestrator, edit the settings for existing devices, and unregister devices.

Device Management Workflows

The Device Management workflow category contains workflows that allow you to manage SNMP devices.

You can access these workflows from Library > SNMP > Device Management on the Workflows view in the Orchestrator client.
### Workflow Name | Description
---|---
Edit an SNMP device | Edits the configuration of a registered SNMP device.
Register an SNMP device | Registers an SNMP-enabled device to the plug-in's inventory.
Unregister an SNMP device | Unregisters an SNMP device from the plug-in's inventory.

## Register an SNMP Device

You can run a workflow to register an SNMP device and optionally configure advanced connection parameters.

### Procedure

1. Log in to the Orchestrator client as an administrator.
2. Click the **Workflows** view in the Orchestrator client.
3. In the workflows hierarchical list, select **Library > SNMP > Device Management** and navigate to the Register an SNMP device workflow.
4. Right-click the Register an SNMP device workflow and select **Start workflow**.
5. In the **Device address** text box, type the IP address or DNS name of the SNMP device.

   **Note** To establish a more reliable connection, you should use IP address.

6. (Optional) In the **Name** text box, type a name for the device as you want it to appear in the **Inventory** view.
   
   If you leave the text box blank, the device address is used to generate a name automatically.

7. (Optional) To configure the advanced connection parameters, select **Yes**.
   
   a. In the **Port** text box, specify the connection port.
      
      The default port is 161.
   
   b. From the **Version** drop-down menu, select the SNMP version that you want to use and provide the credentials.

      **Note** The support for SNMPv3 is limited to the AuthPriv security level with MD5 authentication. The DES passphrase is the same as the MD5 password.

8. Click **Submit** to run the workflow.

After the workflow runs successfully, the SNMP device appears in the **Inventory** view.

### What to do next

You can add queries to the SNMP device and run workflows from the **Inventory** view.
Managing SNMP Queries
You can add queries to SNMP devices, run, copy, and edit existing queries, and remove queries from SNMP devices.

Query Management Workflows
The Query Management workflow category contains workflows that allow you to manage SNMP queries.
You can access these workflows from Library > SNMP > Query Management on the Workflows view in the Orchestrator client.

<table>
<thead>
<tr>
<th>Workflow Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Add a query to an SNMP device</td>
<td>Adds a query to an SNMP device.</td>
</tr>
<tr>
<td>Copy an SNMP query</td>
<td>Copies an SNMP query from one device to another.</td>
</tr>
<tr>
<td>Edit an SNMP query</td>
<td>Edits an existing SNMP query.</td>
</tr>
<tr>
<td>Remove a query from an SNMP device</td>
<td>Removes an SNMP query from a device.</td>
</tr>
<tr>
<td>Run an SNMP query</td>
<td>Runs a query against an SNMP device.</td>
</tr>
</tbody>
</table>

Add a Query to an SNMP Device
You can run a workflow to add a query to an SNMP device from the plug-in's inventory.

Prerequisites
- Verify that you are logged in to the Orchestrator client as an administrator.
- Verify that you have a connection to an SNMP device from the Inventory view.

Procedure
1. Click the Workflows view in the Orchestrator client.
2. In the workflows hierarchical list, select Library > SNMP > Query Management and navigate to the Add a query to an SNMP device workflow.
3. Right-click the Add a query to an SNMP device workflow and select Start workflow.
4. Select the device to which you want to add the query.
5. From the Type drop-down menu, select the query type.
6. In the OID text box, type the object identifier of the variable that you want to query.
   The following are example OID values.
   - 1.3.6.1.2.1.1.5.0
   - .1.3.6.1.2.1.1.5.0
iso.3.6.1.2.1.1.5.0

**Note** The plug-in supports only OID values that are numerical or that begin with `iso` and continue with numbers.

7  (Optional) In the **Name** text box, type a name for the query.

If you leave the text box blank, the type and OID parameters are used to generate a name automatically.

8  Click **Submit** to run the workflow.

**What to do next**

You can run workflows on the query from the **Inventory** view.

**Managing the SNMP Trap Host**

You can start and stop the SNMP trap host, and change the port on which Orchestrator listens for SNMP traps.

The SNMP plug-in supports SNMPv1 and SNMPv2c traps.

**Trap Host Management Workflows**

The Trap Host Management workflow category contains workflows that allow you to manage the SNMP trap host.

You can access these workflows from **Library > SNMP > Trap Host Management** on the **Workflows** view in the Orchestrator client.

<table>
<thead>
<tr>
<th>Workflow Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Set the SNMP trap port</td>
<td>Sets the port on which Orchestrator listens for SNMP traps.</td>
</tr>
<tr>
<td>Start the trap host</td>
<td>Orchestrator starts listening for SNMP traps.</td>
</tr>
<tr>
<td>Stop the trap host</td>
<td>Orchestrator stops listening for SNMP traps.</td>
</tr>
</tbody>
</table>

**Set the SNMP Trap Port**

You can run a workflow to set the port on which Orchestrator listens for SNMP traps.

The default port for SNMP traps is 162. However, on Linux systems, you can open ports below 1024 only with superuser privileges. To ensure better compatibility, the default port for listening to SNMP traps in the SNMP plug-in is set to 4000.

**Prerequisites**

- Verify that you are logged in to the Orchestrator client as an administrator.
- Verify that you have a connection to an SNMP device from the **Inventory** view.
Procedure

1. Click the Workflows view in the Orchestrator client.
2. In the workflows hierarchical list, select Library > SNMP > Trap Host Management and navigate to the Set the SNMP trap port workflow.
3. Right-click the Set the SNMP trap port workflow and select Start workflow.
4. In the Port text box, type the port number on which Orchestrator should listen for SNMP traps.
5. Click Submit to run the workflow.

The workflow stops the trap host, sets the new port, and starts the trap host again.

Receiving SNMP Traps

The SNMP plug-in can receive SNMP traps by running a workflow, which waits for a single trap message, or with a policy, which can handle traps continuously. The plug-in supports SNMPv1 and SNMPv2c traps.

Wait for a Trap on an SNMP Device

You can run a workflow that waits to receive an SNMP trap from a specified device.

This workflow features a trigger, which stops the run of the workflow and waits for an SNMP trap before continuing. When a trap is received, the workflow run resumes. You can use the workflow as part of more complex workflows, or as a sample that you can customize or extend for a specific need.

Prerequisites

- Verify that you are logged in to the Orchestrator client as an administrator.
- Verify that you have a connection to an SNMP device from the Inventory view.

Procedure

1. Click the Workflows view in the Orchestrator client.
2. In the workflows hierarchical list, select Library > SNMP and navigate to the Wait for a trap on an SNMP device workflow.
3. Right-click the Wait for a trap on an SNMP device workflow and select Start workflow.
4. Select the device on which you want to wait for an SNMP trap.
5. (Optional) In the OID text box, type the object identifier of a specific trap.

   **Note** If you leave the text box empty, the workflow run resumes after receiving any trap from the specified SNMP device.

6. Click Submit to run the workflow.

Set a Trap Policy

You can set a policy to continuously listen for traps from an SNMP device.
Prerequisites

- Verify that you are logged in to the Orchestrator client as an administrator.
- Verify that you have a connection to an SNMP device from the Inventory view.

Procedure

1. Click the Policy Templates view in the Orchestrator client.
2. In the workflows hierarchical list, select Library > SNMP and navigate to the SNMP Trap policy template.
3. Right-click the SNMP Trap policy template and select Apply Policy.
4. In the Policy name text box, type a name for the policy that you want to create.
5. (Optional) In the Policy description text box, type a description for the policy.
6. Select an SNMP device for which to set the policy.
7. Click Submit to create the policy.
8. On the Policies view, right-click the policy that you created and select Start policy.

The trap policy starts to listen for SNMP traps.

What to do next

You can edit the trap policy.

Edit a Trap Policy

You can edit a trap policy to customize it for a specific use case. When you edit a trap policy, you can change its priority and startup settings, as well as customize the scripting and permissions associated with the policy.

Prerequisites

- Verify that you are logged in to the Orchestrator client as an administrator.
- Verify that you have a connection to an SNMP device from the Inventory view.

Procedure

1. Click the Policies view in the Orchestrator client.
2. If the policy that you want to edit is running, right-click the policy and select Stop policy.
3. Right-click the policy and select Edit.
4. From the General tab, edit the startup settings, priority, and description of the policy.
5. (Optional) From the Scripting tab, you can associate a specific workflow or scripting code with the policy, for integration in a more complex scenario.

   You can set the policy to trigger a custom workflow when a trap is received.
6 (Optional) From the Permissions tab, you can modify the access rights.

You can give a user or a group permissions to start the policy, without giving permissions to edit the policy.

7 Click Save and close to apply the edited settings.

8 On the Policies view, right-click the policy that you edited and select Start policy.

**Generic SNMP Request Workflows**

The SNMP workflow category contains workflows that allow you to perform basic SNMP requests without having to create a query.

You can access these workflows from Library > SNMP on the Workflows view in the Orchestrator client.

<table>
<thead>
<tr>
<th>Workflow Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Get bulk SNMP values</td>
<td>Runs a GET BULK query against an SNMP device.</td>
</tr>
<tr>
<td>Get next SNMP value</td>
<td>Runs a GET NEXT query against an SNMP device.</td>
</tr>
<tr>
<td>Get SNMP value</td>
<td>Runs a GET query against an SNMP device.</td>
</tr>
<tr>
<td>Send an SNMP trap</td>
<td>Sends an SNMP trap to a specified address.</td>
</tr>
<tr>
<td>Wait for a trap on an SNMP device</td>
<td>Waits to receive an SNMP trap from a specified device.</td>
</tr>
</tbody>
</table>
Using the Active Directory Plug-In

The Active Directory plug-in (VMware vRealize Orchestrator plug-in for Microsoft Active Directory) allows interaction between vRealize Orchestrator and Microsoft Active Directory. You can use the plug-in to run Orchestrator workflows that automate Active Directory processes.

The plug-in contains a set of standard workflows. You can also create custom workflows that implement the plug-in API to automate tasks in your Active Directory environment.

This chapter includes the following topics:

- Configuring the Active Directory Plug-In
- Using the Active Directory Plug-In Workflow Library

Configuring the Active Directory Plug-In

To connect to a Microsoft Active Directory instance by using the Active Directory plug-in, you must configure the connection parameters for the Microsoft Active Directory instance.

You can configure Active Directory by running the configuration workflows included in the plug-in.

Active Directory Configuration Workflows

The Configuration workflow category of the Active Directory plug-in contains workflows that allow you to configure Active Directory servers and manage SSL certificates.

You can access these workflows from Library > Microsoft > Active Directory > Configuration in the Workflows view of the Orchestrator client.

<table>
<thead>
<tr>
<th>Workflow Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Add an Active Directory server</td>
<td>Configures a Microsoft Active Directory server.</td>
</tr>
<tr>
<td>Configure Active Directory plug-in options</td>
<td>Configures the search limitation options of the Active Directory plug-in.</td>
</tr>
<tr>
<td>Configure Active Directory server (Deprecated)</td>
<td>Creates or updates the default Active Directory server configuration. Use Update an Active Directory server.</td>
</tr>
<tr>
<td>Remove an Active Directory server</td>
<td>Removes an Active Directory server configuration.</td>
</tr>
<tr>
<td>Reset configuration (Deprecated)</td>
<td>Deletes the default Active Directory server configuration. Use Remove an Active Directory server.</td>
</tr>
<tr>
<td>Update an Active Directory server</td>
<td>Modifies an existing Active Directory server configuration.</td>
</tr>
</tbody>
</table>
Using the Active Directory Plug-In Workflow Library

The Active Directory plug-in workflow library contains workflows that allow you to run automated processes related to the management of Microsoft Active Directory objects.

The workflows are grouped into categories depending on object type. You can integrate standard workflows from the workflow library in custom workflows.

Using the Active Directory Plug-In Inventory

The Active Directory plug-in exposes all objects in the connected Microsoft Active Directory instance in the **Inventory** view. You can use the **Inventory** view to add authorization elements or to run workflows on Microsoft Active Directory objects.

To display the workflows that are available for an inventory object, navigate to **Tools > User preferences > Inventory** and select the **Use contextual menu in inventory** check box. After the option is enabled, when you right-click an object in the Orchestrator inventory, all available workflows for the object are displayed.

Access the Active Directory Plug-In Workflow Library

You must use the Orchestrator client to access the elements from the Active Directory plug-in workflow library.

**Procedure**

1. Log in to the Orchestrator client as an administrator.
2. Click the **Workflows** view.
3. In the hierarchical list, expand Library > Microsoft > Active Directory and expand the selection.

Active Directory Plug-In Workflows

The Active Directory plug-in contains a set of standard workflows that cover the most common LDAP functionality. You can use the workflows as building blocks for creating complex custom solutions. By combining standard workflows, you can automate multistep processes in the Active Directory environment.

**Computer Workflows**

The Computer workflow category contains workflows related to Active Directory computer management.

You can access these workflows from Library > Microsoft > Active Directory > Computer.

<table>
<thead>
<tr>
<th>Workflow Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Create a computer in a group</td>
<td>Creates an Active Directory computer in a group.</td>
</tr>
<tr>
<td>Create a computer in an organizational unit</td>
<td>Creates an Active Directory computer in an organizational unit.</td>
</tr>
<tr>
<td>Destroy a computer</td>
<td>Deletes a computer from an Active Directory instance.</td>
</tr>
</tbody>
</table>
Workflow Name | Description
--- | ---
Destroy a computer and delete its subtree | Deletes a computer from an Active Directory instance and all objects within the computer subtree.
Disable a computer | Disables a computer from an Active Directory instance.
Enable a computer | Enables a computer in an Active Directory instance.

Organizational Unit Workflows

The Organizational Unit workflow category contains workflows related to Active Directory organizational unit management.

You can access these workflows from Library > Microsoft > Active Directory > Organizational Unit.

Workflow Name | Description
--- | ---
Create an organizational unit | Creates an organizational unit in an existing organizational unit.
Destroy an organizational unit | Deletes an organizational unit from an Active Directory instance.
Destroy an organizational unit and delete its subtree | Deletes an organizational unit from an Active Directory instance and all objects within the organizational unit subtree.

User Workflows

The User workflow category contains workflows related to Active Directory user management.

You can access these workflows from Library > Microsoft > Active Directory > User.

Workflow Name | Description
--- | ---
Add a user to a user group | Adds one user as a member of a user group.
Change a user password | Changes the password for a user. SSL connection is required, and the password must meet the Active Directory restrictions.
Create a user in a group | Creates a user without specifying a password. The password must be changed at the next login. Domain policies must allow users to have empty passwords.
Create a user in an organizational unit | Creates a user in an organizational unit. If SSL connection is disabled, you cannot specify a password. Domain policies must allow users to have empty passwords.
Create a user with a password in a group | Creates a user and sets a password for the user. The password can be changed at the next login.
Create a user with a password in an organizational unit | Creates a user in an organizational unit and sets a password for the user. The password can be changed at the next login. If SSL connection is disabled, you cannot specify a password.
Destroy a user | Deletes a user from an Active Directory instance.
Disable a user | Disables a user from an Active Directory instance.
Enable a user | Enables a user in an Active Directory instance.
Remove a user from a user group | Removes a user from a user group.
User Group Workflows

The User Group workflow category contains workflows related to Active Directory user group management.

You can access these workflows from Library > Microsoft > Active Directory > User Group.

<table>
<thead>
<tr>
<th>Workflow Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Add computers to group members</td>
<td>Adds one or more computers as members of a user group.</td>
</tr>
<tr>
<td>Add groups to group members</td>
<td>Adds one or more user groups as members of a user group.</td>
</tr>
<tr>
<td>Add users to group members</td>
<td>Adds one or more users as members of a user group.</td>
</tr>
<tr>
<td>Create a user group in a group</td>
<td>Creates a user group in an existing container (group).</td>
</tr>
<tr>
<td>Create a user group in a group and set attribute</td>
<td>Creates a user group in an existing container (organizational unit) and sets the Group name (pre-Windows 2000) attribute.</td>
</tr>
<tr>
<td>“Group name (pre-Windows 2000)”</td>
<td></td>
</tr>
<tr>
<td>Create a user group in an organizational unit</td>
<td>Creates a user group in an existing container (organizational unit).</td>
</tr>
<tr>
<td>Destroy a user group</td>
<td>Deletes a user group from an Active Directory instance.</td>
</tr>
<tr>
<td>Remove computers from group members</td>
<td>Removes one or more computers from a user group.</td>
</tr>
<tr>
<td>Remove groups from group members</td>
<td>Removes one or more user groups from a user group.</td>
</tr>
<tr>
<td>Remove users from group members</td>
<td>Removes one or more users from a user group.</td>
</tr>
</tbody>
</table>
Using the Dynamic Types Plug-In

The Orchestrator Dynamic Types plug-in lets you define dynamic types, create objects of these types, and set relations between them.

The definition of a dynamic type contains the descriptions of its properties and a set of finder workflows and actions which can be used to find dynamic objects of this type. Runtime instances of dynamic types are called dynamic objects. You can run workflows on the dynamic objects you create and perform different operations on them.

Each dynamic type must be defined in a namespace. Namespaces are helper dynamic objects that let you group dynamic types in containers.

You can use the Dynamic Types plug-in together with the HTTP-REST plug-in to integrate third-party REST API services into the Orchestrator and expose third-party objects as Orchestrator types.

1 Define a new dynamic type and its properties by running the Define Namespace and Define Type workflows from the Dynamic Types plug-in. In result, you obtain a set of finder and inventory workflows for finding objects of the new dynamic type and their relations with other objects.

2 Modify the new finder and inventory workflows, so that they receive their input from the third-party REST API.
   a Create REST operations by using the Add a REST Operation workflow from the HTTP-REST plug-in and map these operations to the corresponding REST API methods.
   b Modify the finder and inventory workflows to invoke these REST operations and consume their outputs.

Dynamic Types Configuration Workflows

The workflows in the Configuration package of the Dynamic Types plug-in let you create dynamic types, export and import type definitions from an XSD file, and define relations between the dynamic types you created.

You can access these workflows from Library > Dynamic Types > Configuration in the Workflows view of the Orchestrator client.

<table>
<thead>
<tr>
<th>Workflow Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Define Namespace</td>
<td>Defines a new namespace.</td>
</tr>
<tr>
<td>Define Relation</td>
<td>Defines a new relation between types.</td>
</tr>
<tr>
<td><strong>Workflow Name</strong></td>
<td><strong>Description</strong></td>
</tr>
<tr>
<td>--------------------------------</td>
<td>-------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>Define Type</td>
<td>Defines a new type within a given namespace.</td>
</tr>
<tr>
<td>Export Configuration As Package</td>
<td>Exports a Dynamic Types definition configuration as a file-based configuration. The exported package can be used for importing to other servers.</td>
</tr>
<tr>
<td>Import Configuration From Package</td>
<td>Imports a file-based configuration to the plug-in configuration.</td>
</tr>
<tr>
<td>Import Type Definitions From XSD</td>
<td>Imports type definitions from an XSD file.</td>
</tr>
<tr>
<td>Remove Namespace</td>
<td>Removes a namespace.</td>
</tr>
<tr>
<td>Remove Relation</td>
<td>Removes a relation.</td>
</tr>
<tr>
<td>Remove Type</td>
<td>Removes a type.</td>
</tr>
<tr>
<td>Update Namespace</td>
<td>Updates a namespace.</td>
</tr>
<tr>
<td>Update Type</td>
<td>Updates a type.</td>
</tr>
</tbody>
</table>
The PowerShell plug-in workflow library contains workflows that allow you to manage PowerShell hosts and run custom PowerShell operations.

You can use the Inventory view in the Orchestrator client to manage the available PowerShell resources. You can use the scripting API of the plug-in to develop custom workflows.

This chapter includes the following topics:

- Using the PowerShell Plug-In Inventory
- Configuring the PowerShell Plug-In
- Running PowerShell Scripts
- Generating Actions
- Passing Invocation Results Between Actions
- PowerCLI Integration with the PowerShell Plug-In
- Sample Workflows
- Access the PowerShell Plug-In API
- Working with PowerShell Results
- Examples of Scripts for Common PowerShell Tasks
- Troubleshooting

Using the PowerShell Plug-In Inventory

The PowerShell plug-in exposes all objects in the connected PowerShell hosts in the Inventory view. You can use the Inventory view to add authorization elements or to run workflows on PowerShell objects.

Within the inventory of the plug-in, you can monitor PowerShell hosts and their snap-ins and cmdlets. Each remote host can contain snap-ins and each snap-in can contain cmdlets.

To display the workflows that are available for an inventory object, navigate to Tools > User preferences > Inventory and select the Use contextual menu in inventory check box. After the option is enabled, when you right-click an object in the Orchestrator inventory, all available workflows for the object are displayed.
Configuring the PowerShell Plug-In

You must use the Orchestrator client to configure the PowerShell plug-in.

Configuration Workflows

The Configuration workflow category contains workflows that allow you to manage PowerShell hosts.

You can access these workflows from Library > PowerShell > Configuration on the Workflows view in the Orchestrator client.

<table>
<thead>
<tr>
<th>Workflow Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Add a PowerShell host</td>
<td>Adds a PowerShell host to the plug-in inventory.</td>
</tr>
<tr>
<td>Remove a PowerShell host</td>
<td>Removes a PowerShell host from the plug-in inventory.</td>
</tr>
<tr>
<td>Update a PowerShell host</td>
<td>Updates the specified PowerShell host in the plug-in inventory.</td>
</tr>
<tr>
<td>Validate a PowerShell host</td>
<td>Validates the configuration of the specified PowerShell host.</td>
</tr>
</tbody>
</table>

Configure Kerberos Authentication

You can use Kerberos authentication when you add a host.

The krb5.conf file contains the following information:

- Kerberos configuration information
- Locations of Key Distribution Centers (KDC) and administration servers for the Kerberos realms of interest
- Default values for the current realm and for Kerberos applications
- Mappings of host names onto Kerberos realms

Procedure

- Create a krb5.conf file and save it to the following location.

<table>
<thead>
<tr>
<th>Operating System</th>
<th>Path</th>
</tr>
</thead>
<tbody>
<tr>
<td>Windows</td>
<td>C:\Program Files\Common Files\VMware\VMware vCenter Server - Java Components\lib\security\</td>
</tr>
<tr>
<td>Linux</td>
<td>/usr/java/jre-vmware/lib/security/</td>
</tr>
</tbody>
</table>

A krb5.conf file has the following structure:

```
[libdefaults]
default_realm = YOURDOMAIN.COM
udp_preference_limit = 1
[realms]
YOURDOMAIN.COM = {
kdc = kdc.yourdomain.com
default_domain = yourdomain.com
```
Note  The Kerberos authentication requires a Fully Qualified Domain Name (FQDN) host address.

Important  When you add or modify the krb5.conf file, you must restart the Orchestrator server service.

Add a PowerShell Host

You can run a workflow to add a PowerShell host and configure the host connection parameters. You can set up a connection to a remote or a local PowerShell host.

Procedure

1  Log in to the Orchestrator client as an administrator.

2  Click the Workflows view in the Orchestrator client.

3  In the workflows hierarchical list, expand Library > PowerShell > Configuration and navigate to the Add a PowerShell host workflow.

4  Right-click the Add a PowerShell host workflow and select Start workflow.

5  In the Name text box, type the name of the host.

6  In the Host / IP text box, type the address of the host.

   Note  The Kerberos authentication requires a Fully Qualified Domain Name (FQDN) host address.

7  (Optional) In the Port text box, type the port of the host.

8  Select the PowerShell host type that the plug-in connects to.

    a  Select a transport protocol.

       Note  If you use the HTTPS transport protocol, the certificate of the remote PowerShell host is imported into the Orchestrator keystore.

    b  Select the authentication type.

       Important  If you want to use Kerberos authentication, you must enable it on the WinRM service.
Select the type of session mode that the plug-in uses to connect to the PowerShell host.

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Shared Session</td>
<td>The plug-in uses shared credentials to connect to the remote host. You must provide the PowerShell host credentials for the shared session.</td>
</tr>
<tr>
<td>Session per User</td>
<td>The Orchestrator client retrieves credentials from the user who is logged in. You must log in with a <code>user@domain</code> format to Orchestrator to use the Session per User mode.</td>
</tr>
</tbody>
</table>

Click Submit to run the workflow.

After the workflow runs successfully, the PowerShell host appears in the Inventory view.

### Running PowerShell Scripts

You can run workflows to invoke an external or custom script on a PowerShell host.

#### Invoke a PowerShell Script

You can run an existing or custom PowerShell script on a host in the plug-in's inventory.

**Prerequisites**

- Verify that you are logged in to the Orchestrator client as an administrator.
- Verify that you have a connection to a PowerShell host from the Inventory view.

**Procedure**

1. Click the Workflows view in the Orchestrator client.
2. In the workflows hierarchical list, select Library > PowerShell and navigate to the Invoke a PowerShell script workflow.
3. Right-click the Invoke a PowerShell script workflow and select Start workflow.
4. Select a PowerShell host on which to run the script.
5. In the Script text box, type or paste the PowerShell script that you want to run.
6. Click Submit to run the workflow.

#### Invoke an External Script

You can run an external PowerShell script on a host in the plug-in's inventory.

External PowerShell scripts are contained in `.ps1` files. The `.ps1` file that you want to run must be located on the PowerShell host.

**Prerequisites**

- Verify that you are logged in to the Orchestrator client as an administrator.
- Verify that you have a connection to a PowerShell host from the Inventory view.
Verify that you have access to other .ps1 files that the script might reference.

**Procedure**

1. Click the **Workflows** view in the Orchestrator client.

2. In the workflows hierarchical list, select Library > PowerShell and navigate to the Invoke an external script workflow.

3. Right-click the Invoke an external script workflow and select **Start workflow**.

4. Select a PowerShell host on which to run the script.

5. In the **Name** text box, type the filename of the external .ps1 script that you want to run.

   **Note** If the .ps1 file is not in the default folder, you must type the absolute filepath. You can use system environment variables to specify script paths. For example, $env:HOMEPATH\test1.ps1.

6. In the **Arguments** text box, type the script arguments.

   The valid syntax is the same as that used in the PowerShell console.

7. Click **Submit** to run the workflow.

**Generating Actions**

You can run workflows to generate actions based on a PowerShell script or a PowerShell cmdlet. You can use the generated actions as building blocks for custom workflows.

**Generate an Action from a PowerShell Script**

You can run a workflow to generate an action from a PowerShell script that you provide. You can optionally generate a sample workflow that can run the generated action.

You can customize the script of the action that you generate by using placeholders. For each placeholder, the workflow creates a corresponding action parameter of type `string` in the generated action. When you run the action, you can provide an actual value as the action parameter. The value that you provide replaces the placeholder.

**Prerequisites**

- Verify that you are logged in to the Orchestrator client as an administrator.
- Verify that you have a connection to a PowerShell host from the **Inventory** view.

**Procedure**

1. Click the **Workflows** view in the Orchestrator client.

2. In the workflows hierarchical list, select Library > PowerShell > Generate and navigate to the Generate an action from a PowerShell script workflow.

3. Right-click the Generate an action from a PowerShell script workflow and select **Start workflow**.
4 In the **Script** text box, type or paste the PowerShell script from which to generate the action.

**Note** You can use `{#ParamName#}` as a placeholder for user input. If the placeholder is of type `string`, you must use double quotes to pass the value of the placeholder to the action.

The following script is an example of how to link the generated action parameter to a script parameter.

```powershell
param($name={#ParamName#})
echo $name;
```

5 In the **Name** text box, type a name for the action that you want to generate.

6 Select an existing module in which to generate the action.

7 Select whether to generate a workflow.

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
</table>
| Yes    | Generates a sample workflow that can run the generated action. You must select a folder in which to generate the workflow.  
**Note** The name of the generated workflow consists of the predefined string `Invoke Script` and the name of the generated action. |
| No     | A sample workflow is not generated. |

8 Click **Submit** to run the workflow.

**What to do next**

You can integrate the generated action in custom workflows.

**Generate an Action for a PowerShell Cmdlet**

You can run a workflow to generate an action for a PowerShell cmdlet and parameter set that you provide. With this action, you can use PowerShell functionality in Orchestrator. You can optionally generate a sample workflow that can run the generated action.

You can use a large set of data types with the PowerShell script engine. The data types that you can use include primitive types such as `Integer`, `Boolean`, `Char`, any type available from the .NET assembly, or user-defined types. When generating actions based on PowerShell cmdlet definitions, the input and output cmdlet parameters are represented by types that the Orchestrator platform supports. The PowerShell plug-in defines the type mappings. In general, primitive types are mapped to the corresponding Orchestrator types, and complex types are represented by the `PowerShellRemotePSObject` object.

**Prerequisites**

- Verify that you are logged in to the Orchestrator client as an administrator.
- Verify that you have a connection to a PowerShell host from the **Inventory** view.
Procedure

1  Click the **Workflows** view in the Orchestrator client.

2  In the workflows hierarchical list, select **Library > PowerShell > Generate** and navigate to the Generate an action for a PowerShell cmdlet workflow.

3  Right-click the Generate an action for a PowerShell cmdlet workflow and select **Start workflow**.

4  Select a PowerShell cmdlet to run when using the action that you generate.

5  Select a parameter set for the cmdlet.

   The parameter set definition values appear in the **Parameter set definition** text box.

   **Note** You cannot modify the parameter set definition values by editing the string in the **Parameter set definition** text box. You can review the string for information about the parameters that the parameter set contains.

6  In the **Name** text box, type a name for the action that you want to generate.

7  Select an existing module in which to generate the action.

8  Select whether to generate a workflow.

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Yes</td>
<td>Generates a sample workflow that can run the generated action. You should select a folder in which to generate the workflow.</td>
</tr>
<tr>
<td></td>
<td><strong>Note</strong> The name of the generated workflow consists of the predefined string Execute Cmdlet and the name of the generated action.</td>
</tr>
<tr>
<td>No</td>
<td>A sample workflow is not generated.</td>
</tr>
</tbody>
</table>

9  Click **Submit** to run the workflow.

What to do next

You can integrate the generated action in custom workflows.

**Passing Invocation Results Between Actions**

The PowerShell plug-in supports passing of results as parameters from one PowerShell script invocation to another. To pass results correctly, both invocations must happen in the same session.

**PowerCLI Integration with the PowerShell Plug-In**

You can use functionality that is available in a third-party snap-in, such as VMware vSphere PowerCLI, with the PowerShell plug-in.

To use the third-party snap-in functionality, the snap-in must be available on the PowerShell host. To load the snap-in in the current session, you must also call the AddPsSnapin action. When using PowerCLI, you must set the name of the snap-in to **VMware.VimAutomation.Core**.
The PowerShell plug-in does not provide pre-generated actions for third-party snap-ins. You can generate actions for third-party snap-ins by running the Generate an action for a PowerShell cmdlet workflow. See Generate an Action for a PowerShell Cmdlet.

The `com.vmware.library.powershell.converter` package contains basic building blocks that allow conversion from a `VC:<SomeObjectType>` object, to the corresponding object from PowerCLI. This feature allows workflows from the vCenter Server plug-in to interact with workflows from the PowerShell plug-in and to pass parameters between the two plug-ins.

**Converter Workflows**

You can use the sample workflows from the Converter workflow category to test the integration between the PowerShell plug-in and PowerCLI. To test the integration, PowerCLI must be installed on the PowerShell host.

The Converter sample workflows demonstrate the conversion functionality available in the plug-in.

**Note** The PowerShell plug-in does not support all types that are available in PowerCLI and the vCenter Server plug-in. Unsupported types return an exception.

You can access these workflows from Library > PowerShell > Samples > Converter on the Workflows view in the Orchestrator client.

<table>
<thead>
<tr>
<th>Workflow Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Convert PSObject to vCO object</td>
<td>Converts <code>PowerShellRemotePSObject</code> to <code>VC:&lt;SomeObjectType&gt;</code>.</td>
</tr>
<tr>
<td>Convert PSObject to vCO object to PSObject</td>
<td>Converts <code>PowerShellRemotePSObject</code> to <code>VC:&lt;SomeObjectType&gt;</code> and the reverse.</td>
</tr>
<tr>
<td>Convert vCO object to PSObject</td>
<td>Converts <code>VC:&lt;SomeObjectType&gt;</code> to <code>PowerShellRemotePSObject</code>.</td>
</tr>
</tbody>
</table>

**Sample Workflows**

The Samples workflow category contains workflows that allow you to test basic use cases.

You can access these workflows from Library > PowerShell > Samples on the Workflows view in the Orchestrator client.

<table>
<thead>
<tr>
<th>Workflow Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Invoke a script via API</td>
<td>Demonstrates how to call a PowerShell script through the available scripting API.</td>
</tr>
<tr>
<td>List directory content</td>
<td>Lists the contents of a directory.</td>
</tr>
<tr>
<td>Pipeline execution example</td>
<td>Demonstrates how you can run multiple cmdlets arranged into a pipe.</td>
</tr>
<tr>
<td>Toggle virtual machine state</td>
<td>Toggles the power state of a virtual machine.</td>
</tr>
</tbody>
</table>

**Access the PowerShell Plug-In API**

Orchestrator provides an API Explorer to allow you to search the PowerShell plug-in API and see the documentation for JavaScript objects that you can use in scripted elements.
Procedure

1. Log in to the Orchestrator client as an administrator.

2. Access the API Explorer from either the Orchestrator client or from the Scripting tabs of the workflow, policy, and action editors.
   - To access the API Explorer from the Orchestrator client, click Tools > API Explorer in the Orchestrator client toolbar.
   - To access the API Explorer from the Scripting tabs of the workflow, policy, and action editors, click Search API on the left.

3. To expand the hierarchical list of PowerShell plug-in API objects, double-click the PowerShell module in the left pane.

What to do next

You can copy code from API elements and paste it into scripting boxes. For more information about API scripting, see Developing with VMware vRealize Orchestrator.

Working with PowerShell Results

You can use objects from the API of the PowerShell plug-in to work with results that Windows PowerShell returns.

You can use the methods from the PowerShellInvocationResult class to get information about a script that you run.

<table>
<thead>
<tr>
<th>Method</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>getErrors()</td>
<td>Returns a list of errors reported by the PowerShell engine during script invocation.</td>
</tr>
<tr>
<td>getInvocationState()</td>
<td>Status of the script. The possible values are Completed or Failed.</td>
</tr>
<tr>
<td>getHostOutput()</td>
<td>Output of the script as it appears on the PowerShell console.</td>
</tr>
<tr>
<td>getResults()</td>
<td>Objects returned by the PowerShell engine. The returned object is of type PowershellRemotePSObject.</td>
</tr>
</tbody>
</table>

PowershellRemotePSObject is a remote representation of objects returned by the PowerShell engine. PowershellRemotePSObject contains XML serialization of the result that can be accessed by calling the getXml() method.

The PowerShell plug-in also provides an object model that wraps the XML result and provides easier access to particular object properties. The getRootObject() method provides access to the object model. In general, the getRootObject() method maps the PowerShell types to types available in Orchestrator, by using the following rules.

- If the returned object is of a primitive PowerShell type, the object is mapped to the corresponding Orchestrator primitive type.
- If the returned object is of type collection, the object is represented as ArrayList.
- If the returned object is of type dictionary, the object is represented as Hashtable.
Examples of Scripts for Common PowerShell Tasks

You can cut, paste, and edit the JavaScript examples to write scripts for common PowerShell tasks.
For more information about scripting, see the *vRealize Orchestrator Developer's Guide*.

**Example: Run a PowerShell Script Through the API**

You can use JavaScript to run a PowerShell script through the API of the plug-in.
This example script performs the following actions.
- Opens a session to a PowerShell host.
- Provides a script to run.
- Checks invocation results.
- Closes the session.

```javascript
var sess;
try {
    //Open session to PowerShell host
    var sess = host.openSession()
    //Set executed script
    var result = sess.invokeScript('dir')

    //Check for errors
    if (result.invocationState == 'Failed'){
        throw "PowerShellInvocationError: Errors found while executing script \n" +
        result.getErrors();
    }
    //Show result
    System.log( result.getHostOutput() );
} catch (ex){
    System.error (ex)
} finally {
    if (sess) {
        //Close session
        host.closeSession( sess.getSessionId() );
    }
}
```

**Example: Work with Result**

You can use JavaScript to work with the result of a PowerShell script run.
This example script performs the following actions.
- Checks the invocation state.
- Extracts a value from the result.
Checks the RemotePSObject type.

```javascript
var sess = host.openSession()
    sess.addCommandFromString("dir " + directory)
var invResult = sess.invokePipeline();
//Show result
System.log( invResult.getHostOutput() );

//Check for errors
if (invResult.invocationState == 'Failed'){
    System.error(invResult.getErrors());
} else {
    //Get PowerShellRemotePSObject
    var psObject = invResult.getResults();
    var directories = psObject.getRootObject();

    var isList = directories instanceof Array
    if ( isList ){
        for (idx in directories){
            var item = directories[idx];
            if ( item.instanceOf('System.IO.FileInfo') ) { //Check type of object
                System.log( item.getProperty('FullName') ); //Extract value from result
            }
        }
    } else {
        System.log( directories.getProperty('FullName') ); //Extract value from result
    }
}
host.closeSession( sess.getSessionId());
```

Example: Connect with Custom Credentials

You can use JavaScript to connect to a PowerShell host with custom credentials.

```javascript
var sess;
try {
    sess = host.openSessionAs(userName, password);

    var invResult = sess.invokeScript('$env:username');

    //Check for errors
    if (invResult.invocationState == 'Failed'){
        System.error(invResult.getErrors());
    } else {
        //Show result
        System.log( invResult.getHostOutput() );
    }
} catch (ex){
    System.error (ex)
} finally {
```
if (sess) {
    host.closeSession( sess.getSessionId());
}

**Troubleshooting**

If you encounter problems when running or using PowerShell plug-in, you can use a troubleshooting topic to understand and solve the problem, if there is a workaround.

**Servers Not Found in Kerberos Database**

After you add servers with Kerberos authentication, the servers might not be found because they are not added correctly.

**Problem**

When you try to connect to a server, the server is not found in Kerberos database.

```
No valid credentials provided (Mechanism level: No valid
credentials provided (Mechanism level: Server not found in Kerberos
database (7)))
```

**Cause**

This error might be caused by domain or mapping problems, or because of a DNS problem where the service principal name is not built correctly. Kerberos authentication does not work when the destination is an IP address.

**Solution**

When you add a PowerShell host using the Kerberos authentication type a DNS or NetBIOS destination.

**Unable to Obtain a Kerberos Ticket**

When you provide wrong credentials, the plug-in fails to obtain a Kerberos ticket.

**Problem**

You are unable to add a host to the plug-in’s inventory and the result is the following error message.

```
Pre-authentication information was invalid (24)
```

**Cause**

You have provided wrong credentials.

**Solution**

Provide the correct credentials.
Kerberos Authentication Fails Due to Time Differences

System time differences in the environment that uses Kerberos configuration might lead to authentication failure.

Problem
Attempts to use Kerberos for initial authentication of a host or for resource access fail and the following error message appears.

Clock Skew

Cause
If the system time on the computers in the environment deffers with more than 5 minutes from the client computer or from one another, the Kerberos authentication fails.

Solution
Synchronize the system times in the environment.

Kerberos Authentication Session Mode Fails
When you use Kerberos authentication with Shared Session or Session per User, adding the PowerShell host might fail.

Problem
When you attempt to add a PowerShell host to the plug-in's inventory using Shared Session or Session per User, the workflow fails, and the following error appears.

Null realm name (601) - default realm not specified (Dynamic Script Module name : addPowerShellHost#16)

Cause
The default realm is not specified in the Kerberos configuration file krb5.conf neither provided as a part of the user name.

Solution
Provide a default realm in your Kerberos configuration file or specify the realm in your user name when authenticating Kerberos.

Unable to Reach a Key Distribution Center for a Realm
Any misspelling in the krb5.conf file might cause a failure when you add a host.
Problem

When you are adding a host, the Kerberos authentication is unable to reach a Key Distribution Center (KDC) for yourrealm.

| Cannot get kdc for realm YOURREALM.COM |

Cause

The libdefaults and realms sections in the krb5.conf file might be misspelled.

Solution

Verify that the libdefaults and realms sections in your krb5.conf file are spelled correctly.
Using the Multi-Node Plug-In

The Multi-Node plug-in workflow library contains workflows for hierarchical orchestration, management of Orchestrator instances, and scale-out of Orchestrator activities.

This chapter includes the following topics:

- Introduction to the vRealize Orchestrator Multi-Node Plug-In
- Configuring the Multi-Node Plug-In
- Using Proxy Workflows
- Using the Multi-Node Plug-In Inventory
- Access the Multi-Node Plug-In API
- Multi-Node Plug-In Use Cases

Introduction to the vRealize Orchestrator Multi-Node Plug-In

The Multi-Node plug-in creates a master-slave relation between vRealize Orchestrator servers, which extends in the areas of package management and workflow execution.

Figure 20-1. Multi-Node Plug-In Schema

The plug-in contains a set of standard workflows for hierarchical orchestration, management of Orchestrator instances, and scale-out of Orchestrator activities.

Configuring the Multi-Node Plug-In

You must use the Orchestrator client to configure the Multi-Node plug-in.
**Servers Configuration Workflows**

The Servers Configuration workflow category contains workflows that allow you to configure the connected Orchestrator servers.

You can access these workflows from **Library > Orchestrator > Servers Configuration** on the **Workflows** view in the Orchestrator client.

<table>
<thead>
<tr>
<th>Workflow Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Add an Orchestrator server</td>
<td>Adds an Orchestrator server to the plug-in inventory.</td>
</tr>
<tr>
<td>Delete an Orchestrator server</td>
<td>Removes an Orchestrator server from the plug-in inventory and deletes all created proxies for this server.</td>
</tr>
<tr>
<td>Update an Orchestrator server</td>
<td>Updates an Orchestrator server from the plug-in inventory by changing its details.</td>
</tr>
</tbody>
</table>

### Add an Orchestrator Server

You can run a workflow to establish a connection to a new vRealize Orchestrator server.

**Prerequisites**

Verify that the master and slave Orchestrator servers are the same version.

**Procedure**

1. Log in to the Orchestrator client as an administrator.
2. Click the **Workflows** view in the Orchestrator client.
3. In the hierarchical list of workflows, expand **Library > Orchestrator > Servers Configuration** and navigate to the Add an Orchestrator server workflow.
4. Right-click the Add an Orchestrator server workflow and select **Start workflow**.
5. Provide the new server details.
6. Select whether the connection is shared.

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>No</td>
<td>The credentials of the logged-in user are used to connect to the remote Orchestrator server.</td>
</tr>
<tr>
<td>Yes</td>
<td>All users can access the remote Orchestrator server using the same credentials. Provide the credentials for the shared connection.</td>
</tr>
</tbody>
</table>

7. Click **Submit** to run the workflow.

### Using Proxy Workflows

You can use proxy workflows to manage the interaction between the local Orchestrator server and workflows on a remote Orchestrator server.
You can use the Multi-Node plug-in to generate local workflows which interact with remote workflows. These local workflows are called proxy workflows. A proxy workflow takes the input parameters from the inventory of the Multi-Node plug-in. When you run the proxy workflow, it converts the parameters to the types required by the remote workflow. When the remote workflow completes its run, the output parameters are converted back to the local representation on the master Orchestrator server.

**Synchronous Proxy Workflows**

The synchronous type of proxy workflows preserve the API and the operation contract of the remote workflows.

The schema of all synchronous proxy workflows is the same, but contains different scripting.

The synchronous proxy workflow completes the run after the remote workflow completes and provides output parameters.

The local workflow consumes no server resources while waiting for the results of the remote workflow.

At the end of a successful run the output parameters of the proxy workflow contain a local representation of the remote workflow token. The output parameters can be used directly by other workflows on the local Orchestrator server when they are of simple type, such as, boolean, number, string, and similar.

**Asynchronous Proxy Workflows**

You can use asynchronous proxy workflows to optimize the run of remote workflows.
The schema of all asynchronous proxy workflows is the same, but contains different scripting.

An asynchronous proxy workflow returns immediately a result that is a local wrapper of the remote workflow token object. The proxy workflow uses this token to check the state of the run and to retrieve the output parameters when the remote workflow completes its run. The output parameters can be used directly by other workflows on the local Orchestrator server when they are of simple type, such as, boolean, number, string, and similar.

Remote Execution Workflows

The Remote Execution workflow category contains workflows that allow you to manage proxy workflows.

Remote Execution Standard Workflows

You can access the workflows to create proxy workflows from Library > Orchestrator > Remote Execution on the Workflows view in the Orchestrator client.

<table>
<thead>
<tr>
<th>Workflow Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Create a multi proxy action</td>
<td>Creates a multi-proxy action to run workflows on multiple servers.</td>
</tr>
<tr>
<td>Create a proxy workflow</td>
<td>Creates a proxy workflow that you can use to start a workflow on a remote Orchestrator server.</td>
</tr>
<tr>
<td>Create proxy workflows from a folder</td>
<td>Creates proxy workflows for all workflows in a folder on the remote Orchestrator server.</td>
</tr>
</tbody>
</table>

Server Proxies

You can access the workflows for managing server proxies from Library > Orchestrator > Remote Execution > Server Proxies on the Workflows view in the Orchestrator client.
Create proxy workflows for an Orchestrator server

Creates proxy workflows on the local Orchestrator server by mirroring the remote server's structure.

Delete proxy workflows for an Orchestrator server

Removes the proxy workflows for the local Orchestrator server and deletes all generated workflows.

Refresh proxy workflows for an Orchestrator server

Regenerates all proxy workflows for the local Orchestrator server from the remote server.

Using the Multi-Node Plug-In Inventory

The Multi-Node plug-in mirrors all inventories of the connected vRealize Orchestrator servers in the Inventory view. You can use the Inventory view to add authorization elements or to run workflows on remote Orchestrator servers.

To display the workflows that are available for an inventory object, navigate to Tools > User preferences > Inventory and select the Use contextual menu in inventory check box. After the option is enabled, when you right-click an object in the Orchestrator inventory, all available workflows for the object are displayed.

The inventory for a single remote server consist of two major parts, system objects and plug-in objects. Both objects are wrappers of the remote objects into locally usable types:

- **System object**: System objects are under a top-level group called System. They contain configurations, packages, workflows, actions, and related folders. Remote system objects have individual wrapper types.

- **Plug-in objects**: Plug-in objects mirror the inventories of all plug-ins attached to the remote Orchestrator server. Remote plug-in objects are all wrapped into a single local type VCO:RemotePluginObject.

Remote Management Workflows

The Remote Management workflow category contains workflows that allow you to manage packages and workflows on remote Orchestrator instances.

Packages

You can access the workflows for managing remote packages from Library > Orchestrator > Remote Management > Packages on the Workflows view in the Orchestrator client.

<table>
<thead>
<tr>
<th>Workflow Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Delete a package</td>
<td>Deletes a package and its contents from a remote Orchestrator server.</td>
</tr>
<tr>
<td>Delete a package by name</td>
<td>Deletes a package and its contents by name on a remote Orchestrator server.</td>
</tr>
<tr>
<td>Deploy a package from a local server</td>
<td>Deploys a package from a local Orchestrator server to remote Orchestrator servers.</td>
</tr>
</tbody>
</table>
### Workflow Name

<table>
<thead>
<tr>
<th>Workflow Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Deploy a package from a remote server</td>
<td>Deploys a package from one remote Orchestrator server to a list of remote Orchestrator servers.</td>
</tr>
<tr>
<td>Deploy packages from a local server</td>
<td>Deploys packages from a local Orchestrator server to remote Orchestrator servers.</td>
</tr>
</tbody>
</table>

### Workflows

You can access the workflows for managing remote workflows from Library > Orchestrator > Remote Management > Workflows on the Workflows view in the Orchestrator client.

<table>
<thead>
<tr>
<th>Workflow Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Delete a remote workflow</td>
<td>Deletes a workflow from a remote Orchestrator server.</td>
</tr>
<tr>
<td>Delete all finished workflow runs</td>
<td>Deletes all finished workflow runs from a remote workflow.</td>
</tr>
<tr>
<td>Deploy a workflow from a local server</td>
<td>Deploys a workflow from a local Orchestrator server to a list of remote Orchestrator servers.</td>
</tr>
<tr>
<td>Deploy a workflow from a remote server</td>
<td>Deploys a workflow from a remote Orchestrator server to a list of other remote Orchestrator servers.</td>
</tr>
</tbody>
</table>

### Access the Multi-Node Plug-In API

Orchestrator provides an API Explorer to allow you to search the Multi-Node plug-in API and see the documentation for JavaScript objects that you can use in scripted elements.

#### Procedure

1. Log in to the Orchestrator client as an administrator.
2. Access the API Explorer from either the Orchestrator client or from the Scripting tabs of the workflow, policy, and action editors.
   - To access the API Explorer from the Orchestrator client, click Tools > API Explorer in the Orchestrator client toolbar.
   - To access the API Explorer from the Scripting tabs of the workflow, policy, and action editors, click Search API on the left.
3. To expand the hierarchical list of Multi-Node plug-in API objects, double-click the VCO module in the left pane.

#### What to do next

You can copy code from API elements and paste it into scripting boxes. For more information about API scripting, see Developing with VMware vRealize Orchestrator.

### Multi-Node Plug-In Use Cases

The Multi-Node plug-in use cases include user scenarios such as importing a package from the local Orchestrator server to the remote servers, using multi proxy actions, as well as information about maintenance of remote and proxy workflows.
Create a Multi-Proxy Action

You can run the Create a multi-proxy action workflow to run a workflow on several servers. You can create an action, so that you can run a workflow on a remote Orchestrator server at a later stage.

Procedure

1. Log in to the Orchestrator client as an administrator.
2. Click the Workflows view in the Orchestrator client.
3. In the hierarchical list of workflows, expand Library > Orchestrator > Remote Execution and navigate to the Create a multi-proxy action workflow.
4. Right-click the Create a multi-proxy action workflow and select Start workflow.
5. In the Action name text box, type the name of the action.
   The action name must contain only alpha-numeric characters without spaces.

A new action is created even if another action with the same name exists.

6. Select a module in which to add the action.
7. Select whether the workflow is local or remote.
8. Select the workflow that you want to use for this action.

9. Click Submit to run the workflow.

The generated action accepts the same parameters as the source workflow but promotes the parameters to an array in case of multi-selection of objects. The values in the array are indexed.

**Maintenance of Remote and Proxy Workflows**

If the remote and proxy workflows change, you might want to update the proxies or to delete them if you do not need them anymore. For maintenance purposes, the Multi-Node plug-in provides workflows that allow you to update or delete proxy and remote workflow information.
You can access the workflows for managing the proxy workflows from Library > Orchestrator > Remote Execution > Server Proxies in the Workflows view in the Orchestrator client.

<table>
<thead>
<tr>
<th>Workflow Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Refresh proxy workflows for an Orchestrator server</td>
<td>Regenerates all proxy workflows for the local Orchestrator server from the remote server.</td>
</tr>
<tr>
<td>Delete proxy workflows for an Orchestrator server</td>
<td>Removes the proxy workflows for the local Orchestrator server and deletes all generated workflows.</td>
</tr>
</tbody>
</table>

You can access workflows for further maintenance of the proxy workflows from Library > Orchestrator > Remote Management > Workflows in the Workflows view of the Orchestrator client.

<table>
<thead>
<tr>
<th>Workflow Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Delete all finished workflow runs</td>
<td>Deletes all finished workflow runs from a remote workflow.</td>
</tr>
<tr>
<td>Delete a remote workflow</td>
<td>Deletes a workflow from a remote Orchestrator server.</td>
</tr>
<tr>
<td>Deploy a workflow from a local server</td>
<td>Deploys a workflow from a local Orchestrator server to a list of remote Orchestrator servers.</td>
</tr>
</tbody>
</table>

**Deploy a Package from a Local Server**

You can run a workflow to deploy a package from a local Orchestrator server to remote Orchestrator servers.

In this example, you can deploy a package from a local server to an array of remote servers.

**Procedure**

1. Log in to the Orchestrator client as an administrator.
2. Click the Workflows view in the Orchestrator client.
3. In the hierarchical list of workflows, select Library > Orchestrator > Remote Management and navigate to the Deploy a package from a local server workflow.
4. Right-click the Deploy a package from a local server and select Start workflow.
5. Select the package to deploy from the local storage.
6. Select the remote servers to deploy the package to.
7. Select whether you want to overwrite the remote server packages.

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Yes</td>
<td>The packages on the remote server are replaced, discarding the version of the packaged elements.</td>
</tr>
<tr>
<td>No</td>
<td>A version check of the server and the deploying packages is performed. The packages are deployed after a successful check.</td>
</tr>
</tbody>
</table>

8. Click Submit to run the workflow.

After running the workflow, the status information is displayed in the log view and in the inventory of the plug-in.
Using the vCloud Suite API (vAPI) Plug-In

The vCloud Suite API plug-in provides the ability to consume API exposed by any vCloud Suite API provider. The vCloud Suite API provides a service-oriented architecture for accessing resources in the virtual environment by issuing requests to vCenter Server, through the vCloud Suite Endpoint.

The plug-in contains a set of standard workflows and example workflows. You can also create custom workflows that implement the plug-in to automate tasks in your virtual environment. For information about vCloud Suite API, see VMware vCloud Suite SDKs Programming Guide.

This chapter includes the following topics:
- Configuring the vCloud Suite API Plug-In
- Access the vCloud Suite API Plug-In API

Configuring the vCloud Suite API Plug-In

You can configure vCloud Suite API by running the configuration workflows included in the plug-in.

Import a vCloud Suite API Metamodel

The vCloud Suite API plug-in discovers vCloud Suite API services dynamically by querying a vCloud Suite API provider metadata service. vCloud Suite API providers which are not exposing metadata service are not supported.

You must import a vCloud Suite API metamodel and add endpoints afterwards.

Procedure

1. Log in to the Orchestrator client as an administrator.
2. Click the Workflows view in the Orchestrator client.
3. In the workflows hierarchical list, expand Library > VAPI and navigate to the Import vAPI metamodel workflow.
4. Right-click the Import vAPI metamodel workflow and select Start workflow.
5. In the vAPI endpoint URL text box, type the URL of your vCloud Suite API endpoint.
Choose whether to use a secure protocol connection:

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>No</td>
<td>Import the vCloud Suite API metamodel, without using a secure protocol connection.</td>
</tr>
</tbody>
</table>
| Yes    | To import the vCloud Suite API metamodel with secure protocol connection:  
|        | a Choose whether to ignore certificate warnings and accept the vCloud Suite endpoint automatically.  
|        | b Provide the user credentials to authenticate with the vCloud Suite endpoint. |

Click **Submit** to run the workflow.

**What to do next**

**Add a vCloud Suite API Endpoint**

Add a vCloud Suite API endpoint.

**Prerequisites**

Import a vCloud Suite API metamodel.

**Procedure**

1. Log in to the Orchestrator client as an administrator.
2. Click the **Workflows** view in the Orchestrator client.
3. In the workflows hierarchical list, expand **Library > VAPI** and navigate to the **Add vAPI endpoint** workflow.
4. Right-click the **Add vAPI endpoint** workflow and select **Start workflow**.
5. In the **vAPI endpoint URL** text box, type the URL of your vCloud Suite API endpoint.
6. Choose whether to use a secure protocol connection:

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>No</td>
<td>Import the vCloud Suite API metamodel, without using a secure protocol connection.</td>
</tr>
</tbody>
</table>
| Yes    | To import the vCloud Suite API metamodel with secure protocol connection:  
|        | a Choose whether to ignore certificate warnings and accept the vCloud Suite endpoint automatically.  
|        | b Provide the user credentials to authenticate with the vCloud Suite endpoint. |

7. Click **Submit** to run the workflow.
Access the vCloud Suite API Plug-In API

Orchestrator provides an API Explorer to allow you to search the vCloud Suite API plug-in API and see the documentation for JavaScript objects that you can use in scripted elements.

Procedure

1. Log in to the Orchestrator client as an administrator.

2. Access the API Explorer from either the Orchestrator client or from the Scripting tabs of the workflow, policy, and action editors.
   - To access the API Explorer from the Orchestrator client, click Tools > API Explorer in the Orchestrator client toolbar.
   - To access the API Explorer from the Scripting tabs of the workflow, policy, and action editors, click Search API on the left.

3. To expand the hierarchical list of vCloud Suite API plug-in API objects, double-click the VAPI module in the left pane.

What to do next

You can copy code from API elements and paste it into scripting boxes. For more information about API scripting, see Developing with VMware vRealize Orchestrator.