

Using VMware vRealize Orchestrator Plug-Ins

December 2022

vRealize Orchestrator 8.11

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

<https://docs.vmware.com/>

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

Copyright © 2008-2023 VMware, Inc. All rights reserved. [Copyright and trademark information.](#)

Contents

Using VMware vRealize Orchestrator Plug-Ins	10
1 Introduction to vRealize Orchestrator Plug-Ins	11
vRealize Orchestrator Architecture	12
Plug-Ins Installed with the vRealize Orchestrator Server	12
Access the vRealize Orchestrator API Explorer	16
Time Zone Codes	16
Configure Kerberos authentication for vRealize Orchestrator plug-ins	19
2 Configuring the vRealize Orchestrator Plug-Ins	22
Manage vRealize Orchestrator Plug-Ins	22
Install or Update a vRealize Orchestrator Plug-In	23
Delete a Plug-In	23
3 Using the Active Directory Plug-In	25
Configuring the Active Directory Plug-In	25
Using the Active Directory Plug-In Workflow Library	26
Computer Workflows	26
Organizational Unit Workflows	27
User Workflows	27
User Group Workflows	27
Client-Side Load Balancing for the Active Directory Plug-In	28
4 Using the AMQP Plug-In	30
Configuring the AMQP Plug-In	30
Add a Broker	30
Subscribe to Queues	31
Update a Broker	32
Using the AMQP Plug-In Workflow Library	32
Declare a Binding	33
Declare a Queue	33
Declare an Exchange	34
Send a Text Message	35
Delete a Binding	36
5 Using the Auto Deploy Plug-In	37
Configuring and the Auto Deploy Plug-In	37
Create and Apply a Rule	38

Troubleshooting	40
Auto Deploy Plug-In Workflow Library	40
6 Using the Configuration Plug-In	44
7 Using the Dynamic Types Plug-In	46
Dynamic Types Configuration Workflows	46
8 Using the F5 BIG-IP Plug-In	48
Run the Attach BIG-IP Workflow	49
F5 Plug-In Workflow Library	50
9 Using the HTTP-REST Plug-In	55
Persistent and Transient REST Hosts	55
Configuring the HTTP-REST Plug-In	58
Add a REST Host	58
Add a REST Operation	60
Add a Schema to a REST Host	61
Generate a New Workflow from a REST Operation	62
Invoke a REST Operation	62
Invoking a REST Operation	63
10 Using the Library Plug-In	64
11 Using the Mail Plug-In	66
Using the Mail Plug-In Sample Workflows	66
Define the Default SMTP Connection	67
12 Using the Multi-Node Plug-In	68
Introduction to the vRealize Orchestrator Multi-Node Plug-In	68
Configuring the Multi-Node Plug-In	69
Add an Orchestrator Server	69
Using Proxy Workflows	70
Synchronous Proxy Workflows	70
Asynchronous Proxy Workflows	71
Remote Execution Workflows	72
Using the Multi-Node Plug-In Inventory	73
Remote Management Workflows	73
Access the Multi-Node Plug-In API	74
Multi-Node Plug-In Use Cases	74
Create a Multi-Proxy Action	74

Maintenance of Remote and Proxy Workflows 75

Deploy a Package from a Local Server 76

13 Using the Net Plug-In 77

14 Using the PowerShell Plug-In 78

Introduction to the vRealize Orchestrator PowerShell Plug-In 78

PowerShell Plug-In Components 79

Configuring WinRM 79

Configure WinRM to Use HTTP 80

Configure WinRM to use HTTPS 82

Configuring the PowerShell Plug-In 83

Configuration Workflows 83

Add a PowerShell host 83

Access the PowerShell Plug-In API 85

Using the PowerShell Plug-In Inventory 85

Running PowerShell Scripts 85

Invoke a PowerShellScript 85

Invoke an External Script 86

Generating Actions 86

Generate an Action from a PowerShell Script 86

Generate an Action for a PowerShell Cmdlet 88

Passing Invocation Results Between Actions 89

PowerCLI Integration with the PowerShell Plug-In 89

Converter Workflows 89

Working with PowerShell Results 90

Sample Workflows 90

Examples of Scripts for Common PowerShell Tasks 91

Troubleshooting 93

Activate Kerberos Event Logging 93

Servers Not Found in Kerberos Database 94

Unable to Obtain a Kerberos Ticket 94

Kerberos Authentication Fails Due to Different Time Settings 94

Kerberos Authentication Session Mode Fails 95

Unable to Reach a Key Distribution Center for a Realm 95

Unable to Locate the Default Realm 96

15 Using the SNMP Plug-In 97

Managing SNMP Devices 97

Device Management Workflows 97

Register an SNMP Device 98

Managing SNMP Queries	99
Query Management Workflows	99
Add a Query to an SNMP Device	99
Managing the SNMP Trap Host	100
Trap Host Management Workflows	100
Add a SNMP trap port to the vRealize Orchestrator Appliance	100
Set the SNMP Trap Port	101
Receiving SNMP Traps	102
Wait for a Trap on an SNMP Device	102
Set an SNMP Trap Policy	102
Configure an SNMP Trap Host Policy	103
Edit a Trap Policy	104
Generic SNMP Request Workflows	104
16 Using the SOAP Plug-In	105
Configuring the SOAP Plug-In	105
Add a SOAP Host	106
Generate a New Workflow from a SOAP Operation	107
Test a Custom-Generated Workflow	108
Invoke a SOAP Operation	108
17 Using the SQL Plug-In	109
Configuring the SQL Plug-In	109
Add a Database	110
Add Tables to a Database	111
Update a Database	111
Running the SQL Sample Workflows	112
Generate a JDBC URL	112
Test a JDBC Connection	113
Create a Table by Using JDBC	113
Insert a Row into a JDBC Table	114
Select Rows from a JDBC Table	114
Delete an Entry from a JDBC Table	115
Delete All Entries from a JDBC Table	116
Drop a JDBC Table	116
Run a Complete JDBC Cycle	117
Running SQL Operations	117
Generate CRUD Workflows for a Table	118
Adding a JDBC Connector for the vRealize Orchestrator SQL Plug-In	119
18 Using the SSH Plug-In	121

Configuring the SSH Plug-In	121
Add an SSH Host	121
Running the SSH Plug-In Sample Workflows	122
Generate a Key Pair	123
Generate a Key Pair in vRealize Orchestrator	123
Use an Externally Generated Key Pair	124
Change the Key Pair Passphrase	125
Register a vRealize Orchestrator Public Key on an SSH Host	126
Run an SSH Command	126
Copy a File from an SSH Host	127
Copy a File to an SSH Host	128
19 Using the vCenter Plug-In	129
Configuring the vCenter Server Plug-In	129
Configure the Connection to a vCenter Instance	130
vCenter Plug-In Scripting API	131
Using the vCenter Plug-In Inventory	131
Performance Considerations for Querying	132
Using XPath Expressions with the vCenter Plug-In	132
Using XPath Expressions with the vCenter Plug-In Examples	133
vCenter Plug-In Workflow Library	134
Batch Workflows	137
Cluster and Compute Resource Workflows	137
Configuration Workflows	138
Custom Attributes Workflows	138
Datacenter Workflows	138
Datastore and Files Workflows	139
Datacenter Folder Management Workflows	139
Host Folder Management Workflows	140
Virtual Machine Folder Management Workflows	140
Guest Operation Files Workflows	140
Guest Operation Processes Workflows	141
Power Host Management Workflows	141
Basic Host Management Workflows	141
Host Registration Management Workflows	142
Networking Workflows	142
Distributed Virtual Port Group Workflows	143
Distributed Virtual Switch Workflows	143
Standard Virtual Switch Workflows	143
Networking Virtual SAN Workflows	144
Resource Pool Workflows	144

Storage Workflows	144
Storage DRS Workflows	145
Storage VSAN Workflows	146
Basic Virtual Machine Management Workflows	146
Clone Workflows	147
Linked Clone Workflows	147
Linux Customization Clone Workflows	148
Tools Clone Workflows	148
Windows Customization Clone Workflows	149
Device Management Workflows	150
Move and Migrate Workflows	150
Other Workflows	151
Power Management Workflows	151
Snapshot Workflows	152
VMware Tools Workflows	152

20 Using the vCOIN Plug-In 154

Configuring the vRealize Orchestrator Plug-in for vSphere Web Client	155
Unregister the vCOIN Plug-In	157
Create a Context Action	157

21 Using the vCloud Suite API (vAPI) Plug-In 159

Configuring the vCloud Suite API Plug-In	159
Import a vCloud Suite API Metamodel	159
Add a vCloud Suite API Endpoint	160
Access the vCloud Suite API Plug-In API	161

22 Using the vRealize Automation Plug-In 162

Host Configuration Workflows	164
Add a vRealize Automation Host	166
Add a vRealize Automation Cloud Host	166
Infrastructure Workflows	167
Add a vSphere Cloud Account	171
Add a Cloud Zone	172
Add a Disk	173
Add a Machine	174
Add a Network	175
Add a Project	176

23 Using the VUM Plug-In 178

Connect the VUM Plug-In to vCenter	179
------------------------------------	-----

VUM Plug-In Workflow Library 180

24 Using the XML Plug-In 184

Running the XML Plug-In Sample Workflows 184

Create a Simple XML Document 185

Find an Element in an XML Document 185

Modify an XML Document 186

Create an Example Address Book from XML 187

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 data center operations.

Introduction to vRealize Orchestrator Plug-Ins

1

With the vRealize Orchestrator plug-ins, you can access and control external technologies and applications. Exposing an external technology in a vRealize 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.

vRealize 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 vRealize Orchestrator open plug-in architecture lets you develop plug-ins to access other applications. vRealize Orchestrator implements open standards to simplify integration with external systems.

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

Plug-ins extend the vRealize Orchestrator scripting engine with new object types and methods, and plug-ins publish notification events from the external system that triggers events in vRealize Orchestrator and in the plugged-in technology. Plug-ins provide an inventory of JavaScript objects that you can access on the **Inventory** page of the vRealize 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.

Read the following topics next:

- [vRealize Orchestrator Architecture](#)
- [Plug-Ins Installed with the vRealize Orchestrator Server](#)
- [Access the vRealize Orchestrator API Explorer](#)
- [Time Zone Codes](#)
- [Configure Kerberos authentication for vRealize Orchestrator plug-ins](#)

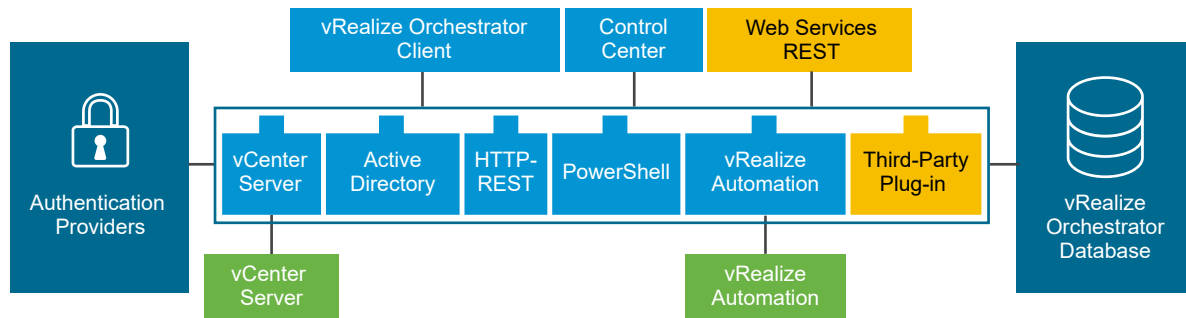
vRealize Orchestrator Architecture

vRealize 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 vRealize Orchestrator accesses through a series of plug-ins.

vRealize Orchestrator provides a standard set of plug-ins, including plug-ins for vCenter and vRealize Automation, to allow you to orchestrate tasks in the different environments that the plug-ins expose.

vRealize Orchestrator also presents an open architecture for plugging 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. vRealize Orchestrator connects to an authentication provider to manage user accounts and to a preconfigured PostgreSQL database to store information from the workflows that it runs. You can access vRealize Orchestrator, the objects it exposes, and the vRealize Orchestrator workflows through the vRealize Orchestrator Client, or through Web services. Monitoring and configuration of vRealize Orchestrator workflows and services is done through the vRealize Orchestrator Client and Control Center.

Figure 1-1. VMware vRealize Orchestrator Architecture



Plug-Ins Installed with the vRealize Orchestrator Server

vRealize Orchestrator includes a collection of standard plug-ins. Each plug-in exposes an external product API to the vRealize 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 installed plug-ins on the **Manage Plug-Ins** page in Control Center.

Table 1-1. Plug-Ins Installed with vRealize Orchestrator

Plug-In	Purpose	Configuration
Active Directory	Provides interaction between vRealize Orchestrator and Microsoft Active Directory.	See Configuring the Active Directory Plug-In .
AMQP	Interact with Advanced Message Queuing Protocol (AMQP) servers, also known as brokers.	See Configuring the AMQP Plug-In .

Table 1-1. Plug-Ins Installed with vRealize Orchestrator (continued)

Plug-In	Purpose	Configuration
Auto Deploy	Perform different operations associated with vCenter server ESXi hosts.	See Configuring and the Auto Deploy Plug-In .
Configuration	Provides workflows for configuring and managing the vRealize Orchestrator server keystores and trusted certificates.	None
Dynamic Types	Define dynamic types and create and use objects of these dynamic types.	See Chapter 7 Using the Dynamic Types Plug-In .
Enumeration	Provides common Enumerated Types that can be used in workflows by other plug-ins.	See Time Zone Codes
HTTP-REST	Manage REST Web services through an interaction between vRealize Orchestrator and REST hosts.	See Configuring the HTTP-REST Plug-In .
Library	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.	None
Mail	Uses Simple Mail Transfer Protocol (SMTP) to send email from workflows.	Set the default values for the <code>EmailMessage</code> object to use. See Define the Default SMTP Connection .
Multi-Node	Contains workflows for hierarchical management, management of vRealize Orchestrator instances, and scale-out of Orchestrator activities.	See Chapter 12 Using the Multi-Node Plug-In .
Net	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. With the Mail plug-in, the Net plug-in provides complete email sending and receiving capabilities in workflows.	
PowerShell	Manage PowerShell hosts and run custom PowerShell operations.	See Chapter 14 Using the PowerShell Plug-In .
SNMP	Connect and receive information from SNMP-enabled systems and devices.	
SOAP	Manage SOAP Web services by providing interaction between vRealize Orchestrator and SOAP hosts.	See Configuring the SOAP Plug-In .
SQL	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.	

Table 1-1. Plug-Ins Installed with vRealize Orchestrator (continued)

Plug-In	Purpose	Configuration
SSH	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 vRealize Orchestrator Client inventory.	See Add an SSH Host .
vCenter	Provides access to the vCenter API so that you can incorporate all the vCenter objects and functions into the management processes that you automate by using vRealize Orchestrator.	See Configuring the vCenter Server Plug-In .
vCloud Suite API (vAPI)	Provides access to the API services exposed by any vAPI provider.	
vRealize Automation	The plug-in comes pre-installed with vRealize Orchestrator deployments that are embedded in vRealize Automation. Integrates vRealize Orchestrator with vRealize Automation and vRealize Automation Cloud.	See Chapter 22 Using the vRealize Orchestrator Plug-In for vRealize Automation .
vSphere Update Manager (VUM)	Provides interaction between vRealize Orchestrator and VMware vSphere Update Manager/vSphere Lifecycle Manager.	See Connect the vSphere Update Manager Plug-In to vCenter .
XML	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 vRealize Orchestrator JavaScript API.	

Plug-In Components

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

Plug-In Name in the Configuration UI	Workflow Categories	API Module
Active Directory	Computer Configuration Organizational Unit User User Group	AD
AMQP	Configuration	AMQP

Table 1-2. Names of Plug-In Components (continued)

Plug-In Name in the Configuration UI	Workflow Categories	API Module
Auto Deploy	Answer Files Configuration Reprovision Host Rules Rule Set Compliance	AutoDeploy
Configuration	Configuration	Configurator
Dynamic Types	Configuration	DynamicTypes
Common enumerated types	None	Enums
HTTP-REST	Configuration	REST
Library	Locking Orchestrator Tagging	Not applicable.
Mail	Mail	Mail
Orchestrator Multi-Node	Servers Configuration Remote Execution Remote Management Tasks Workflows	VCO
Net	None	Net
PowerShell	Configuration Generate Templates	PowerShell
SNMP	Device Management Query Management Trap Host Management	SNMP
SOAP	Configuration	SOAP
SQL	JDBC SQL	SQL
SSH	SSH	SSH
Support	None	Support
vAPI	VAPI	VAPI
vCenter	vCenter	VC
vRealize Automation	Configuration Infrastructure	VRA

Table 1-2. Names of Plug-In Components (continued)

Plug-In Name in the Configuration UI	Workflow Categories	API Module
vSphere Update Manager	Configuration	VUM
XML	XML	XML

Access the vRealize Orchestrator API Explorer

You can use the vRealize Orchestrator API Explorer as an in-product reference guide to JavaScript objects exposed by the vRealize Orchestrator and all installed plug-ins.

You can consult an online version of the Scripting API for the vRealize Orchestrator plug-ins on the vRealize Orchestrator documentation home page.

Procedure

- 1 Log in to the vRealize Orchestrator Client.
- 2 Navigate to **API Explorer**.

Results

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

What to do next

Use the vRealize Orchestrator API Explorer as a reference guide when you write scripts for scriptable items or actions.

Time Zone Codes

When implementing common enumerated types in workflows, you can use time zone codes as possible values for the `Enums:MSTimeZone` enumeration.

Time Zone Code	Time Zone Name	Description
000	Dateline Standard Time	(GMT-12:00) International Date Line West
001	Samoa Standard Time	(GMT-11:00) Midway Island, Samoa
002	Hawaiian Standard Time	(GMT-10:00) Hawaii
003	Alaskan Standard Time	(GMT-09:00) Alaska
004	Pacific Standard Time	(GMT-08:00) Pacific Time (US and Canada); Tijuana
010	Mountain Standard Time	(GMT-07:00) Mountain Time (US and Canada)
013	Mexico Standard Time 2	(GMT-07:00) Chihuahua, La Paz, Mazatlan

Time Zone Code	Time Zone Name	Description
015	U.S. Mountain Standard Time	(GMT-07:00) Arizona
020	Central Standard Time	(GMT-06:00) Central Time (US and Canada)
025	Canada Central Standard Time	(GMT-06:00) Saskatchewan
030	Mexico Standard Time	(GMT-06:00) Guadalajara, Mexico City, Monterrey
033	Central America Standard Time	(GMT-06:00) Central America
035	Eastern Standard Time	(GMT-05:00) Eastern Time (US and Canada)
040	U.S. Eastern Standard Time	(GMT-05:00) Indiana (East)
045	S.A. Pacific Standard Time	(GMT-05:00) Bogota, Lima, Quito
050	Atlantic Standard Time	(GMT-04:00) Atlantic Time (Canada)
055	S.A. Western Standard Time	(GMT-04:00) Caracas, La Paz
056	Pacific S.A. Standard Time	(GMT-04:00) Santiago
060	Newfoundland and Labrador Standard Time	(GMT-03:30) Newfoundland and Labrador
065	E. South America Standard Time	(GMT-03:00) Brasilia
070	S.A. Eastern Standard Time	(GMT-03:00) Buenos Aires, Georgetown
073	Greenland Standard Time	(GMT-03:00) Greenland
075	Mid-Atlantic Standard Time	(GMT-02:00) Mid-Atlantic
080	Azores Standard Time	(GMT-01:00) Azores
083	Cape Verde Standard Time	(GMT-01:00) Cape Verde Islands
085	GMT Standard Time	(GMT) Greenwich Mean Time: Dublin, Edinburgh, Lisbon, London
090	Greenwich Standard Time	(GMT) Casablanca, Monrovia
095	Central Europe Standard Time	(GMT+01:00) Belgrade, Bratislava, Budapest, Ljubljana, Prague
100	Central European Standard Time	(GMT+01:00) Sarajevo, Skopje, Warsaw, Zagreb
105	Romance Standard Time	(GMT+01:00) Brussels, Copenhagen, Madrid, Paris
110	W. Europe Standard Time	(GMT+01:00) Amsterdam, Berlin, Bern, Rome, Stockholm, Vienna
113	W. Central Africa Standard Time	(GMT+01:00) West Central Africa
115	E. Europe Standard Time	(GMT+02:00) Bucharest
120	Egypt Standard Time	(GMT+02:00) Cairo

Time Zone Code	Time Zone Name	Description
125	FLE Standard Time	(GMT+02:00) Helsinki, Kyiv, Riga, Sofia, Tallinn, Vilnius
130	GTB Standard Time	(GMT+02:00) Athens, Istanbul, Minsk
135	Israel Standard Time	(GMT+02:00) Jerusalem
140	South Africa Standard Time	(GMT+02:00) Harare, Pretoria
145	Russian Standard Time	(GMT+03:00) Moscow, St. Petersburg, Volgograd
150	Arab Standard Time	(GMT+03:00) Kuwait, Riyadh
155	E. Africa Standard Time	(GMT+03:00) Nairobi
158	Arabic Standard Time	(GMT+03:00) Baghdad
160	Iran Standard Time	(GMT+03:30) Tehran
165	Arabian Standard Time	(GMT+04:00) Abu Dhabi, Muscat
170	Caucasus Standard Time	(GMT+04:00) Baku, Tbilisi, Yerevan
175	Transitional Islamic State of Afghanistan Standard Time	(GMT+04:30) Kabul
180	Ekaterinburg Standard Time	(GMT+05:00) Ekaterinburg
185	West Asia Standard Time	(GMT+05:00) Islamabad, Karachi, Tashkent
190	India Standard Time	(GMT+05:30) Chennai, Kolkata, Mumbai, New Delhi
193	Nepal Standard Time	(GMT+05:45) Kathmandu
195	Central Asia Standard Time	(GMT+06:00) Astana, Dhaka
200	Sri Lanka Standard Time	(GMT+06:00) Sri Jayawardenepura
201	N. Central Asia Standard Time	(GMT+06:00) Almaty, Novosibirsk
203	Myanmar Standard Time	(GMT+06:30) Yangon (Rangoon)
205	S.E. Asia Standard Time	(GMT+07:00) Bangkok, Hanoi, Jakarta
207	North Asia Standard Time	(GMT+07:00) Krasnoyarsk
210	China Standard Time	(GMT+08:00) Beijing, Chongqing, Hong Kong SAR, Urumqi
215	Singapore Standard Time	(GMT+08:00) Kuala Lumpur, Singapore
220	Taipei Standard Time	(GMT+08:00) Taipei
225	W. Australia Standard Time	(GMT+08:00) Perth
227	North Asia East Standard Time	(GMT+08:00) Irkutsk, Ulaan Bataar

Time Zone Code	Time Zone Name	Description
230	Korea Standard Time	(GMT+09:00) Seoul
235	Tokyo Standard Time	(GMT+09:00) Osaka, Sapporo, Tokyo
240	Yakutsk Standard Time	(GMT+09:00) Yakutsk
245	A.U.S. Central Standard Time	(GMT+09:30) Darwin
250	Cen. Australia Standard Time	(GMT+09:30) Adelaide
255	A.U.S. Eastern Standard Time	(GMT+10:00) Canberra, Melbourne, Sydney
260	E. Australia Standard Time	(GMT+10:00) Brisbane
265	Tasmania Standard Time	(GMT+10:00) Hobart
270	Vladivostok Standard Time	(GMT+10:00) Vladivostok
275	West Pacific Standard Time	(GMT+10:00) Guam, Port Moresby
280	Central Pacific Standard Time	(GMT+11:00) Magadan, Solomon Islands, New Caledonia
285	Fiji Islands Standard Time	(GMT+12:00) Fiji Islands, Kamchatka, Marshall Islands
290	New Zealand Standard Time	(GMT+12:00) Auckland, Wellington
300	Tonga Standard Time	(GMT+13:00) Nuku'alofa

Configure Kerberos authentication for vRealize Orchestrator plug-ins

You can use Kerberos authentication for vRealize Orchestrator plug-ins.

Configure the `krb5.conf` file

- 1 Create or edit the `krb5.conf` file at `/data/vco/usr/lib/vco/app-server/conf/`.

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

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

The `krb5.conf` must contain specific configuration parameters with their values.

Kerberos configuration tags	Details
<code>default_realm</code>	The default Kerberos realm that a client uses to authenticate against an Active Directory server. Must be in uppercase letters.
<code>kdc</code>	The domain controller that acts as a Key Distribution Center (KDC) and issues Kerberos tickets.
<code>default_domain</code>	The default domain that is used to produce a fully qualified domain name. This tag is used for Kerberos 4 compatibility.

To allow ticket forwarding to other external systems, add the `forwardable = true` flag. For additional information, see [the Oracle documentation on the `krb5.conf` file](#).

By default, the Java Kerberos configuration uses the UDP protocol. To use only the TCP protocol, you must specify the `udp_preference_limit` parameter with a value `1`.

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 vRealize Orchestrator server service.

If you have a clustered vRealize Orchestrator environment, make sure that the `krb5.conf` file exists in all three appliances with the same configuration before you restart the vRealize Orchestrator pods.

2 Change permissions.

```
chmod 644 krb5.conf
```

3 Redeploy the vRealize Orchestrator pod.

```
kubectl -n prelude get pods
```

Look for an entry similar to `vco-app-<ID>`.

4 Destroy the pod.

```
kubectl -n prelude delete pod vco-app-<ID>
```

A new pod is automatically deployed to replace the pod you destroyed.

Enable Kerberos debug logging

You can troubleshoot vRealize Orchestrator plug-in problems by modifying the Kerberos configuration file used by the plug-in.

The Kerberos configuration file is located in the `/data/vco/usr/lib/vco/app-server/conf/` directory of the vRealize Orchestrator Appliance.

- 1 Log in to the vRealize Orchestrator Appliance command line as **root**.
- 2 Run the `kubect1 -n prelude edit deployment vco-app` command.
- 3 In the deployment file, locate and edit the `-Djava.security.krb5.conf=/usr/lib/vco/app-server/conf/krb5.conf` string.

```
-Djava.security.krb5.conf=/usr/lib/vco/app-server/conf/krb5.conf  
-Dsun.security.krb5.debug=true
```

- 4 Save the changes and exit the file editor.
- 5 Run the `kubect1 -n prelude get pods` command. Wait until all pods are running.
- 6 To monitor the Kerberos login, run the following command.

```
tail -f /services-logs/prelude/vco-app/console-logs/vco-server-app.log
```

- 7 Alternatively, you can enable debug logging in the vRealize Orchestrator configurator by adding the `sun.security.krb5.debug = true` system property.

Configuring the vRealize Orchestrator Plug-Ins

2

The vRealize Orchestrator Appliance provides access to a preinstalled library of default plug-ins. The default vRealize Orchestrator plug-ins are configured with plug-in specific workflows run in the vRealize Orchestrator Client.

The default vRealize Orchestrator plug-ins come with configuration workflows. You can run these workflows from the vRealize Orchestrator Client to register endpoints for management.

The configuration workflows have the *configuration* tag. For example, to access workflows that are used to manage AMQP brokers and subscriptions, enter the tags *AMQP* and *Configuration* in the search text box of the workflow library.

Read the following topics next:

- [Manage vRealize Orchestrator Plug-Ins](#)
- [Install or Update a vRealize Orchestrator Plug-In](#)
- [Delete a Plug-In](#)

Manage vRealize Orchestrator Plug-Ins

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

Install or Upgrade a Plug-In

With the vRealize Orchestrator plug-ins, the vRealize Orchestrator server can integrate with other software products. vRealize Orchestrator comes with a set of preinstalled default plug-ins. You can further expand the capabilities of the vRealize Orchestrator platform by installing custom plug-ins.

You can install or upgrade plug-ins from the **Manage Plug-Ins** page of the vRealize Orchestrator. The file extension that can be used is `.vmoapp`.

For more information on installing or upgrading vRealize Orchestrator plug-ins, see [Install or Update a vRealize Orchestrator Plug-In](#).

Change Plug-In Logging Level

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

Deactivate a Plug-In

You can deactivate a plug-in by deselecting the **Enable plug-in** option 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 vRealize Orchestrator, see [Delete a Plug-In](#).

Install or Update a vRealize Orchestrator Plug-In

You can install or update third-party plug-ins in the vRealize Orchestrator Control Center.

Prerequisites

Download the *.dar* or *.vmoapp* file of the plug-in.

Note The preferred file format for vRealize Orchestrator plug-ins is *.vmoapp*.

Procedure

- 1 Log in the Control Center as **root**.
- 2 Select the **Manage Plug-ins** page.
- 3 Click **Browse** and select the *.dar* or *.vmoapp* file of the plug-in you want to install or update.
- 4 Click **Upload**.
- 5 Review the plug-in information, if applicable, accept the end-user license agreement, and click **Install**.

The plug-in is installed or updated and the vRealize Orchestrator server service is restarted.

What to do next


Verify that the correct plug-in information is listed on the **Manage Plug-ins** page.

Delete a Plug-In

You can delete third-party plug-ins from the vRealize Orchestrator Appliance through Control Center.

Procedure

- 1 Log in to the Control Center as **root**.
- 2 Select **Manage Plug-ins**.

- 3 Find the plug-in you want to delete and click the delete icon ().
- 4 Confirm that you want to delete the plug-in, and click **Delete**.

Results

You deleted the plug-in from the vRealize Orchestrator Appliance. You do not have to delete the plug-in package manually from the vRealize Orchestrator Client.

Using the Active Directory Plug-In

3

The 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 vRealize Orchestrator workflows that automate Active Directory processes.

The Active Directory 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.

Read the following topics next:

- [Configuring the Active Directory Plug-In](#)
- [Using the Active Directory Plug-In Workflow Library](#)
- [Client-Side Load Balancing for the Active Directory Plug-In](#)

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.

To access these workflows in the vRealize Orchestrator Client, navigate to **Library > Workflows** and enter the **active_directory** and **configuration** tags in the workflow search box.

Workflow Name	Description
Add an Active Directory server	Adds a new Active Directory domain configuration.
Configure Active Directory plug-in options	Configures the search limitation options of the Active Directory plug-in.
Update an Active Directory server	Modifies an existing Active Directory server configuration.
Remove an Active Directory server	Removes an Active Directory server configuration.
Configure Active Directory server (Deprecated)	Creates or updates the default Active Directory server configuration. Use Update an Active Directory server.
Reset configuration (Deprecated)	Deletes the default Active Directory server configuration. Use Remove an Active Directory server.

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.

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.

To display the workflows that are available for an Active Directory inventory object, navigate to **Administration > Inventory > Active Directory** in the vRealize Orchestrator Client.

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.

Active Directory Plug-In Computer Workflows

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

To access these workflows, navigate to **Library > Workflows** and enter the **active_directory** and **computer** tags in the workflow search box.

Workflow Name	Description
Create a computer in a group	Creates an Active Directory computer in a group.
Create a computer in an organizational unit	Creates an Active Directory computer in an organizational unit.
Create a computer with password in an organizational unit	Creates an Active Directory computer in an organizational unit with a default password. The default password is <i>computername\$</i> . When you run the workflow, you can change the default password. To set a custom password, select the Choose password different from the default one check box, and enter a new password.
Destroy a computer	Deletes a computer from an Active Directory instance.
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	Deactivates a computer from an Active Directory instance.
Enable a computer	Activates a computer in an Active Directory instance.

Active Directory Plug-In Organizational Unit Workflows

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

To access these workflows, navigate to **Library > Workflows** and enter the **active_directory** and **organizational_unit** tags in the workflow search box.

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.

Active Directory Plug-In User Workflows

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

To access these workflows, navigate to **Library > Workflows** and enter the **active_directory** and **user** tags in the workflow search box.

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 a password. The password must be created 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 deactivated, you cannot create a password for the user. 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 deactivated, you cannot specify a password.
Destroy a user	Deletes a user from an Active Directory instance.
Disable a user	Deactivates a user from an Active Directory instance.
Enable a user	Activates a user in an Active Directory instance.
Remove a user from a user group	Removes a user from a user group.

Active Directory Plug-In User Group Workflows

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

To access these workflows, navigate to **Library > Workflows** and enter the **active_directory** and **user_group** tags in the workflow search box.

Workflow Name	Description
Add computers to group members	Adds one or more computers as members of a user group.
Add groups to group members	Adds one or more user groups as members of a user group.
Add users to group members	Adds one or more users as members of a user group.
Create a user group in a group	Creates a user group in an existing container (group).
Create a user group in a group and set attribute "Group name (pre-Windows 2000)"	Creates a user group in an existing container (organizational unit) and sets the <code>Group name (pre-Windows 2000)</code> attribute.
Create a user group in an organizational unit	Creates a user group in an existing container (organizational unit).
Destroy a user group	Deletes a user group from an Active Directory instance.
Remove computers from group members	Removes one or more computers from a user group.
Remove groups from group members	Removes one or more user groups from a user group.
Remove users from group members	Removes one or more users from a user group.

Client-Side Load Balancing for the Active Directory Plug-In

You can use client-side load balancing and failover to improve the stability of your Active Directory plug-in configuration.

You can configure client-side load balancing when running the **Add an Active Directory server** and **Update an Active Directory server** workflows. Client-side load balancing is possible through the ServerSet Java class.

Procedure

- 1 Log in to the vRealize Orchestrator Client as an administrator.
- 2 Navigate to **Library > Workflows** and enter the **active_directory** and **configuration** tags in the workflow search box.
- 3 Run the **Add an Active Directory server** or **Update an Active Directory server** workflow.
- 4 Select the **Alternative hosts** tab.
- 5 From the drop-down menu, select **Single Server**, **Round-Robin DNS Server**, **Round-Robin**, or **Failover**.

Option	Description
Single Server	A server set implementation that connects to only one server.
Round-Robin DNS Server	A server set where server handles the case in which a given host name may resolve to multiple IP addresses. This server set does strictly require DNS server setup. The ordering mechanism for selecting an address is round-robin.

Option	Description
Round-Robin	A server set where load is distributed evenly between several directory servers. If a server is unavailable, the connection will move to the next server in the set.
Failover	A server set where server connections are established in order. This implementation can establish connections between separate server sets. Useful for providing high availability in complex environments.

- When you finish configuring the workflow run, click **Run**.

Using the AMQP Plug-In

4

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.

Read the following topics next:

- [Configuring the AMQP Plug-In](#)
- [Using the AMQP Plug-In Workflow Library](#)

Configuring the AMQP Plug-In

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

You can configure AMQP by running the configuration workflows included in the plug-in. The Configuration workflow category contains workflows that allow you to manage AMQP brokers.

To access these workflows in the vRealize Orchestrator Client, navigate to **Library > Workflows** and enter the **amqp** and **configuration** tags in the workflow search box.

Workflow Name	Description
Add a broker	Adds an AMQP broker.
Remove a broker	Removes an AMQP broker.
Remove a subscription	Removes an AMQP message subscription.
Subscribe to queues	Creates a subscription element.
Update a broker	Updates broker properties.
Validate a broker	Validate a broker by attempting to start a connection.

Add a Broker

You can run a workflow to add an AMQP broker.

Procedure

- 1 Log in to the vRealize Orchestrator Client as an administrator.
- 2 Navigate to **Library > Workflows** and enter the **amqp** and **configuration** tags in the workflow search box.
- 3 Locate the **Add a broker** workflow and click **Run**.
- 4 On the **AMQP broker properties** tab, enter the name of the broker.
- 5 On the **AMQP connection properties** tab, provide the information required for the broker connection.

Option	Action
Host	Enter the address of the host.
Port	Enter the port of the AMQP broker service. The default port is 5672.
Virtual host	Enter the address of the virtual host. The default value provided is /.
Use SSL	Select whether to use SSL certificates.
Accept all certificates	Select whether to accept all SSL certificates without validation.
User name	Enter the user name for the broker.
Password	Enter the password for the broker.

- 6 Click **Run**.

Results

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 properties of the broker before validating again.

Subscribe to Queues

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

Prerequisites

- Verify that you are logged in to the vRealize 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 Navigate to **Library > Workflows** and enter the **amqp** and **configuration** tags in the workflow search box.
- 2 Locate the **Subscribe to queues** workflow and click **Run**.

- 3 On the **Subscription** tab, enter the name of the queue to display.
- 4 On the **AMQP Broker** tab, select the broker to which you want to add the subscription.
- 5 On the **Queues** tab, select all the queues for message subscription.
- 6 Click **Run**.

Results

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 vRealize Orchestrator Client as an administrator.
- Verify that you have a connection to an AMQP broker from the **Inventory** view.

Procedure

- 1 Navigate to **Library > Workflows** and enter the **amqp** and **configuration** tags in the workflow search box.
- 2 Locate the **Update a broker** workflow and click **Run**.
- 3 On the **AMQP Broker** tab, select the broker that you want to update.
Current properties of the broker appear on the **New AMQP connection properties** tab.
- 4 On the **New AMQP connection properties** tab, edit the properties that you want.
- 5 Click **Run**.

Using the AMQP Plug-In Workflow Library

The AMQP workflow category contains workflows that allow you to run AMQP operations.

To access these workflows in the vRealize Orchestrator Client, navigate to **Library > Workflows** and enter the **amqp** tag in the workflow search box.

Workflow Name	Description
Bind	Creates a binding in a specified broker.
Declare a queue	Adds a queue to a specified broker.
Declare an exchange	Adds an exchange to a specified broker.
Delete a queue	Deletes a queue from a specified broker.

Workflow Name	Description
Delete an exchange	Deletes an exchange from a specified broker.
Receive a text message	Receives a text message from a specified broker.
Send a test message	Sends a text message using a specified broker.
Unbind	Unbinds binding in a specified broker.

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 vRealize Orchestrator Client as an administrator.
- Verify that you have a connection to an AMQP broker from the **Inventory** view.

Procedure

- 1 Navigate to **Library > Workflows** and enter the **amqp** tag in the workflow search box.
- 2 Locate the **Bind** workflow and click **Run**.
- 3 On the **AMQP Broker** tab, select a broker in which you want to create a binding.
- 4 On the **Binding Properties** tab, provide information about the binding.

Option	Action
Queue name	Enter the name of the queue.
Exchange name	Enter the name of the exchange.
Routing key	Enter the routing key.

- 5 Click **Run**.

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 vRealize Orchestrator Client as an administrator.
- Verify that you have a connection to an AMQP broker from the **Inventory** view.

Procedure

- 1 Navigate to **Library > Workflows** and enter the **amqp** tag in the workflow search box.
- 2 Locate the **Declare a queue** workflow and click **Run**.
- 3 On the **AMQP Broker** tab, select a broker to which you want to add the queue.

4 On the **Queue Properties** tab, define the queue properties.

- a In the **Name** text box, enter the name of the queue to display.
- b Select whether the queue is durable.

Option	Description
Yes	The queue is removed after a broker restart.
No	The queue remains after a broker restart.

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

Option	Description
Yes	Sets one client for this specific queue.
No	Sets more clients for this specific queue.

- d Select whether to delete the queue with activated subscription automatically.

Option	Description
Yes	Automatically deletes the queue when no more clients are connected to it. The queue remains until at least one client subscribes to it.
No	Does not delete the queue.

5 Click **Run**.

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 vRealize Orchestrator Client as an administrator.
- Verify that you have a connection to an AMQP broker from the **Inventory** view.

Procedure

- 1 Navigate to **Library > Workflows** and enter the **amqp** tag in the workflow search box.
- 2 Locate the **Declare an exchange** workflow and click **Run**.
- 3 On the **AMQP Broker** tab, select a broker to which you want to add the exchange.

4 On the **Exchange Properties** tab, define the exchange properties.

- a In the **Name** text box, enter the name of the queue to display.
- b Select the exchange type.

Option	Description
direct	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.
fanout	Forwards any message sent to this exchange to all queues bound to it. Queues that are bound to this exchange contain no arguments.
headers	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.
topic	Performs a wildcard match between the routing key and the routing pattern specified in the binding.

- c Select whether the exchange is durable.

Option	Description
Yes	The exchange remains after a broker restart.
No	The exchange is removed after a broker restart.

- d Select whether to delete the exchange with activated subscription automatically.

Option	Description
Yes	Automatically deletes the exchange when no more queues are bound to it. The exchange remains until at least one queue is bound to it.
No	Does not delete the exchange.

5 Click **Run**.

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 vRealize Orchestrator Client as an administrator.
- Verify that you have a connection to an AMQP broker from the **Inventory** view.

Procedure

- 1 Navigate to **Library > Workflows** and enter the **amqp** tag in the workflow search box.
- 2 Locate the **Send a text message** workflow and click **Run**.
- 3 On the **AMQP Broker** tab, select a broker from which you want to send a message.

- 4 On the **Exchange** tab, specify the name of the exchange and the routing key.
- 5 On the **Message** tab, enter the message you want to send.
- 6 Click **Run**.

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 vRealize Orchestrator Client as an administrator.
- Verify that you have a connection to an AMQP broker from the **Inventory** view.

Procedure

- 1 Navigate to **Library > Workflows** and enter the **amqp** tag in the workflow search box.
- 2 Locate the **Unbind** workflow and click **Run**.
- 3 On the **AMQP Broker** tab, select a broker to remove the binding from.
- 4 On the **Binding Properties** tab, enter the name of the queue, the name of the exchange, and the routing key.
- 5 Click **Run**.

Using the Auto Deploy Plug-In

5

You can use the preinstalled Auto Deploy plug-in for vRealize Orchestrator to manage the ESXi hosts on your vCenter Server.

You can use the Auto Deploy plug-in for vRealize Orchestrator to perform different operations associated with your vCenter ESXi hosts, such as applying image profiles and host profiles or changing the location where the host is stored.

An Image profile includes the vSphere Installation Bundles (VIBs) that define the installation and upgrade process for your ESXi host. Image profiles are managed through the vSphere Auto Deploy service and defined through the vSphere Image Builder service. For more information on image profiles, see [Image Profiles](#).

Image profiles and their associated VIBs are stored in software depots. A software depot is a hierarchy of files and folders and can be available through an HTTP or HTTPS URL address (online depot) or a ZIP file (offline depot).

A host profile includes the configuration profile used by a specific ESXi host. Host profiles can be extracted from one ESXi host and applied to a different host that you want to apply the same configuration to. For more information on host profiles, see [vSphere Host Profiles](#).

Read the following topics next:

- [Configuring and the Auto Deploy Plug-In](#)
- [Create and Apply an Auto Deploy Rule](#)
- [Auto Deploy Plug-In Workflow Library](#)

Configuring and the Auto Deploy Plug-In

You must configure the Auto Deploy plug-in before you can begin using it to manage your ESXi hosts.

Prerequisites

- 1 Enable the Auto Deploy service in vCenter.
 - Log in to vCenter and then navigate to **Home > Auto Deploy**. By default, only users with an administrator role have the privileges to use the **vCenter Auto Deploy Service**.

- On the Auto Deploy page, click **Enable Auto Deploy and Image Builder** to activate the service. If the **Image Builder** service is already active, select the **Configure** tab and click **Enable Auto Deploy Service**.

- 2 Add a vCenter to your vRealize Orchestrator deployment.

Log in to the vRealize Orchestrator Client and run the **Add a vCenter Server instance** workflow included with the vCenter plug-in for vRealize Orchestrator.

Note Each instance of the Auto Deploy plug-in is associated with a single vCenter instance.

Procedure

- 1 Log in to the vRealize Orchestrator Client.
- 2 Add a vCenter host and image profile depot.
 - a Select the **Add an Auto Deploy host** workflow.
 - b Select the vCenter that you have added to your vRealize Orchestrator deployment and click **Run**.
 - c Select the **Add a depot** workflow.
 - d Enter a depot name, enter the depot URL address, and click **Run**.

Note You can find the official VMware depot [here](#).

You have added a depot to your Auto Deploy plug-in. You can view the image profiles hosted on the depot by navigating to the **Inventory** page and expanding the **Auto Deploy** entry.

Create and Apply an Auto Deploy Rule

Modify your ESXi host by creating and applying an Auto Deploy rule.

The following use case describes how you can use the Auto Deploy plug-in for vRealize Orchestrator to create and apply a simple deploy rule that adds an image profile to one of your ESXi hosts and changes the vSphere Client folder where the host is located.

Prerequisites

Configure the Auto Deploy plug-in. See [Configuring and the Auto Deploy Plug-In](#).

Procedure

- 1 Log in to the vRealize Orchestrator Client.
- 2 Create a deployment rule.
 - a Select the **Create a deploy rule** workflow.
 - b Select your Auto Deploy host, enter a name for your rule, and select an image profile.

- c (Optional) Apply an extracted host profile.

By extracting a host profile from an ESXi host, you can apply the host configuration, including settings like virtual switch configurations, to any host to which you apply the deploy rule. To extract a host profile, log in to the vSphere Client, right-click the host, and select **Host Profiles > Extract Host Profile**.

- d (Optional) Select a host pattern.

Selecting a pattern can be useful when you want to specify the type of host to which you want to apply the deploy rule. For example, you can specify that you want the rule to apply to hosts that use a specific hardware vendor.

- e Under **Type**, select **Host Folder**.

- f Under **Host folder**, select the vCenter folder where you want to move the host when the rule is activated and the host is re-provisioned.

- g Click **Run**.

3 Validate that the rule is added to the vCenter.

- a Log in to the vSphere Client as an **administrator**.
- b Under the top-left menu, select **Auto Deploy** and navigate to the **Deploy Rules** tab.

4 Activate the deploy rule.

- a Log in to the vRealize Orchestrator Client.
- b Select the **Activate a deploy rule and a working set** workflow.
- c Select your deploy rule, and click **Run**.
- d (Optional) Under **Position**, select the rule order.

Note An ESXi host can match more than one vSphere Auto Deploy rule criteria. In this case, the rule order set in the **Position** field is considered.

- e Click **Run**.

After the workflow finishes running, the status of the rule in the vSphere Client changes to **Active**. To apply the rule to a host, you must set the host to run in maintenance mode, and then re-provision the host.

5 Set the host to maintenance mode.

- a Select the **Enter maintenance mode** workflow included with the vCenter plug-in for VMware vRealize Orchestrator.
- b Select your host.
- c Under **VSAN host decommission mode**, select **noAction**.
- d Click **Run**.

6 Re-provision your host.

- a Select the **Reprovision a host by a simple reboot** workflow.
- b Select your Auto Deploy host and ESXi host to which you want to apply the deploy rule.
- c Click **Run**.

The new image profile is applied and the host is moved to the vSphere Client folder specified in step 3.

7 Exit maintenance mode.

- a Select the **Exit maintenance mode** workflow.
- b Select your host, and click **Run**.

Results

You have created and applied a deploy rule to one of your vCenter ESXi hosts.

Troubleshooting

If you add or modify an object, such as a deploy rule, directly from the vCenter instance for which you have an Auto Deploy host, the plug-in can't detect and fetch changes in the vCenter.

To re-fetch all new or modified objects, remove the Auto Deploy host and then add it back again.

Auto Deploy Plug-In Workflow Library

The Auto Deploy plug-in for vRealize Orchestrator contains out-of-the-box workflows.

To access these workflows, navigate to **Library > Workflows** and enter the **auto_deploy** in the workflow search box.

Answer Files Workflows

Workflow	Description
Clear and answer file	Clears the content of an answer file associated with a specified ESXi host.
Get an answer file	Retrieves an answer file associated with a specific ESXi host.
Get answer file status	Retrieves the status of an answer file associated with a specific ESXi host. The possible states for the answer file are valid, invalid, and unknown.
Update an answer file	Updates an answer file from XML content. Creates an answer file when it is missing.

Configuration Workflows

Workflow	Description
Add an Auto Deploy host	Adds a new Auto Deploy host for an already configured vCenter instance.
Remove an Auto Deploy host	Removes an already configured Auto Deploy host.
Add a depot	Adds a public depot.
Remove a depot	Removes a public depot.
Update a depot	Updates a public depot.
Get deploy options	Retrieves the Auto Deploy global configuration options.
Reload plug-in configuration	Reloads the list of public depots. You can run this workflow after you import a package of public depots.
Set deploy option	Sets a value for an Auto Deploy global configuration option.

Re-provision Host Workflows

Workflow	Description
Apply image profile on hosts	Associates a specified image profile with the specified ESXi hosts.
Reprovision a host by a simple reboot	Reprovisions the host by repairing the rule set compliance and rebooting the host.
Reprovision a host with a host profile and an answer file	Reprovisions the ESXi host with a new host profile and an answer file. Validates the answer file before the ESXi host reboot.
Reprovision a host with an answer file	Reprovisions the host by updating or creating an answer file. Validates the answer file before remediating the rule set compliance, and rebooting the host.
Reprovision a host with a new image	Reprovisions the host by changing a specific deployment rule with a new image profile, repairing rule set compliance, and rebooting the host.
Reprovision a host with a new location	Reprovisions the host by changing a specific deploy rule with a new vCenter location, repairing the rule set compliance, and rebooting the host.

Rules Workflows

Workflow	Description
Activate a deploy rule and a working set	Activates a deploy rule by adding it to both working and active rule sets. All rules already in the working rule set are activated too.
Activate a working set	Activates all rules from a working rule set by moving them to an active set.
Add to working set	Adds a rule to a working rule set.
Copy a deploy rule	Replaces the original rule by creating a new copy and applying all the modifications. As a result, the old rule is hidden and the new modified rule takes the place of the original rule.
Create a deploy rule	Creates a new deploy rule.
Delete a deploy rule	Deletes a deploy rule from the Auto Deploy host.
Delete all hidden rules	Deletes all hidden rules from the rule engine.
Edit a deploy rule	Modifies a deploy rule on a specific Auto Deploy host.
Get deploy rules	Retrieves all deploy rules from the rule engine of the Auto Deploy server.
Get hidden rules	Retrieves all hidden rules from the rule engine of the Auto Deploy server.
Get host matching rules	Retrieves all rules which apply to a host.
Get rules from active set	Retrieves all rules from an active rule set.
Get rules from working set	Retrieves all rules from a working rule set.
Remove a rule from a rule set	Removes a rule from a rule set without deleting the rule.

Rule Set Compliance Workflows

Workflow	Description
Repair an active rule set compliance	Remediates the host to use the revised rule set the next time you boot the host. Updates the image profile, host profile, and location for the specified host in the vCenter inventory.
Test a rule set compliance	Checks whether the items associated with a specified ESXi host are in compliance with a rule set.

Other Workflows

Workflow	Description
Get host attributes	Retrieves the ESXi host attributes which are used when the Auto Deploy server evaluates the rules.

Using the Configuration Plug-In

6

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

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

SSL Trust Manager Workflows

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

To access these workflows in the vRealize Orchestrator Client, navigate to **Library > Workflows** and enter the **configuration** and **ssl_trust_manager** tags in the workflow search box.

Workflow Name	Description
Delete a trusted certificate	Deletes an SSL certificate from the server trust store.
Import certificate from URL	Imports an SSL certificate from a URL into the server trust store.
Import a certificate from URL using authenticated proxy server	Imports an SSL certificate from a URL that is reachable through an authenticated proxy server.
Import certificate from URL using proxy server	Imports an SSL certificate from a URL that is reachable through a proxy server.
Import certificate from URL with certificate alias	Imports an SSL certificate from a URL into the server trust store.
Import trusted certificate from a file	Imports an SSL certificate from a file into the server trust store.

Keystore Workflows

To access these workflows in the vRealize Orchestrator Client, navigate to **Library > Workflows** and enter the **configuration** and **keystores** tags in the workflow search box.

Workflow Name	Description
Add certificate	Adds a certificate to a keystore.
Add key	Adds a key.

Workflow Name	Description
Create a keystore	Creates a new keystore.
Delete a keystore	Deletes a keystore.
Delete certificate	Deletes a certificate from a keystore.
Delete entry	Deletes an entry.
Delete key	Deletes a key.

Using the Dynamic Types Plug-In

7

The vRealize Orchestrator Dynamic Types plug-in allows you to define dynamic types, create objects of these types, and set relations between them. You can use the Dynamic Types plug-in to expose third-party objects as custom types in the vRealize Orchestrator Scripting API.

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.

- 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.

Read the following topics next:

- [Dynamic Types Configuration Workflows](#)

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.

To access these workflows in the vRealize Orchestrator Client, navigate to **Library > Workflows** and enter the **dynamic_types** tag in the workflow search box.

Workflow Name	Description
Define Namespace	Defines a new namespace.
Define Relation	Defines a new relation between types.
Define Type	Defines a new type within a given namespace.
Export Configuration As Package	Exports a Dynamic Types definition configuration as a file-based configuration. The exported package can be used for importing to other servers.
Import Configuration From Package	Imports a file-based configuration to the plug-in configuration.
Import Type Definitions From XSD	Imports type definitions from an XSD file.
Remove Namespace	Removes a namespace.
Remove Relation	Removes a relation.
Remove Type	Removes a type.
Update Namespace	Updates a namespace.
Update Type	Updates a type.

Using the vRealize Orchestrator Plug-In for F5 BIG-IP



The VMware vRealize Orchestrator Plug-in for F5 BIG-IP replicates the entire F5 REST API, providing workflows and actions that enable users to automate and configure their F5 environments from vRealize Orchestrator.

The vRealize Orchestrator Plug-in for F5 BIG-IP provides nearly 200 out-of-the-box workflows for common F5 administrative tasks. For a full list of available workflows, see [F5 Plug-In Workflow Library](#).

System requirements

Before installing the plug-in on your vRealize Orchestrator 8.x deployment, ensure your system meets the following requirements.

F5 BIG-IP Requirements	
Version	F5 BIG-IP 14.x, 15.x, 16.x
Connection	Hostname (Management IP or DNS name) of the F5 BIG-IP system
Credentials	User name and password with Administrative level access

Installing the F5 plug-in

Download the plug-in installation file from the [VMware Marketplace](#). For instructions how to install the plug-in, see [Install or Update a vRealize Orchestrator Plug-In](#).

Configuring the F5 plug-in

After the plug-in has been installed, you must configure the plug-in to an F5 BIG-IP instance. See [Run the Attach BIG-IP Workflow](#).

Accessing the F5 plug-in API

The vRealize Orchestrator Plug-in for F5 BIG-IP also provides more than 900 scripting objects available for creating your own custom workflows. To view all of the available scripting objects that are available with the plug-in, navigate to the **API Explorer**.

You can search by specific F5 object names and keywords, or scroll through the list to browse. Select a scripting object to view related properties.

Read the following topics next:

- [Run the Attach BIG-IP Workflow](#)
- [F5 Plug-In Workflow Library](#)

Run the Attach BIG-IP Workflow

The **Attach BIG-IP** workflow configures an F5 BIG-IP endpoint to pull in the necessary F5 objects.

Prerequisites

Download and install the vRealize Orchestrator Plug-in for F5 BIG-IP. See [Chapter 8 Using the vRealize Orchestrator Plug-In for F5 BIG-IP](#)

Procedure

- 1 Log in to the vRealize Orchestrator Client.
- 2 Navigate to **Library > Workflows** and enter **Attach BIG-IP** in the workflow search box.
- 3 Locate the **Attach BIG-IP** workflow and click **Run**.
- 4 On the **General** tab, enter the following information.
 - The name of your F5 BIG-IP instance.
 - The hostname or IP address of your F5 BIG-IP instance.
 - The user name of your F5 BIG-IP instance.
 - The password associated with the user name of your F5 BIG-IP instance.
- 5 On the **Advanced Parameters** tab, configure the following settings.
 - Enter the interval length in minutes for data collection.
 - Enter the maximum number of threads that you want to be used during data collection.
 - (Optional) Decide how SSL is used during collection. Select one of the following options.
 - **No Verify**. Trust all server certificates.
 - **Verify**. Validate the certificate against the Java trust store.
 - **No SSL**. Do not use SSL.
 - Enter the timeout value in seconds for API requests.
- 6 On the **Collection Parameters** tab, set the modules for which you want configuration data collected to **true**.

7 Click **Run**.

After the workflow finishes, a green checkmark appears next to the workflow indicating it was successful.

Note If the workflow fails, a red X appears next to the workflow, and errors are logged at the bottom of the screen.

What to do next

- Verify the F5 Networks Inventory Tree. After your F5 BIG-IP instance is configured, click the **Inventory** tab to ensure that the F5 objects appear in the F5 Networks Inventory Tree.
- Run any of the other workflows provided with the plug-in. See [F5 Plug-In Workflow Library](#) for the full list of available workflows.

F5 Plug-In Workflow Library

The vRealize Orchestrator Plug-in for F5 BIG-IP contains out-of-the-box workflows.

To access these workflows, navigate to **Library > Workflows** and enter the **f5_** tag in the workflow search box.

Category	Workflows
Asm	<ul style="list-style-type: none"> ■ ASM Activate Policy ■ ASM Assign Policy to VIP ■ ASM Blocking/Transparent Policy ■ ASM Export Policy ■ ASM Install Policy
Auth Partition	<ul style="list-style-type: none"> ■ Create Partition ■ Delete Partition
Basic	<ul style="list-style-type: none"> ■ Add Device to Device Group ■ Attach BIG-IP ■ Change Device Name ■ Create Device Group ■ Detach BIG-IP ■ License BIG-IP ■ License Plugin ■ Make REST Call ■ Provision Module ■ Save Configuration ■ Sync Device Group ■ Update Management IP and Route ■ Update Plugin License from 2.0

Category	Workflows
Gtm	<ul style="list-style-type: none"> ■ Create DNS Datacenter ■ Create DNS Link ■ Create DNS Listener ■ Create DNS Pool ■ Create DNS Server ■ Create iRule ■ Create Wide-IP ■ Remove DNS Datacenter ■ Remove DNS Link ■ Remove DNS Listener ■ Remove DNS Pool A ■ Remove DNS Pool AAAA ■ Remove DNS Pool CNAME ■ Remove DNS Pool MX ■ Remove DNS Pool NAPTR ■ Remove DNS Pool SRV ■ Remove DNS Server
Net Route	<ul style="list-style-type: none"> ■ Create/Delete Route ■ Create/Delete Route Domain ■ Create Route Domain Member ■ Delete Route ■ Delete Route Domain

Category	Workflows
Ltm	<ul style="list-style-type: none"> ■ Create DNS Zone ■ Instantiate App Services iApp ■ Upload/Add iRule ■ Upload/Install iApp <p>Monitor</p> <ul style="list-style-type: none"> ■ Create LTM Monitor Diameter ■ Create LTM Monitor DNS ■ Create LTM Monitor External ■ Create LTM Monitor Firepass ■ Create LTM Monitor FTP ■ Create LTM Monitor Gateway ICMP ■ Create LTM Monitor HTTP ■ Create LTM Monitor HTTPS ■ Create LTM Monitor ICMP ■ Create LTM Monitor IMAP ■ Create LTM Monitor Inband ■ Create LTM Monitor LDAP ■ Create LTM Monitor Module Score ■ Create LTM Monitor MSSQL ■ Create LTM Monitor MySQL ■ Create LTM Monitor NNTP ■ Create LTM Monitor Oracle ■ Create LTM Monitor Pop3 ■ Create LTM Monitor PostgreSQL ■ Create LTM Monitor Radius ■ Create LTM Monitor Radius Accounting ■ Create LTM Monitor Real Server ■ Create LTM Monitor RPC ■ Create LTM Monitor SASP ■ Create LTM Monitor Scripted ■ Create LTM Monitor SIP ■ Create LTM Monitor SMB ■ Create LTM Monitor SMTP ■ Create LTM Monitor SNMP DCA ■ Create LTM Monitor SNMP DCA BASE ■ Create LTM Monitor SOAP ■ Create LTM Monitor TCP ■ Create LTM Monitor TCP Echo ■ Create LTM Monitor TCP Half Open ■ Create LTM Monitor UDP ■ Create LTM Monitor Virtual Location ■ Create LTM Monitor WAP ■ Create LTM Monitor WMI ■ Remove LTM Monitor Diameter ■ Remove LTM Monitor DNS

Category	Workflows
	<ul style="list-style-type: none"> ■ Remove LTM Monitor External ■ Remove LTM Monitor Firepass ■ Remove LTM Monitor FTP ■ Remove LTM Monitor Gateway ICMP ■ Remove LTM Monitor HTTP ■ Remove LTM Monitor HTTPS ■ Remove LTM Monitor ICMP ■ Remove LTM Monitor IMAP ■ Remove LTM Monitor Inband ■ Remove LTM Monitor LDAP ■ Remove LTM Monitor Module Score ■ Remove LTM Monitor MSSQL ■ Remove LTM Monitor MySQL ■ Remove LTM Monitor NNTP ■ Remove LTM Monitor Oracle ■ Remove LTM Monitor Pop3 ■ Remove LTM Monitor PostgreSQL ■ Remove LTM Monitor Radius ■ Remove LTM Monitor Radius Accounting ■ Remove LTM Monitor Real Server ■ Remove LTM Monitor RPC ■ Remove LTM Monitor SASP ■ Remove LTM Monitor Scripted ■ Remove LTM Monitor SIP ■ Remove LTM Monitor SMB ■ Remove LTM Monitor SMTP ■ Remove LTM Monitor SNMP DCA ■ Remove LTM Monitor SNMP DCA BASE ■ Remove LTM Monitor SOAP ■ Remove LTM Monitor TCP ■ Remove LTM Monitor TCP Echo ■ Remove LTM Monitor TCP Half Open ■ Remove LTM Monitor UDP ■ Remove LTM Monitor Virtual Location ■ Remove LTM Monitor WAP ■ Remove LTM Monitor WMI
	Node
	<ul style="list-style-type: none"> ■ Create Node ■ Delete Node
	Pool
	<ul style="list-style-type: none"> ■ Create Pool ■ Create Pool Member ■ Create SNAT Pool ■ Delete Pool ■ Delete Pool Member

Category	Workflows
	<ul style="list-style-type: none"> ■ Disable Pool Member ■ Enable Pool Member ■ Get Pool Member by Name ■ Get Pool Members ■ Get Pool Member Stats ■ Get Pools <p>Profile</p> <ul style="list-style-type: none"> ■ Create Client SSL Profile ■ Create Server SSL Profile ■ Delete Client SSL Profile ■ Delete Server SSL Profile <p>Virtual Server</p> <ul style="list-style-type: none"> ■ Add iRule to Virtual Server ■ Add Persistence Profile to Virtual Server ■ Add Protocol Profile to Virtual Server ■ Add Standard Profile to Virtual Server ■ Create Virtual Server ■ Delete Virtual Server ■ Duplicate Virtual Server ■ Remove Profile from Virtual Server ■ Set Firewall Policy on Virtual Server ■ Set Virtual Server SNAT
Security	<ul style="list-style-type: none"> ■ Create AFM Address List ■ Create AFM Port List ■ Create AFM Rule ■ Create AFM Rules List ■ Create AFM Schedule ■ Create Firewall Policy ■ Remove AFM Address List ■ Remove AFM Port List ■ Remove AFM Rule ■ Remove AFM Rules List ■ Remove AFM Schedule
Sys	<ul style="list-style-type: none"> ■ Set DNS Settings ■ Set NTP Settings ■ Set Syslog Settings <p>SSL</p> <ul style="list-style-type: none"> ■ Create SSL Cert ■ Create SSL Key ■ Delete SSL Cert ■ Delete SSL Key ■ Upload/Install Certificate

Using the HTTP-REST Plug-In

9

The HTTP-REST plugin 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.

Security Hardening

Sensitive information is considered internal to the corresponding HTTP-REST plug-in objects and intended for use only in scenarios involving these objects. Therefore, specific scripting properties and methods that might be used to access sensitive information are removed. Instead, you can store sensitive information in vRealize Orchestrator configuration elements or external password vaults.

Read the following topics next:

- [Persistent and Transient REST Hosts](#)
- [Configuring the HTTP-REST Plug-In](#)
- [Generate a New Workflow from a REST Operation](#)
- [Invoke a REST Operation](#)

Persistent and Transient REST Hosts

The HTTP-REST plug-in supports two types of REST hosts that you can use to make requests to REST endpoints - persistent hosts and transient hosts.

Differences between persistent and transient hosts

The following table compares the two types of REST hosts.

Persistent Hosts	Transient Hosts
Stored in the vRealize Orchestrator database.	Not stored in the vRealize Orchestrator database. Transient hosts are virtual objects which reside in memory while a script is being executed.
Stored in the vRealize Orchestrator inventory. Persistent hosts can also be viewed in form drop-down menus of the RESTHost type.	Not stored in the vRealize Orchestrator inventory.
Available after restart, failover, and upgrade. When a workflow token gets interrupted, it can continue from where it left off if the workflow item takes a persistent REST host as input. Use persistent hosts as inputs/outputs of workflow items. You can create them at the beginning of the scripting and delete them if you don't need them anymore.	Not available after restart and failover. When a workflow gets interrupted, it can't restore a workflow item input which carries a transient REST host. Use transient hosts in scripting when you make isolated requests against a server which you otherwise don't use.
Can be exported and imported as resource elements.	Transferable across different vRealize Orchestrator instances because they are created and managed entirely from scripting. Use transient hosts when you work on multiple environments with no need to migrate persistent hosts.
Each persistent host has a dedicated HTTP client that is used for managing requests to the endpoint.	Hosts reuse the same HTTP client instance.
Parallel requests are supported for persistent and transient hosts. <ul style="list-style-type: none"> ■ If you activate parallel requests, each request is executed with a separate context, and the state, including cookies, is not preserved between requests. ■ If support for parallel requests is deactivated, consecutive requests share the same HTTP context. 	

Considerations for transient hosts

When creating transient hosts, consider the following.

- Transient hosts passed between workflow items as input/output might not work in all cases. Transient hosts rely on workflow cache, which doesn't work when asynchronous workflows are started, for example. Nested workflows might also fail.
- Only `GET` and `HEAD` requests get redirected automatically. URL redirection uses the `default` strategy.
- Host name verification is not supported.
- Client certificate authentication is not supported.

Troubleshooting

If you use transient hosts without support for parallel requests, you might experience scripting regressions after upgrading your vRealize Orchestrator environment or upgrading the HTTP-REST plug-in to version 2.4.1.19272162 or later. Using different transient host instances to run requests, which depend on one another for cookies, is not supported as of vRealize Orchestrator 8.7.

To avoid this problem, use one of the following methods.

- 1 Instead of transient hosts, use persistent hosts and operations. You can create persistent REST hosts in one of two ways.

- a Create a REST host pointing to the server using the **Add a REST host** workflow.

Instead of using transient hosts, use the REST host as input everywhere where you need to create a request to it.

- Do not create transient operations pointing to this host. Create regular operations instead.
- Support for parallel requests must be deactivated, otherwise cookies are not preserved.

This approach is not recommended if you make multiple requests in parallel to this host in your workflows.

- b Create a REST host per workflow run from the scripting, and then delete it.

Use this method if you make parallel requests to the server. For example, if you have two parallel requests, create two different hosts.

- 1 Clone a workflow.
 - 2 Add a scripting element which creates the host that you want to use for future requests.
 - 3 Use the host as the output of the workflow and as input to all other scripting that makes requests to that host.
 - 4 To clean the state, add an element at the end of the scripting which deletes the host you created.
- 2 Use one transient host for all dependent requests in a given workflow and pass it between workflow items as input/output as needed.

Passing transient hosts between multiple workflow elements is not officially supported, but expected to work. Note that during restart, the workflow state might be lost and the workflow might not resume successfully.

If you use transient hosts and you want to make requests which depend on each other for cookies, you must use the same transient host instance for all requests. If the requests span multiple workflow items, create the host in the first workflow item and then pass it as input to the rest.

- 3 Use your current transient hosts, but modify the failing requests to include the necessary cookies by adding the respective headers.

You might have to parse the cookies from the previous response and use them in subsequent requests.

Configuring the HTTP-REST Plug-In

You can configure HTTP-REST by running the configuration workflows included in the plug-in. The Configuration workflow category contains workflows that help you to manage REST hosts.

To access these workflows in the vRealize Orchestrator Client, navigate to **Library > Workflows** and enter the **http-rest** and **configuration** tags in the workflow search box.

Workflow Name	Description
Add a REST host	Adds a REST host to the plug-in inventory.
Add a REST host by Swagger spec as a string	Adds a REST host based on a Swagger spec web resource provided as a string.
Add a REST host by Swagger spec from a URL	Adds a REST host based on a Swagger spec available at a specific URL.
Add a REST operation	Adds an operation to a REST host.
Add schema to a REST host	Adds an XSD schema to a REST host.
Clone a REST host	Creates a clone of a REST host.
Clone a REST operation	Creates a clone of a REST operation.
Reload plug-in configuration	Refreshes the list of REST hosts in the plug-in inventory.
Remove a REST host	Removes a REST host from the plug-in inventory.
Remove a REST operation	Removes an operation from a REST host.
Remove schemas form a REST host	Removes all associated XSD schemas from a REST host.
Update a REST host	Updates a REST host in the plug-in inventory.
Update a REST operation	Updates an operation on a REST host.

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 vRealize Orchestrator Client as an administrator.
- 2 Navigate to **Library > Workflows** and enter the **http-rest** and **configuration** tags in the workflow search box.
- 3 Locate the **Add a REST host** workflow and click **Run**.
- 4 On the **Host Properties** tab, enter the properties of the new host.
 - a In the **Name** text box, enter the name of the host.
 - b In the **URL** text box, enter the address of the host.

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

- c In the **Connection timeout** text box, enter the number of seconds before a connection times out.
- d In the **Operation timeout** text box, enter the number of seconds before an operation times out.
- e Select whether you want to accept the REST host certificate and add it to the vRealize Orchestrator server trust store.
- f Select whether you want to be able to run parallel requests.
- g Select a redirect strategy.

Option	Description
defaultRedirect	Redirects only when the <code>HEAD</code> and <code>GET</code> methods are used.
alwaysRedirect	Redirects all HTTP methods, including <code>POST</code> , <code>PUT</code> , <code>DELETE</code> , and so on.
neverRedirect	Forbids all redirects and does not follow 3xx status codes with any HTTP method.

- 5 On the **Host Authentication** tab, select the authentication type.

Option	Description
None	No authentication is required.
OAuth 1.0	On the OAuth 1.0 tab, provide the required authentication parameters.
OAuth 2.0	<p>On the OAuth 2.0 tab, provide the authentication token and select a token sending strategy.</p> <ul style="list-style-type: none"> ■ If you select Authorization header, the token is sent in the authorization header of each request made to the host as a bearer token. ■ If you select Query parameter, the token is sent in the <code>oauth_token</code> query parameter of each request made to the host. <p>Alternatively, you can configure the authentication settings using the vRealize Orchestrator scripting API. For instructions, see Using the Scripting API to Configure or Update a REST Host Authentication.</p>
Basic	<p>Provides basic access authentication.</p> <p>On the User credentials tab, select the session mode.</p> <ul style="list-style-type: none"> ■ If you select Shared Session, provide credentials for the shared session. ■ If you select Per User Session, the vRealize Orchestrator client retrieves credentials from the user who is logged in.
Digest	<p>Provides digest access authentication that uses encryption.</p> <p>On the User credentials tab, select the session mode.</p> <ul style="list-style-type: none"> ■ If you select Shared Session, provide credentials for the shared session. ■ If you select Per User Session, the vRealize Orchestrator client retrieves credentials from the user who is logged in.

Option	Description
NTLM	<p>Provides NT LAN Manager (NTLM) access authentication within the Window Security Support Provider (SSPI) framework.</p> <p>On the User credentials tab, select the session mode.</p> <ul style="list-style-type: none"> ■ If you select Shared Session, provide credentials for the shared session. ■ If you select Per User Session, the vRealize Orchestrator client retrieves credentials from the user who is logged in. <p>On the NTLM tab, provide the NTLM settings.</p>
Kerberos	<p>Provides Kerberos access authentication.</p> <p>On the User credentials tab, select the session mode.</p> <ul style="list-style-type: none"> ■ If you select Shared Session, provide credentials for the shared session. ■ If you select Per User Session, the vRealize Orchestrator client retrieves credentials from the user who is logged in.

6 (Optional) On the **Proxy Settings** tab, select whether to use a proxy server.

- Enter the address and the port of the proxy server.
- Select the proxy authentication type.

Option	Description
None	No authentication is required.
Basic	<p>Provides basic access authentication.</p> <p>On the Proxy Credentials tab, select the session mode.</p> <ul style="list-style-type: none"> ■ If you select Shared Session, provide credentials for the shared session. ■ If you select Per User Session, the vRealize Orchestrator client retrieves credentials from the user who is logged in.

- On the **SSL** tab, select whether you want the target hostname to match the name stored in the server certificate.
- (Optional) Select a keystore entry to use to authenticate against the server. The keystore entry must be of the `PrivateKeyEntry` type.
- Click **Run**.

Results

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. You can execute requests to the REST host based on the created REST operations using the default REST plug-in workflows, or using the vRealize Orchestrator scripting.

Add a REST Operation

You can run a workflow to add an operation to a REST host from the plug-in inventory.

Prerequisites

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

Procedure

- 1 Navigate to **Library > Workflows** and enter the **http-rest** and **configuration** tags in the workflow search box.
- 2 Locate the **Add a REST operation** workflow and click **Run**.
- 3 Select the parent host to which you want to add the operation.
- 4 In the **Name** text box, enter the name of the operation.
- 5 In the **Template URL** text box, enter 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}
```

- 6 Select the HTTP method that the operation uses.
If you select **POST** or **PUT**, you can provide a Content-Type request header for the method.
- 7 Click **Run**.

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 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 vRealize Orchestrator Client as an administrator.
- Verify that you have a connection to a REST host from the **Inventory** view.

Procedure

- 1 Navigate to **Library > Workflows** and enter the **http-rest** and **configuration** tags in the workflow search box.
- 2 Locate the **Add a schema to a REST host** workflow and click **Run**.
- 3 On the **Host** tab, select the host to which you want to add the XSD schema.

- 4 On the **XSD Schema Details** tab, select whether to load the schema from URL.

Option	Action
Yes	Enter the URL of the schema.
No	Provide the schema content.

- 5 Click **Run**.

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 *Developing Workflows with vRealize Orchestrator* guide.

Prerequisites

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

Procedure

- 1 Navigate to **Library > Workflows** and enter the **http-rest** tag in the workflow search box.
- 2 Locate the **Generate a new workflow from a REST operation** workflow and click **Run**.
- 3 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.
- 4 In the **Name** text box, type the name of the workflow to generate.
- 5 Select the workflow folder in which to generate the new workflow.
You can select any existing folder from the workflow library.
- 6 Click **Run**.

Invoke a REST Operation

Call a REST operation directly.

Prerequisites

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

Procedure

- 1 Navigate to **Library > Workflows** and enter the **http-rest** and **configuration** tags in the workflow search box.
- 2 Locate the **Invoke a REST operation** workflow and click **Run**.
- 3 On the **Operation** tab, select the REST operation from the list of available operations.
- 4 Provide the input parameters and content that the operation requires.
- 5 Click **Run**.

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 > Workflows**. 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 > Workflows**. 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.

Using the Library Plug-In

10

You can use the Library plug-in workflows as templates for customization and automation of client processes, and to troubleshoot vRealize Orchestrator. The Library plug-in provides workflows in the **Locking**, **Orchestrator**, and **Tagging** workflow categories.

Locking Workflows

To access these workflows in the vRealize Orchestrator Client, navigate to **Library > Workflows** and enter the **locking** tag in the workflow search box.

Workflow Name	Description
Display all locks	Shows all locks.
Locking test	A test workflow that creates a lock.
Locking test (x5)	A test workflow that creates five locks.
Release all locks	Releases all locks.

Task Workflows

To access these workflows in the vRealize Orchestrator Client, navigate to **Library > Workflows** and enter the **tasks** tag in the workflow search box.

Workflow Name	Description
Create recurrent task	Creates a recurrent task and returns the newly created task.
Create task	Schedules a workflow to run at a later time and date, as a task.

Orchestrator Workflows

To access these workflows in the vRealize Orchestrator Client, navigate to **Library > Workflows** and enter the **orchestrator** and **workflows** tags in the workflow search box.

Workflow Name	Description
Refresh stale workflow runs in waiting state	Processes all workflow runs that are in waiting state for the specified remote server and updates the workflow state according to the remote workflow run. You can use this workflow if there is data loss between the workflow runs, for example, when there is loss of connectivity between the vRealize Orchestrator servers.
Start workflows in a series	Runs a workflow multiple times in a series, one instance after the other. You provide workflow parameters in an array. You also provide a property list, with one property per workflow input, for each instance of the workflow that starts. The number of properties in the array define the number of workflow runs.
Start workflows in parallel	Runs a workflow multiple times, with different parameters. You provide workflow parameters in an array. You also provide a property list, with one property per workflow input, for each instance of the workflow that starts. The number of properties in the array define the number of workflow runs.

Tagging Workflows

To access these workflows in the vRealize Orchestrator Client, navigate to **Library > Workflows** and enter the **tagging** tag in the workflow search box.

Workflow Name	Description
Find objects by tag	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.
List workflow tags	Lists the tags assigned to the workflow you specified as an input parameter.
Tag workflow	Assigns a tag to a workflow. You must specify the workflow you want to tag and the tag name and value.
Tagging example	Demonstrates workflow tagging.
Untag workflow	Removes a tag from a workflow. You must specify the workflow you want to untag and the tag you want to remove from the workflow.

Using the Mail Plug-In

11

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.

Read the following topics next:

- [Using the Mail Plug-In Sample Workflows](#)
- [Define the Default SMTP Connection](#)

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 vRealize Orchestrator and your SMTP server.

To access these workflows in the vRealize Orchestrator Client, navigate to **Library > Workflows** and enter the **mail** tag in the workflow search box.

Note Before you can access the workflows, verify that the user account that you are logged in with has the necessary permissions to run Mail workflows.

Workflow Name	Description
Configure mail	Defines the connection to the SMTP server, the SMTP authentication account, and the address and display name of the sender.
Retrieve messages	Retrieves the messages of a given email account by using the POP3 protocol.
Retrieve messages (via MailClient)	Retrieves the messages of a certain email account, without deleting them, by using the new scripting API provided by the <code>MailClient</code> class.
Send notification	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.
Send notification to mailing list	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.

What to read next

Define the Default SMTP Connection

The Mail plug-in is installed together with the vRealize 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 vRealize Orchestrator. Otherwise, you might receive `SMTP_HOST_UNREACHABLE` error.

Procedure

- 1 Log in to the vRealize Orchestrator client.
- 2 Navigate to **Library > Workflows** and enter the **mail** tag in the workflow search box.
- 3 Locate the **Configure mail** workflow and click **Run**.
- 4 On the **Host** tab, enter the IP address or domain name of your SMTP server and a port number to match your SMTP configuration.

The default SMTP port is 25.

- 5 On the **Credentials** tab, enter a user name and password for authentication.
Enter a valid email account and an associated password. vRealize Orchestrator uses the email account to send emails.
- 6 On the **Email Content** tab, enter a sender's email address and name.
The sender information appears in all emails sent by vRealize Orchestrator.
- 7 Click **Run**.

Using the Multi-Node Plug-In

12

The Multi-Node plug-in workflow library contains workflows for hierarchical orchestration, management of vRealize Orchestrator instances, and scale-out of vRealize Orchestrator activities.

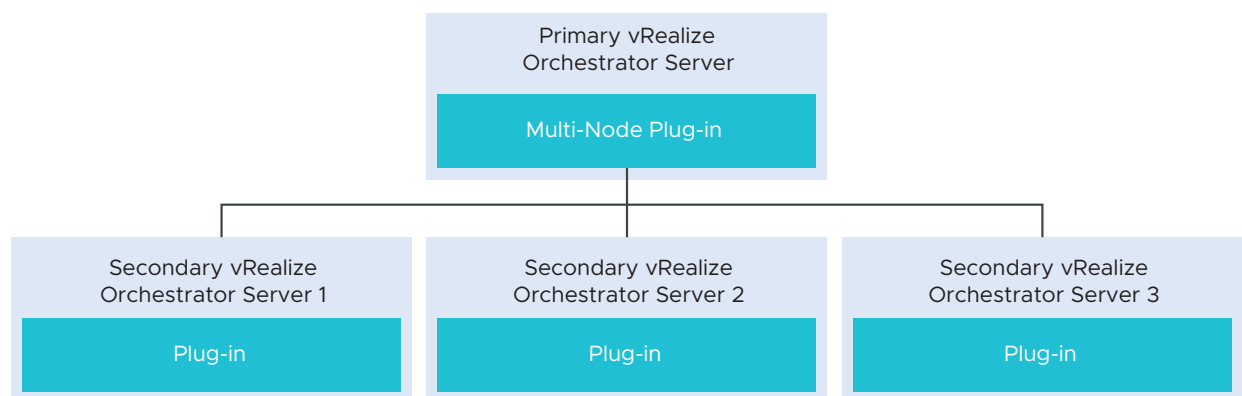
Read the following topics next:

- [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 primary-secondary relation between vRealize Orchestrator servers, which extends in the areas of package management and workflow execution.

Figure 12-1. Multi-Node Plug-In Schema



The plug-in contains a set of standard workflows for hierarchical orchestration, management of vRealize Orchestrator instances, and scale-out of vRealize Orchestrator activities.

Configuring the Multi-Node Plug-In

You must use the vRealize Orchestrator Client to configure the Multi-Node plug-in.

The Servers Configuration workflow category contains workflows that allow you to configure the connected vRealize Orchestrator servers.

To access these workflows, navigate to **Library > Workflows** and enter the **orchestrator** and **servers_configuration** tags in the workflow search box.

Workflow Name	Description
Add an Orchestrator server	Adds a vRealize Orchestrator server to the plug-in inventory.
Delete an Orchestrator server	Removes a vRealize Orchestrator server from the plug-in inventory and deletes all created proxies for this server.
Update an Orchestrator server	Updates a vRealize Orchestrator server from the plug-in inventory by changing its details.

Add an Orchestrator Server

You can run a workflow to establish a connection to a new vRealize Orchestrator server.

Prerequisites

Verify that the primary and secondary vRealize Orchestrator servers are the same version.

Procedure

- 1 Log in to the vRealize Orchestrator Client as an administrator.
- 2 Navigate to **Library > Workflows** and enter the **orchestrator** and **servers_configuration** tags in the workflow search box.
- 3 Locate the **Add an Orchestrator server** workflow and click **Run**.
- 4 On the **Server details** tab, provide the host address and the port of the remote vRealize Orchestrator server.
 - a Select whether the certificate is accepted silently and added to the trusted store.
 - b Select whether to generate proxy workflows for the remote vRealize Orchestrator server.
- 5 On the **Connection settings** tab, provide the connection settings.
 - a In the **Connection timeout** text box, enter the number of seconds within which vRealize Orchestrator must connect to the remote server, otherwise the connection times out.
 - b In the **Socket timeout** text box, enter the number of seconds within which the request must succeed before it times out.
 - c In the **Retry timeout** text box, enter the number of seconds within which the proxy workflows wait to receive a notification from the remote vRealize Orchestrator server if there is no connectivity.

- 6 On the **Connection mode** tab, select whether the connection is shared.

Option	Description
No	The credentials of the logged-in user are used to connect to the remote vRealize Orchestrator server.
Yes	All users can access the remote Orchestrator server using the same credentials. Provide the credentials for the shared connection.

- 7 Click **Run**.

Using Proxy Workflows

You can use proxy workflows to manage the interaction between the local vRealize Orchestrator server and workflows on a remote vRealize 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 finishes its run, the output parameters are converted back to the local representation on the primary vRealize 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 vRealize 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 vRealize 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 Standart Workflows

To access these workflows, navigate to **Library > Workflows** and enter the **orchestrator** and **remote_execution** tags in the workflow search box.

Workflow Name	Description
Create a multi proxy action	Creates a multi-proxy action to run workflows on multiple servers.
Create a proxy workflow	Creates a proxy workflow that you can use to start a workflow on a remote Orchestrator server.
Create proxy workflows from a folder	Creates proxy workflows for all workflows in a folder on the remote Orchestrator server.

Server Proxies

To access these workflows, navigate to **Library > Workflows** and enter the **orchestrator**, **remote_execution** and **server_proxies** tags in the workflow search box.

Workflow Name	Description
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.

The inventory for a single remote server consists 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 vRealize 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 vRealize Orchestrator instances.

Remote Management Packages

To access these workflows, navigate to **Library > Workflows** and enter the **orchestrator**, **remote_management** and **packages** tags in the workflow search box.

Workflow Name	Description
Delete a package	Deletes a package and its contents from a remote vRealize Orchestrator server.
Delete a package by name	Deletes a package and its contents by name on a remote vRealize Orchestrator server.
Deploy a package from a local server	Deploys a package from a local vRealize Orchestrator server to remote Orchestrator servers.
Deploy a package from a remote server	Deploys a package from one remote vRealize Orchestrator server to a list of remote Orchestrator servers.
Deploy packages from a local server	Deploys packages from a local vRealize Orchestrator server to remote vRealize Orchestrator servers.

Remote Management Workflows

To access these workflows, navigate to **Library > Workflows** and enter the **orchestrator**, **remote_management** and **workflows** tags in the workflow search box.

Workflow Name	
Delete a remote workflow	Deletes a workflow from a remote vRealize Orchestrator server.
Delete all finished workflow runs	Deletes all finished workflow runs from a remote workflow.
Deploy a workflow from a local server	Deploys a workflow from a local vRealize Orchestrator server to a list of remote Orchestrator servers.
Deploy a workflow from a remote server	Deploys a workflow from a remote vRealize Orchestrator server to a list of other remote Orchestrator servers.

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 vRealize Orchestrator Client as an administrator.
- 2 Access the API Explorer from either the vRealize Orchestrator Client or from the **Scripting** tabs of the workflow, policy, and action editors.
 - To access the API Explorer from the vRealize Orchestrator Client, click **API Explorer** in the vRealize Orchestrator Client navigation pane.
 - 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 vRealize 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 vRealize Orchestrator server at a later stage.

Procedure

- 1 Log in to the vRealize Orchestrator Client as an administrator.

2 Navigate to **Library > Workflows** and enter the **orchestrator** and **remote_execution** tags in the workflow search box.

3 Locate the **Create a multi-proxy action** workflow and click **Run**.

4 On the **Action details** tab, enter the action name and module.

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.

5 On the **Workflow details** tab, select whether the workflow is local or remote.

Option	Description
Yes	Select the remote workflow that you want to use for this action.
No	Select the local workflow that you want to use for this action.

6 Click **Run**.

Results

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.

To access the workflows for managing the proxy workflows in the vRealize Orchestrator Client, navigate to **Library > Workflows** and enter the **orchestrator**, **remote_execution**, and **servers_proxies** tags in the workflow search box.

Workflow Name	Description
Refresh proxy workflows for an Orchestrator server	Regenerates all proxy workflows for the local vRealize Orchestrator server from the remote server.
Delete proxy workflows for an Orchestrator server	Removes the proxy workflows for the local vRealize Orchestrator server and deletes all generated workflows.

To access the workflows for further maintenance of the proxy workflows in the vRealize Orchestrator Client, navigate to **Library > Workflows** and enter the **orchestrator**, **remote_management**, and **workflows** tags in the workflow search box.

Workflow Name	Description
Delete all finished workflow runs	Deletes all finished workflow runs from a remote workflow.
Delete a remote workflow	Deletes a workflow from a remote vRealize Orchestrator server.
Deploy a workflow from a local server	Deploys a workflow from a local vRealize Orchestrator server to a list of remote vRealize Orchestrator servers.

Deploy a Package from a Local Server

You can run a workflow to deploy a package from a local vRealize Orchestrator server to remote vRealize 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 vRealize Orchestrator Client as an administrator.
- 2 Navigate to **Library > Workflows** and enter the **orchestrator** and **remote_management** tags in the workflow search box.
- 3 Locate the **Deploy a package from a local server** workflow and click **Run**.
- 4 Select the package to deploy from the local storage.
- 5 Select the remote servers to deploy the package to.
- 6 Select whether you want to overwrite the remote server packages.

Option	Description
Yes	The packages on the remote server are replaced, discarding the version of the packaged elements.
No	A version check of the server and the deploying packages is performed. The packages are deployed after a successful check.

- 7 Click **Run**.

Results

After running the workflow, the status information is displayed in the log view and in the inventory of the plug-in.

Using the Net Plug-In

13

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 PowerShell Plug-In

14

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

Read the following topics next:

- [Introduction to the vRealize Orchestrator PowerShell Plug-In](#)
- [Configuring the PowerShell Plug-In](#)
- [Access the PowerShell Plug-In API](#)
- [Using the PowerShell Plug-In Inventory](#)
- [Running PowerShell Scripts](#)
- [Generating Actions](#)
- [Passing Invocation Results Between Actions](#)
- [PowerCLI Integration with the PowerShell Plug-In](#)
- [Working with PowerShell Results](#)
- [Sample Workflows](#)
- [Examples of Scripts for Common PowerShell Tasks](#)
- [Troubleshooting](#)

Introduction to the vRealize Orchestrator PowerShell Plug-In

The PowerShell plug-in allows interaction between vRealize Orchestrator and Windows PowerShell.

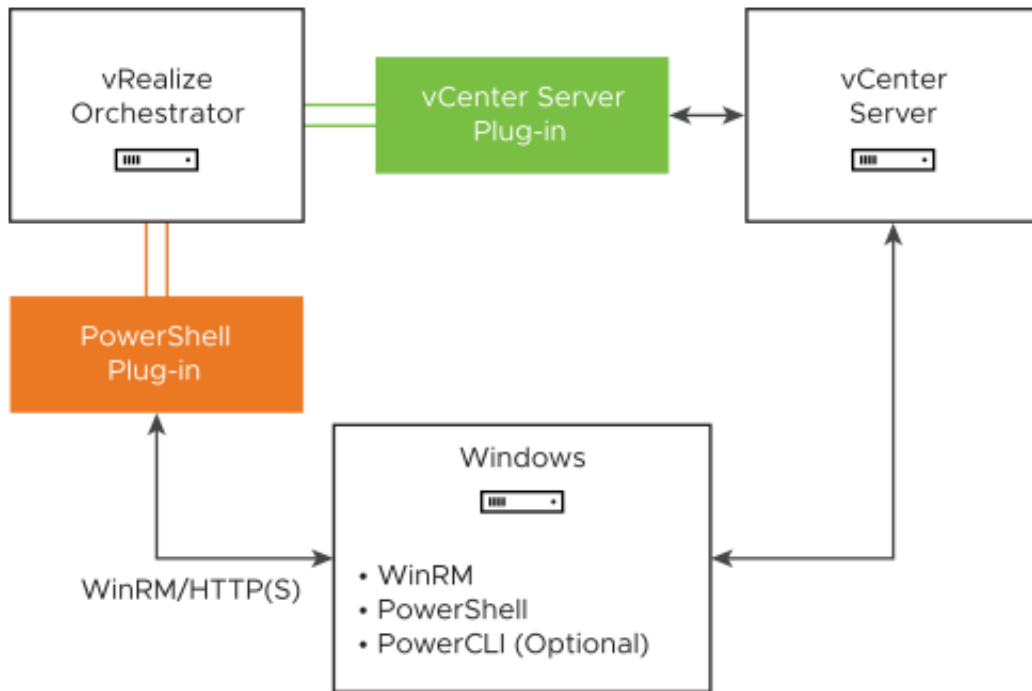
You use the plug-in to call PowerShell scripts and cmdlets from vRealize Orchestrator actions and workflows, and to work with the result. The plug-in contains a set of standard workflows. You can also create custom workflows that implement the plug-in API.

PowerShell Plug-In Components

The PowerShell plug-in relies on a number of components to function properly.

vRealize Orchestrator and Windows PowerShell provide the platform for the plug-in, and the plug-in provides interaction between those products. The PowerShell plug-in can also interact with other components, such as vCenter and vSphere PowerCLI.

Figure 14-1. Component Relations



The plug-in communicates with Windows PowerShell through WinRM communication protocol. See [Configuring WinRM](#).

Optionally, you can integrate the PowerShell plug-in with vSphere PowerCLI and vCenter. See [PowerCLI Integration with the PowerShell Plug-In](#).

Note You can install all components on a local host. The usage, functionality, and communication protocol requirements of the PowerShell plug-in do not change if vRealize Orchestrator and Windows PowerShell are installed on the same machine.

Configuring WinRM

To establish a connection between the PowerShell plug-in and Windows PowerShell, you must configure WinRM to use one of the supported communication protocols.

The PowerShell plug-in supports Windows Remote Management (WinRM) 2.0 as a management protocol.

The following authentication methods are supported.

Authentication method	Details
Basic	Non-secure authentication mechanism that requires a user name and a password.
Kerberos	Secure authentication protocol that uses tickets to verify the identity of the client and the server.

Note The PowerShell plug-in does not support delegation of user credentials in WinRM and CredSSP is not a supported authentication method.

WinRM Through HTTP

The PowerShell plug-in supports communication with the WinRM host through the HTTP protocol. Although WinRM authenticates the communication, the data transfer is not encrypted and is sent as plain text on the network. You should use the HTTP protocol if IPsec is configured between the machines that communicate.

To use Basic authentication, you must set the `AllowUnencrypted` property to `true` in both the service and client WinRM configuration. For an example of HTTP configuration, see [Configure WinRM to Use HTTP](#).

WinRM Through HTTPS

The PowerShell plug-in supports communication with the WinRM host through the HTTPS protocol. You can use the HTTPS protocol as a more secure communication method.

To use the HTTPS protocol, you must generate a certificate for server authentication and install the certificate on the WinRM host. For an example of HTTPS configuration, see [Configure WinRM to use HTTPS](#).

Kerberos authentication

You can use Kerberos authentication when you add and manage a PowerShell host. With Kerberos authentication, domain users can run commands on remote PowerShell-enabled machines over WinRM.

To configure WinRM on the PowerShell host, run the following command.

```
winrm quickconfig
winrm set winrm/config/service/auth '@{Kerberos="true"}'
winrm set winrm/config/winrs @{MaxMemoryPerShellMB="2048"}
```

For more information, see [Configure Kerberos authentication for vRealize Orchestrator plug-ins](#).

Configure WinRM to Use HTTP

You can configure the WinRM host to enable communication with the PowerShell plug-in through the HTTP protocol.

You must modify the WinRM configuration by running commands on the WinRM host machine. You can use the same machine as both the WinRM service and WinRM client.

Important If you skip any of the steps when configuring WinRM to use HTTP, the host might not be added, and you might get an error message in the logs such as:

```
Caused by: org.dom4j.DocumentException: Error on line -1 of document : Premature end of file.
Nested exception: Premature end of file.
at org.dom4j.io.SAXReader.read(SAXReader.java:482)
at org.dom4j.DocumentHelper.parseText(DocumentHelper.java:278)
at
com.xebialabs.overthere.cifs.winrm.connector.JdkHttpConnector.sendMessage(JdkHttpConnector.java:117)
```

Procedure

- 1 Run the following command to set the default WinRM configuration values.

```
c:\> winrm quickconfig
```

- 2 (Optional) Run the following command to check whether a listener is running, and verify the default ports.

```
c:\> winrm e winrm/config/listener
```

The default ports are 5985 for HTTP, and 5986 for HTTPS.

- 3 Enable basic authentication on the WinRM service.

- a Run the following command to check whether basic authentication is allowed.

```
c:\> winrm get winrm/config/service
```

- b Run the following command to enable basic authentication.

```
c:\> winrm set winrm/config/service/auth '@{Basic="true"}'
```

- 4 Run the following command to allow transfer of unencrypted data on the WinRM service.

```
c:\> winrm set winrm/config/service @{AllowUnencrypted="true"}
```

- 5 If the channel binding token hardening level of the WinRM service is set to **strict**, change its value to **relaxed**.

```
c:\> winrm set winrm/config/service/auth @{CbtHardeningLevel="relaxed"}
```

- 6 Enable basic authentication on the WinRM client.

- a Run the following command to check whether basic authentication is allowed.

```
c:\> winrm get winrm/config/client
```

- b Run the following command to enable basic authentication.

```
c:\> winrm set winrm/config/client/auth @{Basic="true"}
```

- 7 Run the following command to allow transfer of unencrypted data on the WinRM client.

```
c:\> winrm set winrm/config/client @{AllowUnencrypted="true"}
```

- 8 If the WinRM host machine is in an external domain, run the following command to specify the trusted hosts.

```
c:\> winrm set winrm/config/client @{TrustedHosts="host1, host2, host3"}
```

- 9 Run the following command to test the connection to the WinRM service.

```
c:\> winrm identify -r:http://winrm_server:5985 -auth:basic -u:user_name  
-p:password -encoding:utf-8
```

Configure WinRM to use HTTPS

You can configure the WinRM host to enable communication with the PowerShell plug-in through the HTTPS protocol.

The WinRM host requires a certificate so that it can communicate through the HTTPS protocol. You can either obtain a certificate or generate one.

Prerequisites

- Configure WinRM to use the HTTP protocol. For more information, see [Configure WinRM to Use HTTP](#).
- Verify that you can access the Microsoft Management Console (`mmc.exe`) on the WinRM host.

Procedure

- 1 Generate a self-signed certificate.

The following command line contains example syntax for creating a certificate on the WinRM host by using the Powershell Cmdlet `New-SelfSignedCertificate`.

```
New-SelfSignedCertificate -CertStoreLocation cert:\localmachine\my -DnsName  
("machinename.subdomain.company.com", "machinename") -NotAfter (get-date).AddYears(5)  
-Provider "Microsoft RSA SChannel Cryptographic Provider" -KeyLength 2048
```

In this example, the `-NotAfter` parameter specifies that the certificate will expire in 5 years. If this parameter isn't specified, the certificate expires in 1 year. Learn more about [creating self-signed certificates](#).

- 2 Add the generated certificate by using the Microsoft Management Console.
 - a Run `mmc.exe`.
 - b Select **File > Add/Remove Snap-in**.
 - c From the list of available snap-ins, select **Certificates** and click **Add**.
 - d Select **Computer account** and click **Next**.

- e Click **Finish**.
- f Verify that the certificate is installed in **Console Root > Certificates (Local Computer) > Personal > Certificates** and **Console Root > Certificates (Local Computer) > Trusted Root Certification Authorities > Certificates**.

If the certificate is not installed in the Trusted Root Certification Authorities and Personal folders, you must install it manually.

- 3 Create an HTTPS listener by using the correct thumbprint and host name.

The following command line contains example syntax for creating an HTTPS listener.

```
winrm create winrm/config/Listener?Address=*&Transport=HTTPS
@{Hostname="host_name";CertificateThumbprint="certificate_thumbprint"}
```

Note Omit the spaces in the certificate thumbprint.

- 4 Test the connection.

The following command line contains example syntax for testing the connection.

```
winrs -r:https://host_name:port_number -u:user_name -p:password hostname
```

Configuring the PowerShell Plug-In

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

Configuration Workflows

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

To access these workflows, navigate to **Library > Workflows** and enter the **powershell** and **configuration** tags in the workflow search box.

Workflow Name	Description
Add a PowerShell host	Adds a PowerShell host to the plug-in inventory.
Remove a PowerShell host	Removes a PowerShell host from the plug-in inventory.
Update a PowerShell host	Updates the specified PowerShell host in the plug-in inventory.
Validate a PowerShell host	Validates the configuration of the specified PowerShell host.

Add a PowerShell host

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

Procedure

- 1 Log in to the vRealize Orchestrator Client as an administrator.

- 2 Navigate to **Library > Workflows** and enter the **powershell** and **configuration** tags in the workflow search box.
- 3 Locate the Add a PowerShell host workflow and click **Run**.
- 4 In the **Name** text box, enter the name of the host.
- 5 In the **Host / IP** text box, enter the address of the host.

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

- 6 (Optional) In the **Port** text box, type the port of the host.
You use port 5985 for the HTTP or 5986 for the HTTPS protocol.
- 7 On the **Host Type** tab, specify 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 vRealize Orchestrator keystore.

- b Select the authentication type.

Important If you want to use Kerberos authentication, you must enable it on the WinRM service.

- 8 On the **User Credentials** tab, select the type of session mode that the plug-in uses to connect to the PowerShell host.

Option	Description
Shared Session	The plug-in uses shared credentials to connect to the remote host. You must provide the PowerShell host credentials for the shared session. The PowerShell host credentials must belong to a member of the local administrators group.
Session per User	The vRealize Orchestrator client retrieves credentials from the user who is logged in. You must log in with a user@domain format to vRealize Orchestrator to use the Session per User mode.

- 9 On the **Advanced Options** tab, from the **Shell Code Page** drop-down menu, select the type of encoding that the PowerShell uses.
- 10 Click **Run**.

Results

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

Troubleshooting

If the workflow fails with the `Cannot Locate KDC (Dynamic Script Module name: addPowerShellHost#30)` error, see [Troubleshooting the workflow Add a PowerShell host in VMware Aria Automation Orchestrator](#).

Access the PowerShell Plug-In API

With the Orchestrator API Explorer you can 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 vRealize Orchestrator Client as an administrator.
- 2 Access the API Explorer from either the vRealize Orchestrator Client or from the **Scripting** tabs of the workflow, policy, and action editors.
 - To access the API Explorer from the vRealize Orchestrator Client, click **API Explorer** in the vRealize Orchestrator Client navigation pane.
 - 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*.

Using the PowerShell Plug-In Inventory

The PowerShell plug-in exposes all objects in the connected PowerShell hosts in the **Inventory** view.

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.

Running PowerShell Scripts

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

Invoke a PowerShellScript

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

Prerequisites

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

Procedure

- 1 Navigate to **Library > Workflows** and enter the **powershell** tag in the workflow search box.

- 2 Locate the Invoke a PowerShell script workflow and click **Run**.
- 3 On the **PowerShell Host** tab, select a PowerShell host on which to run the script.
- 4 On the **Script** tab, in the **Script** text box, type or paste the PowerShell script that you want to run.
- 5 Click **Run**.

Invoke an External Script

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

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

Prerequisites

- Verify that you are logged in to the vRealize 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 Navigate to **Library > Workflows** and enter the **powershell** tag in the workflow search box.
- 2 Locate the **Invoke an external script** workflow and click **Run**.
- 3 On the **PowerShell Host** tab, select a PowerShell host on which to run the script.
- 4 On the **External Script** tab, 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 file path. You can use system environment variables to specify script paths. For example, **\$env:HOME\PATH\test1.ps1**.

- 5 In the **Arguments** text box, type the script arguments.
The valid syntax is the same as used in the PowerShell console.
- 6 Click **Run**.

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 to replace the placeholder.

Prerequisites

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

Procedure

- 1 In the vRealize Orchestrator Client, navigate to **Library > Workflows** and enter the **powershell** and **generate** tags in the workflow search box.
- 2 Locate the **Generate an action from a PowerShell script** workflow and click **Run**.
- 3 On the **Script** tab, 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.

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

- 4 On the **Action** tab, in the **Name** text box, enter a name for the action that you want to generate, and select an existing module in which to generate the action.
- 5 On the **Workflow** tab, select whether to generate a workflow.

Option	Description
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 <code>Invoke Script</code> and the name of the generated action.
No	A sample workflow is not generated.

- 6 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 vRealize Orchestrator. You can optionally generate a sample workflow that runs 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 vRealize Orchestrator platform supports. The PowerShell plug-in defines the type mappings. In general, primitive types are mapped to the corresponding vRealize Orchestrator types, and complex types are represented by the `PowerShellRemotePSObject` object.

Prerequisites

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

Procedure

- 1 In the vRealize Orchestrator Client, navigate to **Library > Workflows** and enter the **powershell** and **generate** tags in the workflow search box.
- 2 Locate the **Generate an action for a PowerShell cmdlet** workflow and click **Run**.
- 3 On the **Cmdlet** tab, select a PowerShell cmdlet to run when using the action that you generate.
- 4 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.

- 5 On the **Action** tab, in the **Name** text box, enter a name for the action that you want to generate, and select an existing module in which to generate the action.
- 6 On the **Workflow** tab, select whether to generate a workflow.

Option	Description
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 <code>Execute Cmdlet</code> and the name of the generated action.
No	A sample workflow is not generated.

- 7 Click **Run**.

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 invoke 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 PowerCLI object. This feature allows workflows from the vCenter 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 plug-in. Unsupported types return an exception.

To access these workflows, navigate to **Library > Workflows** and enter the **powershell** and **converter** tags in the workflow search box.

Workflow Name	Description
Convert PObject to vCO object	Converts <code>PowerShellRemotePObject</code> to <code>VC:<SomeObjectType></code> .
Convert PObject to vCO object to PObject	Converts <code>PowerShellRemotePObject</code> to <code>VC:<SomeObjectType></code> and the reverse.
Convert vCO object to PObject	Converts <code>VC:<SomeObjectType></code> to <code>PowerShellRemotePObject</code> .

Working with PowerShell Results

You can use objects from the PowerShell plug-in API to work with results that Windows PowerShell returns.

You can use the methods from the `PowerShellInvocationResult` class to retrieve information about a script that you run.

Method	Description
<code>getErrors()</code>	Returns a list of errors reported by the PowerShell engine during script invocation.
<code>getInvocationState()</code>	Status of the script. The possible values are <code>Completed</code> or <code>Failed</code> .
<code>getHostOutput()</code>	Output of the script as it appears on the PowerShell console.
<code>getResults()</code>	Objects returned by the PowerShell engine. The returned object is of type <code>PowershellRemotePObject</code> .

`PowershellRemotePObject` is a remote representation of objects returned by the PowerShell engine. `PowershellRemotePObject` 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 easy 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 vRealize 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`.
- If the returned object is of type `complex`, the object is represented as `PObject`.

Sample Workflows

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

To access these workflows, navigate to **Library > Workflows** and enter the **powershell** and **samples** tags in the workflow search box.

Workflow Name	Description
Invoke a script via API	Demonstrates how to call a PowerShell script through the available scripting API.
List directory content	Lists the contents of a directory on the PowerShell host file system.
Pipeline execution example	Demonstrates how you can run multiple cmdlets arranged into a pipe.
Toggle virtual machine state	Toggles the power state of a virtual machine.

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 plug-in API.

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.

```
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.

```
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.

```
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 using the PowerShell plug-in, you can refer to a troubleshooting topic to understand the problem or solve it, if there is a workaround.

Activate Kerberos Event Logging

For troubleshooting purposes, you might want to activate Kerberos event logging on the Key Distribution Center (KDC) machine.

Prerequisites

Back up the Windows registry.

Procedure

- 1 Log in to the domain controller that acts as a Key Distribution Center (KDC).
- 2 Run the registry editor as an **administrator**.
- 3 In the registry window, expand
HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Control\Lsa\Kerberos\Parameters.
- 4 If a **LogLevel** registry key value does not exist, right-click to create it.
 - a Right-click **Parameter**, select **New > DWORD (32-bit) Value**, and enter **LogLevel**.
 - b Select **Parameter** and in the right pane, double-click **LogLevel** and enter **1** in the **Value data:** text box.

The new setting becomes effective without a reboot on Windows Server 2003 and later.

Results

The Kerberos error event entries are recorded in the System Windows Event Log.

What to do next

To deactivate Kerberos event logging, delete the **LogLevel** registry key value or change its value data to **0**.

Servers Not Found in Kerberos Database

After you add PowerShell 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 several misconfigurations.

- The PowerShell host is not part of a domain.
- The host to realm mapping is not correct.
- The Service Principal Name of the PowerShell host is not built correctly.

Note Kerberos authentication does not work when the destination is an IP address.

Solution

When you add a PowerShell host using the Kerberos authentication, enter 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 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 Different Time Settings

Inconsistent time settings 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 differs with more than 5 minutes from the domain controller, 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 inventory using Shared Session or Session per User, the workflow fails with the following error.

```
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 is provided as a part of the user name.

Solution

Provide a default realm in your Kerberos configuration file or include the realm in your user name when authenticating with 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.

Unable to Locate the Default Realm

vRealize Orchestrator workflows that require Kerberos authentication might fail if the Kerberos configuration file does not have the correct format or encoding.

Problem

Kerberos authentication cannot identify the default realm.

```
Cannot locate default realm
```

Cause

The Kerberos configuration file `krb5.conf` that you upload to the vRealize Orchestrator Appliance has been edited on a non-UNIX operating system. As a result, the format and the encoding might be incorrect.

Solution

In order for the vRealize Orchestrator appliance to read the `krb5.conf` file, the format of the file must be UNIX and the character encoding must be ANSI as UTF-8.

Using the SNMP Plug-In

15

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 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.

Read the following topics next:

- [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 vRealize 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.

To access these workflows in the vRealize Orchestrator Client, navigate to **Library > Workflows** and enter the `snmp` and `device_management` tags in the workflow search box.

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 inventory.
Unregister an SNMP device	Unregisters an SNMP device from the plug-in 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 vRealize Orchestrator Client.
- 2 Navigate to **Library > Workflows** and enter the **snmp** and **device_management** tag in the workflow search box.
- 3 Locate the **Register an SNMP device** workflow and click **Run**.
- 4 In the **Device address** text box, enter the IP address or DNS name of the SNMP device.

Note To establish a more reliable connection, use an IP address.

- 5 (Optional) In the **Name** text box, enter 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.

- 6 (Optional) To configure the advanced connection parameters, select the check box, and navigate to the **Advanced** tab.

- 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.

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

Note The support for SNMPv3 is deprecated.

- c In the **Community name** text box, enter a name for the device. The default name is **public**.

- 7 Click **Run**.

Results

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 registered SNMP devices, run, copy, and edit existing queries, and remove queries from SNMP devices. You can use SNMP queries as building blocks in more complex workflows.

Query Management Workflows

The Query Management workflow category contains workflows that allow you to manage SNMP queries.

To access these workflows in the vRealize Orchestrator Client, navigate to **Library > Workflows** and enter the **snmp** and **query_management** tags in the workflow search box.

Workflow Name	Description
Add a query to an SNMP device	Adds a query to an SNMP device.
Copy an SNMP query	Copies an SNMP query from one device to another.
Edit an SNMP query	Edits an existing SNMP query.
Remove a query from an SNMP device	Removes an SNMP query from a device.
Run an SNMP query	Runs a query against an SNMP device.

Add a Query to an SNMP Device

You can run a workflow to add a query to an SNMP device from the plug-in inventory.

Prerequisites

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

Procedure

- 1 Navigate to **Library > Workflows** and enter the **snmp** and **query_management** tags in the workflow search box.
- 2 Locate the **Add a query to an SNMP device** workflow and click **Run**.
- 3 Select the device to which you want to add the query.
- 4 From the **Type** drop-down menu, select the query type.
- 5 In the **OID** text box, enter 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.

- 6 (Optional) In the **Name** text box, enter a name for the query.

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

- 7 Click **Run**.

What to do next

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

Managing the SNMP Trap Host

vRealize Orchestrator can act as an SNMP listener. You can start and stop the SNMP trap host, and change the port on which vRealize Orchestrator listens for SNMP traps.

The SNMP plug-in supports SNMPv1 and SNMPv2c traps.

Note The support for SNMPv3 is deprecated.

Important When using a clustered vRealize Orchestrator deployment, you must send the SNMP trap to all nodes in the cluster.

Trap Host Management Workflows

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

To access these workflows in the vRealize Orchestrator Client, navigate to **Library > Workflows** and enter the **snmp** and **trap_host_management** tags in the workflow search box.

Workflow Name	Description
Set the SNMP trap port	Sets the port on which vRealize Orchestrator listens for SNMP traps.
Start the trap host	vRealize Orchestrator starts listening for SNMP traps.
Stop the trap host	vRealize Orchestrator stops listening for SNMP traps.

Add a SNMP trap port to the vRealize Orchestrator Appliance

Before running the **Set the SNMP trap port** workflow, you must add the port to the vRealize Orchestrator Appliance.

Procedure

- 1 Log in to the vRealize Orchestrator Appliance command line as **root**.
- 2 Create a variable for the SNMP trap port.

```
SNMP_PORT=port_value
```

Note The ports open for SNMP traps are 81 to 32767.

- 3 Run the `vracli network ports expose` command.

```
vracli network ports expose --proto udp --targetPort $SNMP_PORT $SNMP_PORT vco-app
```

Set the SNMP Trap Port

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

The default port for SNMP traps is 162. On Linux systems, you can open ports below 1024 only with superuser privileges. The ports open for SNMP traps are 81 to 32767.

Prerequisites

- Prepare the vRealize Orchestrator Appliance for the configuration of the SNMP trap port. See [Add a SNMP trap port to the vRealize Orchestrator Appliance](#).
- In the vRealize Orchestrator Client, select **Administration > Inventory** and verify that the SNMP device is connected.

Procedure

- 1 Log in to the vRealize Orchestrator Client as an **administrator**.
- 2 Navigate to **Library > Workflows** and enter the `snmp` and `trap_host_management` tags in the workflow search box.
- 3 Run the **Set the SNMP trap port** workflow.
- 4 Enter the port number on which vRealize Orchestrator listens for SNMP traps.
- 5 Click **Run**.

Results

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

Note If you upgrade your vRealize Orchestrator environment, you must run the vRealize Orchestrator Appliance and the **Set the SNMP trap port** workflow 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 vRealize Orchestrator Client as an administrator.
- Verify that you have a connection to an SNMP device from the **Inventory** view.

Procedure

- 1 Navigate to **Library > Workflows** and enter the **snmp** tag in the workflow search box.
- 2 Locate the **Wait for a trap on an SNMP device** workflow and click **Run**.
- 3 Select the device on which you want to wait for an SNMP trap.
- 4 (Optional) In the **OID** text box, enter 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.

- 5 Click **Run**.

Set an SNMP Trap Policy

You can set a policy to continuously listen for traps from an SNMP device that is already registered in the plug-in inventory.

Prerequisites

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

Procedure

- 1 In the vRealize Orchestrator Client, navigate to **Library > Policies**.
- 2 Open the **SNMP Trap policy** template and click **Run**.
- 3 Enter a name for the policy that you want to create.

- 4 (Optional) Enter a description for the policy.
- 5 Select an SNMP device for which to set the policy.
- 6 Click **Run** to create the policy.

The vRealize Orchestrator Client switches to **Run** perspective.

- 7 On the **Policies** view, right-click the policy that you created and select **Start policy**.

Results

The trap policy starts to listen for SNMP traps.

What to do next

You can edit the SNMP Trap policy.

Configure an SNMP Trap Host Policy

With the SNMP Trap Host policy listens for SNMP traps from hosts that might not be added as registered SNMP devices.

Prerequisites

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

Procedure

- 1 In the vRealize Orchestrator Client, navigate to **Library > Policies**.
- 2 Open the **SNMP Trap Host** policy template and click **Run**.
- 3 Enter a name for the policy that you want to create.
- 4 (Optional) Enter a description for the policy.
- 5 Select **Trap Host (Online)** from the inventory tree.
- 6 Click **Run** to create the policy.
The Orchestrator client switches to **Run** perspective.
- 7 Right-click the policy and select **Edit**.
- 8 In the **Scripting** tab, expand **host > OnTrapAll**.
 - a Select a workflow or a script to associate with the policy.
- 9 Click **Save and close** to apply the edited settings.
- 10 On the **Policies** view, right-click the policy that you edited and select **Start policy**.

What to do next

You can edit the SNMP Trap Host 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, and customize the scripting and permissions associated with the policy.

Prerequisites

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

Procedure

- 1 In the vRealize Orchestrator Client, navigate to **Library > Policies**.
- 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 permissions to a user or to a group 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.

To access these workflows in the vRealize Orchestrator Client, navigate to **Library > Workflows** and enter the **snmp** tag in the workflow search box.

Workflow Name	Description
Get bulk SNMP values	Runs a GET BULK query against an SNMP device.
Get next SNMP value	Runs a GET NEXT query against an SNMP device.
Get SNMP value	Runs a GET query against an SNMP device.
Send an SNMP trap	Sends an SNMP trap to a specified address.
Wait for a trap on all devices	Waits to receive an SNMP trap from all hosts that send traps to vRealize Orchestrator.
Wait for a trap on an SNMP device	Waits to receive an SNMP trap from a specified device.

Using the SOAP Plug-In

16

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.

Read the following topics next:

- [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 vRealize Orchestrator Client to configure the SOAP plug-in.

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

To access these workflows in the vRealize Orchestrator Client, navigate to **Library > Workflows** and enter the **soap** and **configuration** tags in the workflow search box.

Workflow Name	Description
Add a SOAP host	Adds a SOAP host to the plug-in inventory.
Reload plug-in configuration	Refreshes the list of SOAP hosts in the plug-in inventory.
Remove a SOAP host	Removes a SOAP host from the plug-in inventory. Caution When you remove a host from the inventory, all workflows generated from it stop working.
Update a SOAP host	Updates a SOAP host in the plug-in inventory.
Update a SOAP host with an endpoint URL	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.

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 vRealize Orchestrator Client as an administrator.
- 2 Navigate to **Library > Workflows** and enter the **soap** and **configuration** tags in the workflow search box.
- 3 Locate the **Add a SOAP host** workflow and click **Run**.
- 4 On the **SOAP host** tab, provide the properties to create a host.
 - a In the **Name** text box, enter the name of the host.
 - b Select whether to provide the WSDL content as text.

Option	Action
Yes	Copy the text in the WSDL content text box.
No	Enter the correct path in the WSDL URI text box.

- c In the **Connection timeout** text box, enter the number of seconds within which vRealize Orchestrator must connect to the SOAP host, otherwise the connection times out.
 - d In the **Request timeout** text box, enter the number of seconds within which a SOAP request must succeed before it times out.
- 5 On the **Proxy** tab, select whether to use a proxy.

Option	Action
Yes	Provide the proxy address and proxy port.
No	Continue to the next step.

- 6 On the **Host authentication** tab, select the authentication type.

Option	Description
None	No authentication is required.
Basic	<p>Provides basic access authentication.</p> <p>On the User credentials tab, select the session mode.</p> <ul style="list-style-type: none"> ■ If you select Shared Session, provide credentials for the shared session. ■ If you select Per User Session, the vRealize Orchestrator Client retrieves credentials from the user who is logged in.
Digest	<p>Provides digest access authentication that uses encryption.</p> <p>On the User credentials tab, select the session mode.</p> <ul style="list-style-type: none"> ■ If you select Shared Session, provide credentials for the shared session. ■ If you select Per User Session, the vRealize Orchestrator Client retrieves credentials from the user who is logged in.

Option	Description
NTLM	<p>Provides NT LAN Manager (NTLM) access authentication within the Window Security Support Provider (SSPI) framework.</p> <p>On the User credentials tab, select the session mode.</p> <ul style="list-style-type: none"> ■ If you select Shared Session, provide credentials for the shared session. ■ If you select Per User Session, the vRealize Orchestrator Client retrieves credentials from the user who is logged in. <p>On the NTLM settings tab, enter the NTLM domain and NTLM workstation.</p>
Negotiate	<p>Provides Kerberos access authentication.</p> <p>On the User credentials tab, select the session mode.</p> <ul style="list-style-type: none"> ■ If you select Shared Session, provide credentials for the shared session. ■ If you select Per User Session, the vRealize Orchestrator Client retrieves credentials from the user who is logged in. <p>On the Negotiate tab, provide the Kerberos service SPN.</p>

7 Click **Run**.

Results

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.

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 *Developing Workflows with vRealize Orchestrator* guide.

Prerequisites

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

Procedure

- 1 Navigate to **Library > Workflows** and enter the **soap** tag in the workflow search box.
- 2 Locate the **Generate a new workflow from a SOAP operation** workflow and click **Run**.
- 3 Select the SOAP operation from the list of available operations.
- 4 In the **Name** text box, enter the name of the workflow to generate.
- 5 Select the workflow folder in which to generate the new workflow.

You can select any existing folder from the workflow library.

- 6 Click **Run**.

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 vRealize Orchestrator Client as an administrator.
- Verify that you have a connection to a SOAP host from the **Inventory** view.

Procedure

- 1 Navigate to **Library > Workflows**.
- 2 Locate the custom workflow and click **Run**.
- 3 Provide the input parameters that the SOAP operation requires.
- 4 Click **Run**.
- 5 (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 vRealize Orchestrator Client as an administrator.
- Verify that you have a connection to a SOAP host from the **Inventory** view.

Procedure

- 1 Navigate to **Library > Workflows** and enter the **soap** tag in the workflow search box.
- 2 Locate the **Invoke a SOAP operation** workflow and click **Run**.
- 3 Select the SOAP operation from the list of available operations.
- 4 Provide the input parameters that the SOAP operation requires.
- 5 Click **Run**.
- 6 (Optional) In the **Logs** tab, review the list of available output parameters.

Using the SQL Plug-In

17

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 plug-in also provides sample workflows that demonstrate how to use the API in workflows.

Read the following topics next:

- [Configuring the SQL Plug-In](#)
- [Running the SQL Sample Workflows](#)
- [Running SQL Operations](#)
- [Adding a JDBC Connector for the vRealize Orchestrator SQL Plug-In](#)

Configuring the SQL Plug-In

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

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

To access these workflows in the vRealize Orchestrator Client, navigate to **Library > Workflows** and enter the **sql** and **configuration** tags in the workflow search box.

Workflow Name	Description
Add a database	Adds a database object to the SQL plug-in inventory.
Add tables to a database	Adds database tables to a database in the SQL plug-in inventory.
Remove a database	Removes a database object from the SQL plug-in inventory.
Remove a table from a database	Removes a database table from a database in the SQL plug-in inventory.
Update a database	Updates the configuration of a database object in the SQL plug-in inventory.
Validate a database	Validates a database in the SQL plug-in inventory.

Add a Database

You can run a workflow to add a database to the vRealize 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.

Note You might need to install a third party connector to use the database if it is not already installed. For more information, go to [Adding a JDBC Connector for the vRealize Orchestrator SQL Plug-In](#).

Procedure

- 1 Log in to the vRealize Orchestrator Client.
- 2 Navigate to **Library > Workflows** and enter the **sql** and **configuration** tags in the workflow search box.
- 3 Locate the **Add a database** workflow and click **Run**.
- 4 In the **Name** text box, enter the name of the database.
- 5 Select the type of the database.
- 6 In the **Connection URL** text box, enter the address of the database.

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

- 7 On the **User credentials** tab, select the session mode that the plug-in uses to connect to the database.

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 vRealize 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. Do not use `domain\user` or `user@domain` for authentication.

8 Click **Run**.

Results

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

Add Tables to a Database

You can run a workflow to add tables to a database that is in the SQL plug-in inventory.

Prerequisites

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

Procedure

- 1 Navigate to **Library > Workflows** and enter the **sql** and **configuration** tags in the workflow search box.
- 2 Locate the **Add tables to a database** workflow and click **Run**.
- 3 Select a database to which to add tables.
- 4 Select the tables that you want to add.
- 5 Click **Run**.

Results

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

Update a Database

You can run a workflow to update the configuration of a database that is in the SQL plug-in inventory.

Procedure

- 1 Log in to the vRealize Orchestrator Client.
- 2 Navigate to **Library > Workflows** and enter the **sql** and **configuration** tags in the workflow search box.
- 3 Locate the **Update a database** workflow and click **Run**.
- 4 Select the database that you want to update.
- 5 In the **Name** text box, enter the new name of the database.
The database appears in the **Inventory** view with the name that you entered.
- 6 Select the type of the database.

- 7 In the **Connection URL** text box, enter the new address of the database.
- 8 On the **User credentials** tab, select the session mode that the plug-in uses to connect to the database.

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 vRealize 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. Do not use <i>domain\user</i> or <i>user@domain</i> for authentication.	

- 9 Click **Run**.

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, and to run SQL operations.

Generate a JDBC URL

You can run a workflow from the vRealize 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 Log in to the vRealize Orchestrator Client.
- 2 Navigate to **Library > Workflows** and enter the **jdbc** tag in the workflow search box.
- 3 Locate the **JDBC URL generator** workflow and click **Run**.
- 4 On the **General** tab, select the type of database for which to generate a URL.

Note If you use a Microsoft database, select the **Microsoft** tab and provide the database instance name and database user domain name.

- 5 Provide the required information to generate a database URL.
 - a Enter a database server name or IP address.
 - b Enter a database name.

- c (Optional) Enter a database port number.

If you do not specify a port number, the workflow uses a default port number.

- d Enter a user name to access the database.
- e Enter a password to access the database.

- 6 Click **Run**.

Test a JDBC Connection

You can run a workflow from the vRealize 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 Log in to the vRealize Orchestrator Client.
- 2 Navigate to **Library > Workflows** and enter the `jdbc_examples` tag in the workflow search box.
- 3 Locate the **JDBC connection example** workflow and click **Run**.
- 4 Provide the required information to test a database connection.
 - a Enter a user name to access the database.
 - b Enter the URL to test.
 - c Enter a password to access the database.
- 5 Click **Run**.

Create a Table by Using JDBC

You can run a workflow from the vRealize 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 Log in to the vRealize Orchestrator Client.
- 2 Navigate to **Library > Workflows** and enter the `jdbc_examples` tag in the workflow search box.
- 3 Locate the **JDBC create table example** workflow and click **Run**.

- 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 Enter an SQL create statement.

Example syntax is:

```
CREATE TABLE "table_name"
("column1" "data_type_for_column1",
"column2" "data_type_for_column2")
```

- 6 Click **Run**.

Insert a Row into a JDBC Table

You can run a workflow from the vRealize 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 Log in to the vRealize Orchestrator Client.
- 2 Navigate to **Library > Workflows** and enter the `jdbc_examples` tag in the workflow search box.
- 3 Locate the **JDBC insert into table example** workflow and click **Run**.
- 4 On the **Database connection** tab, provide the required information.
 - a Type a database connection URL.
 - b Type a user name to access the database.
 - c Type a password to access the database.
- 5 On the **SQL statement** tab, enter an SQL insert statement similar to the following example.

```
INSERT INTO "table_name" ("column1", "column2")
VALUES ("value1", "value2")
```

- 6 On the **Values to insert** tab, enter the values to insert into the row.
- 7 Click **Run**.

Select Rows from a JDBC Table

You can run a workflow from the vRealize 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 Log in to the vRealize Orchestrator Client.
- 2 Navigate to **Library > Workflows** and enter the `jdbc_examples` tag in the workflow search box.
- 3 Locate the **JDBC select from table example** workflow and click **Run**.
- 4 On the **Database connection tab**, provide the required information.
 - a Type a database connection URL.
 - b Type a user name to access the database.
 - c Type a password to access the database.
- 5 On the **SQL statement** tab, type an SQL select statement similar to the following example.
Example syntax is:

```
SELECT * FROM "table_name"
```

- 6 Click **Run**.

Delete an Entry from a JDBC Table

You can run a workflow from the vRealize 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 Navigate to **Library > Workflows** and enter the `jdbc_examples` tag in the workflow search box.
- 2 Locate the **JDBC delete entry from table example** workflow and click **Run**.
- 3 Provide the required information.
 - a Enter the first name of the user entry to be deleted.
 - b Type a user name to access the database.
 - c Enter a JDBC connection URL.

- d Enter the last name of the user entry to be deleted.
 - e Type a password to access the database.
- 4 Enter an SQL delete statement similar to the following example syntax.

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

- 5 Click **Run**.

Delete All Entries from a JDBC Table

You can run a workflow from the vRealize 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 Navigate to **Library > Workflows** and enter the **jdbc_examples** tag in the workflow search box.
- 2 Locate the **JDBC delete all from table example** workflow and click **Run**.
- 3 Provide the required information.
 - a Type a database connection URL.
 - b Type a user name to access the database.
 - c Type a password to access the database.
- 4 Type an SQL delete statement similar to the following example syntax.

```
DELETE FROM "table_name"
```

- 5 Click **Run**.

Drop a JDBC Table

You can run a workflow from the vRealize 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 Navigate to **Library > Workflows** and enter the **jdbc_examples** tag in the workflow search box.

- 2 Locate the **JDBC drop table example** workflow and click **Run**.
- 3 Provide the required information.
 - a Type a password to access the database.
 - b Type a database connection URL.
 - c Type a user name to access the database.
- 4 Enter an SQL drop statement similar to the following example syntax.

```
DROP TABLE "table_name"
```

- 5 Click **Run**.

Run a Complete JDBC Cycle

You can run a workflow from the vRealize 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 Navigate to **Library > Workflows** and enter the `jdbc_examples` tag in the workflow search box.
- 2 Locate the **Full JDBC cycle example** workflow and click **Run**.
- 3 Provide the required information.
 - a Type a database connection URL.
 - b Type a user name to access the database.
 - c Type a password to access the database.
- 4 Enter the values to be used as entries in the database.
- 5 Click **Run**.

Running SQL Operations

You can use the SQL workflows to run SQL operations.

To access the SQL operations workflows in the vRealize Orchestrator Client, navigate to **Library > Workflows** and enter the `sql` tag in the workflow search box.

Workflow Name	Description
Execute a custom query on a database	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.
Generate CRUD workflows for a table	Generates Create, Read, Update, and Delete workflows for a particular table.
Read a custom query from a database	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.

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 vRealize Orchestrator Client as an administrator.
- Verify that you have a connection to a database from the **Inventory** view.

Procedure

- 1 Navigate to **Library > Workflows** and enter the **sql** tag in the workflow search box.
- 2 Locate the **Generate CRUD workflows for a table** workflow and click **Run**.
- 3 Select a table for which to generate the workflows.
- 4 Select the workflow folder in which to generate the workflows.
- 5 Select whether to overwrite any existing workflows.

Option	Description
Yes	The generated workflows overwrite existing workflows with the same name.
No	New workflows are not generated if workflows with the same name exist in the folder.

- 6 (Optional) On the **Select read-only columns** tab, select columns that should not be populated.

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

- 7 Click **Run**.

Results

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.

Adding a JDBC Connector for the vRealize Orchestrator SQL Plug-In

This example demonstrates how you can add a MySQL connector for the vRealize Orchestrator SQL plug-in.

Prerequisites

The vRealize Orchestrator SQL plug-in supports only certain database types. Before adding a MySQL connector, verify that you are using one of the following database types:

- Oracle
- Microsoft SQL Server
- PostgreSQL
- MySQL

Procedure

- 1 Add the MySQL connector.jar file to the vRealize Orchestrator Appliance.

Note For clustered vRealize Orchestrator deployments, perform this operation on the appliances of all the nodes.

- a Log in to the vRealize Orchestrator Appliance command line over SSH as **root**.
- b Navigate to the `/data/vco/var/run/vco` directory.

```
cd /data/vco/var/run/vco
```

- c Create a `plugins/SQL/lib/` directory.

```
mkdir -p plugins/SQL/lib/
```

- d Copy your MySQL connector.jar file from your local machine to the `/data/vco/var/run/vco/plugins/SQL/lib/` directory by running a secure copy (SCP) command.

```
scp ~/local_machine_dir/your_mysql_connector.jar root@orchestrator_FQDN_or_IP:/data/vco/var/run/vco/plugins/SQL/lib/
```

Note You can also use alternative methods for copying your connector.jar file to the vRealize Orchestrator Appliance, such as PSCP.

- 2 Add the new MySQL property to the Control Center.
 - a Log in to the Control Center as **root**.
 - b Select **System Properties**.
 - c Click **New**.

- d Under **Key**, enter `o11n.plugin.SQL.classpath`.
- e Under **Value**, enter `/var/run/vco/plugins/SQL/lib/your_mysql_connector.jar`.

Note The value text box can include multiple JDBC connectors. Each JDBC connector is separated by a semicolon (";"). For example:

```
/var/run/vco/plugins/SQL/lib/your_mysql_connector.jar;/var/run/vco/plugins/SQL/lib/your_mssql_connector.jar;/var/run/vco/plugins/SQL/lib/your_other_connector.jar
```

- f (Optional) Enter a description for the MySQL system property.
- g Click **Add**, and wait for the vRealize Orchestrator server to restart.

Note Do not save your JDBC connector.jar file in another directory and do not set a different value to the `o11n.plugin.SQL.classpath` property. Otherwise, the JDBC connector becomes unavailable to your vRealize Orchestrator deployment.

Using the SSH Plug-In

18

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

Read the following topics next:

- [Configuring the SSH Plug-In](#)
- [Running the SSH Plug-In Sample Workflows](#)

Configuring the SSH Plug-In

You can run the SSH plug-in configuration workflows to manage the connections between vRealize Orchestrator and SSH hosts.

To access these workflows in the vRealize Orchestrator, navigate to **Library > Workflows** and enter the **ssh** and **configuration** tags in the workflow search box.

Workflow Name	Description
Add a Root Folder to SSH Host	Adds a root folder to an existing connection to an SSH host.
Add SSH Host	Adds a connection to an SSH host to the existing configuration.
Remove a Root Folder from SSH Host	Removes a root folder from an existing connection to an SSH host.
Remove SSH Host	Removes an existing connection to an SSH host from the existing configuration.
Update SSH Host	Updates an existing connection to an SSH host.

Add an SSH Host

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

Procedure

- 1 Log in to the vRealize Orchestrator Client.
- 2 Navigate to **Library > Workflows** and enter the **ssh** and **configuration** tags in the workflow search box.
- 3 Locate the **Add SSH host** workflow and click **Run**.

- 4 On the **General Information** tab, in the **Host name** text box, enter the name of the host that you want to access with SSH through vRealize Orchestrator.
- 5 Enter the target port. The default SSH port is 22.
The host is added to the list of SSH connections.
- 6 (Optional) Configure an entry path on the server.
 - a Click **New root folder**.
 - b Enter the new path and click **Insert value**.
- 7 On the **Authentication** tab, enter the user name for a user who has the necessary permissions to run SSH commands.
- 8 Select the authentication type.

Option	Action
Yes	To use password authentication, enter a password.
No	To use key authentication, enter the path to the private key and the private key passphrase.

- 9 Click **Run**.

Results

The SSH host is available in the **Inventory** view of the vRealize Orchestrator Client.

Running the SSH Plug-In Sample Workflows

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

■ [Generate a Key Pair](#)

You can generate a key pair by running a workflow in the vRealize Orchestrator Client.

Alternatively, you can use a key pair that you generate in third-party tools. 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 vRealize Orchestrator Client to change the passphrase for the key pair that you generated most recently.

■ [Register a vRealize Orchestrator Public Key on an SSH Host](#)

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

■ [Run an SSH Command](#)

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

- [Copy a File to an SSH Host](#)

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

Generate a Key Pair

You can generate a key pair by running a workflow in the vRealize Orchestrator Client. Alternatively, you can use a key pair that you generate in third-party tools. 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. vRealize Orchestrator can use the private key to connect to the public key on an SSH host.

Generate a Key Pair in vRealize Orchestrator

You can run a workflow from the vRealize Orchestrator Client to generate a key pair.

Caution All vRealize 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 Log in to the vRealize Orchestrator Client.
- 2 Navigate to **Library > Workflows** and enter the **ssh** tag in the workflow search box.
- 3 Locate the **Generate key pair** workflow and click **Run**.

4 Provide the required information.

a Select the key type and the key size.

Depending on which key type you select during the key-pair generation, you might have to change the SSH service endpoint configuration to enable the type of certificate generated.

Key Type	Key Sizes
RSA	<ul style="list-style-type: none"> ■ 1024 ■ 2048 ■ 3072 ■ 4096 ■ 8192
DSA	<ul style="list-style-type: none"> ■ 1024 ■ 2048 ■ 3072
ECDSA	<ul style="list-style-type: none"> ■ 256 ■ 384 ■ 521

b (Optional) Enter a passphrase.

You can use a passphrase to improve security. You can change the passphrase later.

c (Optional) Enter a comment.

5 Click **Run**.

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

Use an Externally Generated Key Pair

You can generate public or private key pairs using external tools such as OpenSSH or OpenSSL.

When generating RSA, DSA, and ECDSA private keys, the key must be in the PEM format.

Procedure

- 1 To generate key pairs with an external tool, you must pass specific parameters to the commands of each tool.

Tool	Parameters
OpenSSH	<p>Use the <code>-m pem</code> parameter.</p> <p>For example, you can run the following command:</p> <pre>ssh-keygen -m pem -t ecdsa -b {256, 384, 521} -f vco_key</pre>
OpenSSL	<p>The default format is PEM, so no additional parameters are needed.</p> <p>For example, you can run the following command:</p> <pre>openssl ecparam -name {prime256v1, secp384r1, secp521r1} -genkey -noout -out vco_key</pre> <p>Using OpenSSH, extract the corresponding public key from the private key.</p> <pre>ssh-keygen -f vco_key -y > vco_key.pub</pre>

- 2 Transfer the externally generated key pairs to `/data/vco/usr/lib/vco/app-server/conf` in vRealize Orchestrator.

Change the Key Pair Passphrase

You can run a workflow from the vRealize 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 Log in to the vRealize Orchestrator Client.
- 2 Navigate to **Library > Workflows** and enter the **ssh** tag in the workflow search box.
- 3 Locate the **Change key pair passphrase** workflow and click **Run**.
- 4 On the **Change passphrase** tab, reset the key pair passphrase.
 - a Enter the current passphrase.
 - b Enter the new passphrase.
- 5 Click **Run**.

Register a vRealize Orchestrator Public Key on an SSH Host

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

Prerequisites

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

Procedure

- 1 Log in to the vRealize Orchestrator Client.
- 2 Navigate to **Library > Workflows** and enter the **ssh** tag in the workflow search box.
- 3 Locate the **Register vCO public key on host** workflow and click **Run**.
- 4 On the **Register VS-O on Host** tab, 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 **Run**.

Results

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 vRealize 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 Log in to the vRealize Orchestrator Client.
- 2 Navigate to **Library > Workflows** and enter the **ssh** tag in the workflow search box.
- 3 Locate the **Run SSH command** workflow and click **Run**.
- 4 On the **Host selection** tab, enter an SSH host name or IP address.
- 5 On the **Command** tab, enter an SSH command to run.

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

- 6 On the **Encoding** tab, specify the encoding method.
Leave this field empty to use the default system encoding.
- 7 On the **Authentication** tab, enter a user name.
- 8 (Optional) Select the check box to use password authentication.

Note The default option is to use key file authentication.

- 9 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.
- 10 Click **Run**.

Copy a File from an SSH Host

You can run a workflow on the vRealize Orchestrator Client to copy files from an SSH host to the vRealize 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 vRealize Orchestrator must have explicit write permissions to write in folders.

Procedure

- 1 Log in to the vRealize Orchestrator Client.
- 2 Navigate to **Library > Workflows** and enter the **ssh** tag in the workflow search box.
- 3 Locate the **SCP get command** workflow and click **Run**.
- 4 On the **Host** tab, enter the source host information.
 - a Enter an SSH host name or IP address.
 - b Enter the SSH user name and password.
- 5 On the **File** tab, enter the file information.
 - a Enter the path to the file to get from the remote SSH host.
 - b Enter the path to the directory on the vRealize Orchestrator server into which to copy the file.
- 6 Click **Run**.

Copy a File to an SSH Host

You can run a workflow from the vRealize Orchestrator Client to copy files from the vRealize 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 Navigate to **Library > Workflows** and enter the **ssh** tag in the workflow search box.
- 2 Locate the **SCP put command** workflow and click **Run**.
- 3 On the **Host** tab, enter the source host information.
 - a Enter an SSH host name or IP address.
 - b Enter the SSH user name and password.
- 4 On the **File** tab, 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.
- 5 Click **Run**.

Using the vCenter Plug-In

19

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

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

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

Note Most vCenter plug-in workflows communicate only with the vCenter. However, some guest operations workflows require communication with the ESXi host managed by vCenter. Before you run these workflows, you must import the ESXi host certificate through the vRealize Orchestrator Control Center. For more information, see *Manage vRealize Orchestrator Certificates* in *Installing and Configuring VMware vRealize Orchestrator*.

The vCenter plug-in includes the Policy-Based Management (PBM) and the Storage Monitoring Service (SMS) APIs as scripting objects in the vRealize Orchestrator scripting API. The Storage Policy-Based Management policies and components appear in the **Inventory** page of the vRealize Orchestrator Client.

Read the following topics next:

- [Configuring the vCenter Server Plug-In](#)
- [vCenter Plug-In Scripting API](#)
- [Using the vCenter Plug-In Inventory](#)
- [Performance Considerations for Querying](#)
- [Using XPath Expressions with the vCenter Plug-In](#)
- [vCenter Plug-In Workflow Library](#)

Configuring the vCenter Server Plug-In

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

You can configure the vCenter plug-in by running the vCenter configuration workflows from the vRealize Orchestrator Client. See [Configuration Workflows](#)

Configure the Connection to a vCenter Instance

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

Procedure

- 1 Log in to the vRealize Orchestrator Client as an administrator.
- 2 Navigate to **Library > Workflows** and enter the **vcenter** and **configuration** tags in the workflow search box.
- 3 Locate the **Add a vCenter Server instance** workflow and click **Run**.
- 4 On the **Set the vCenter Server instance properties** tab, enter the IP address or the host name of the machine on which the vCenter instance you want to add is installed.

Note The host name that you enter is case-sensitive.

- 5 Retain the default port value, **443**.
- 6 Retain the default location of the SDK to use to connect to your vCenter instance.
- 7 Select whether you want to ignore certificate warnings for the vCenter instances that you want to add.

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

- 8 On the **Set the connection properties** tab, select whether you want to use a session per user method to manage user access on the vCenter system.

Option	Description
Share a unique session	<p>Select if vRealize Orchestrator does not use the same PSC as an authentication provider as the vCenter server instance that you want to orchestrate.</p> <p>Allows vRealize Orchestrator to create only one connection to vCenter.</p> <p>In the User name and Password text boxes, enter the credentials for vRealize Orchestrator to use to establish the connection to the vCenter host.</p> <p>The user that you select must be a valid user with privileges to manage vCenter extensions and a set of custom defined privileges. vRealize Orchestrator uses these credentials to monitor the VirtualCenter Web service, typically to run vRealize Orchestrator system workflows.</p>
Session per user	<p>Select if vRealize Orchestrator and your vCenter Server use the same PSC as an authentication provider.</p> <p>Creates a session to vCenter.</p>
<hr/> <p>Note This action might rapidly use CPU, memory, and bandwidth.</p> <hr/>	

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 inventory and cannot collect the necessary data.

- 9 (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 Session per user is selected.

- 10 (Optional) On the **Additional Endpoints** tab, enter the URLs for the vSphere storage management endpoints.

You can configure the Policy-Based Management (PBM) endpoint, the Storage Monitoring Service (SMS) endpoint, or both.

- a Enter the URL of PBM endpoint. The default vSphere 6.5 PBM endpoint URL is **https://vSphereHostName:443/pbm**.
- b Enter the URL of SMS endpoint. The default vSphere 6.5 SMS endpoint URL is **https://vSphereHostName:443/sms/sdk**.

- 11 Click **Run**.

Results

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

vCenter Plug-In Scripting API

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

For a list of API objects that are available to the vCenter plug-in, browse the API Explorer in the vRealize Orchestrator Client.

Using the vCenter Plug-In Inventory

The vCenter plug-in exposes all objects of the connected vCenter instances in the Inventory view.

To display the workflows that are available for a vCenter Server inventory object, navigate to **Administration > Inventory > vSphere vCenter Plug-in** in the vRealize Orchestrator Client.

What to read next

Performance Considerations for Querying

With the vCenter plug-in for vRealize Orchestrator, you can query the vCenter inventory for specific objects.

Querying Methods

For querying, you can either use the `vcSearchIndex` managed object, or the object finder methods that are included in the plug-in inventory, such as `getAllDatastores()`, `getAllVirtualMachines()`, `findAllForType()`, and others.

Performance

By default, both methods return the queried objects without including any of their properties, unless you specify a set of properties as an argument for the method parameters in the search query.

Note You must always use query expressions with the `getAll...` and `findAll...` finder objects to prevent the Orchestrator client from filtering large sets of returned objects, which might affect the overall performance of the Orchestrator server.

You can use two types of expressions for querying the vCenter inventory.

Type of Expression	Description
Name expressions	<p>You can specify a name as an argument for a query parameter.</p> <p>Note The objects are filtered by the specified name argument according to the name of the plug-in object as it is appears in the vCenter plug-in inventory.</p>
XPath expressions	<p>You can use expressions based on the XPath query language. For more information, see Using XPath Expressions with the vCenter Plug-In.</p>

When you invoke a vCenter inventory object with custom properties, each reference to this object, in a workflow or an action, sends a query to the vCenter, which generates a noticeable performance overhead. To optimize performance and avoid serializing and deserializing the object multiple times within a workflow run, it is best to use a shared resource to store the object, instead of storing it as a workflow attribute, an input, or an output parameter. Such shared resource can be a configuration element or a resource element.

Using XPath Expressions with the vCenter Plug-In

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

The vCenter 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 instances that are connected to your vRealize Orchestrator server and search for objects by ID, name, or other properties.

For performance reasons, the finder methods do not return any properties for the queried objects, unless you specify a set of properties in the search query.

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

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

Using XPath Expressions with the vCenter Plug-In Examples

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. If you want to query for any 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 managed objects and contain the string `ds` in their names.

```
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.

```
var datastores = Server.findAllForType("VC:Datastore", "xpath:name[contains(.,'ds')]");
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`.

```
var hosts = VcPlugin.getAllHostSystems(null, "xpath:id[starts-with(.,'1')]");
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. The script also retrieves the **tag** property.

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

vCenter Plug-In Workflow Library

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

- [Batch Workflows](#)

Batch workflows populate configuration elements or run workflows on a selected vCenter object.

- [Cluster and Compute Resource Workflows](#)

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

- [Configuration Workflows](#)

The Configuration workflow category of the vCenter plug-in contains workflows that let you manage the connections to vCenter 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 activate and deactivate 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 object.

To access these workflows, navigate to **Library > Workflows** and enter the **vcenter** and **batch** tags in the workflow search box.

Workflow Name	Description
Fill batch configuration elements	<p>Populates the configuration elements that the Run a workflow on a selection of objects workflow uses. Performs the following tasks:</p> <ul style="list-style-type: none"> ■ Resets the <code>BatchObject</code> and <code>BatchAction</code> configuration elements. ■ Fills the <code>BatchObject</code> configuration element with all the workflows that have only one input parameter. ■ Fills the <code>BatchAction</code> configuration element with all the actions that have no input parameters or one input parameter and that have an array as the <code>returnType</code>.
Run a workflow on a selection of objects	Runs a workflow on a selection of vCenter 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 activate or deactivate high availability, Distributed Resource Scheduler, and vCloud Distributed Storage on a cluster.

To access these workflows, navigate to **Library > Workflows** and enter the **vcenter** and **cluster_and_compute_resource** tags in the workflow search box.

Workflow Name	Description
Add DRS virtual machine group to cluster	Adds a DRS virtual machine group to a cluster.
Add virtual machines to DRS group	Adds a virtual machine list to an existing DRS virtual machine group.
Create cluster	Creates a cluster in a host folder.
Delete cluster	Deletes a cluster.
Disable DRS on cluster	Deactivates DRS on a cluster.
Disable HA on cluster	Deactivates high availability on a cluster.
Disable vCloud Distributed Storage on cluster	Deactivates vCloud Distributed Storage on a cluster.
Enable DRS on cluster	Activates DRS on a cluster.
Enable HA on cluster	Activates high availability on a cluster.
Enable vCloud Distributed Storage on cluster	Activates vCloud Distributed Storage on a cluster.
Remove virtual machine DRS group from cluster	Removes a DRS virtual machine group from a cluster.

Workflow Name	Description
Remove virtual machines from DRS group	Removes virtual machines from a cluster DRS group.
Rename cluster	Renames a cluster.

Configuration Workflows

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

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

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

Custom Attributes Workflows

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

To access these workflows, navigate to **Library > Workflows** and enter the **vcenter** and **custom_attributes** tags in the workflow search box.

Workflow Name	Description
Add custom attribute to a virtual machine	Adds a custom attribute to a virtual machine.
Add custom attribute to multiple virtual machines	Adds a custom attribute to a selection of virtual machines.
Get custom attribute	Gets a custom attribute for a virtual machine in vCenter.

Datacenter Workflows

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

To access these workflows, navigate to **Library > Workflows** and enter the **vcenter** and **datacenter** tags in the workflow search box.

Workflow Name	Description
Create datacenter	Creates a data center in a data center folder.
Delete datacenter	Deletes a data center.
Reload datacenter	Forces vCenter to reload data from a data center.
Rename datacenter	Renames a data center and waits for the task to complete.
Rescan datacenter HBAs	Scans the hosts in a data center and initiates a rescan on the host bus adapters to discover new storage.

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.

To access these workflows, navigate to **Library > Workflows** and enter the **vcenter** and **datastore_and_files** tags in the workflow search box.

Workflow Name	Description
Delete all files	Deletes a list of files.
Delete all unused datastore files	Searches all datastores in the vCenter environment and deletes all unused files.
Export unused datastore files	Searches all datastores and creates an XML descriptor file that lists all unused files.
Find unused files in datastores	Searches the vCenter environment for all unused disks (*.vmdk), virtual machines (*.vmx), and template (*.vmtx) files that are not associated with any vCenter instances registered with Orchestrator.
Get all configuration, template, and disk files from virtual machines	Creates a list of all virtual machine descriptor files and a list of all virtual machine disk files, for all datastores.
Log all datastore files	Creates a log for every virtual machine configuration file and every virtual machine file found in all datastores.
Log unused datastore files	Searches the vCenter environment for unused files that are registered on virtual machines and exports a log of the files in a text file.
Upload file to datastore	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.

Datacenter Folder Management Workflows

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

To access these workflows, navigate to **Library > Workflows** and enter the **vcenter** and **datacenter_folder** tags in the workflow search box.

Workflow Name	Description
Create datacenter folder	Creates a data center folder.
Delete datacenter folder	Deletes a data center folder and waits for the task to complete.
Rename datacenter folder	Renames a data center folder and waits for the task to complete.

Host Folder Management Workflows

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

To access these workflows, navigate to **Library > Workflows** and enter the **vcenter** and **host_folder** tags in the workflow search box.

Workflow Name	Description
Create host folder	Creates a host folder.
Delete host folder	Deletes a host folder and waits for the task to complete.
Rename host folder	Renames a host folder and waits for the task to complete.

Virtual Machine Folder Management Workflows

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

To access these workflows, navigate to **Library > Workflows** and enter the **vcenter** and **vm_folder** tags in the workflow search box.

Workflow Name	Description
Create virtual machine folder	Creates a virtual machine folder.
Delete virtual machine folder	Deletes a virtual machine folder and waits for the task to complete.
Rename virtual machine folder	Renames a virtual machine folder and waits for the task to complete.

Guest Operation Files Workflows

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

To access these workflows, navigate to **Library > Workflows** and enter the **vcenter**, **guest_operations** and **files** tags in the workflow search box.

Workflow Name	Description
Check for directory in guest	Verifies that a directory exists in a guest virtual machine.
Check for file in guest	Verifies that a file exists in a guest virtual machine.
Copy file from guest to Orchestrator	Copies a specified file from a guest file system to an Orchestrator server.
Copy file from Orchestrator to guest	Copies a specified file from an Orchestrator server to a guest file system.
Create directory in guest	Creates a directory in a guest virtual machine.
Create temporary directory in guest	Creates a temporary directory in a guest virtual machine.

Workflow Name	Description
Create temporary file in guest	Creates a temporary file in a guest virtual machine.
Delete directory in guest	Deletes a directory from a guest virtual machine.
Delete file in guest	Deletes a file from a guest virtual machine.
List path in guest	Shows a path in a guest virtual machine.
Move directory in guest	Moves a directory in a guest virtual machine.
Move file in guest	Moves a file in a guest virtual machine.

Guest Operation Processes Workflows

With guest operation processes workflows, you can get information and control the running processes in a guest operating system.

To access these workflows, navigate to **Library > Workflows** and enter the **vcenter**, **guest_operations** and **processes** tags in the workflow search box.

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

Power Host Management Workflows

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

To access these workflows, navigate to **Library > Workflows** and enter the **vcenter**, **host_management** and **power** tags in the workflow search box.

Workflow Name	Description
Reboot host	Reboots a host. If the Orchestrator client is connected directly to the host, it loses the connection to the host and does not receive an indication of success in the returned task.
Shut down host	Shuts down a host. If the Orchestrator client is connected directly to the host, it loses the connection to the host and does not receive an indication of success in the returned task.

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.

To access these workflows, navigate to **Library > Workflows** and enter the **vcenter**, **host_management** and **basic** tags in the workflow search box.

Workflow Name	Description
Enter maintenance mode	Puts the host into maintenance mode. You can cancel the task.
Exit maintenance mode	Exits maintenance mode. You can cancel the task.
Move host to cluster	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.
Move host to folder	Moves a host into a folder as a standalone host. The host must be part of a <code>ClusterComputeResource</code> in the same data center and the host must be in maintenance mode.
Reload host	Forces vCenter to 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.

To access these workflows, navigate to **Library > Workflows** and enter the **vcenter**, **host_management** and **registration** tags in the workflow search box.

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

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.

To access these workflows, navigate to **Library > Workflows** and enter the **vcenter** and **networking** tags in the workflow search box.

Workflow Name	Description
Add port group to distributed virtual switch	Adds a new distributed virtual port group to a specified distributed virtual switch.
Attach host system to distributed virtual switch	Adds a host to a distributed virtual switch.
Create distributed virtual switch with port group	Creates a new distributed virtual switch with a distributed virtual port group.

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.

To access these workflows, navigate to **Library > Workflows** and enter the **vcenter**, **networking** and **distributed_virtual_port_group** tags in the workflow search box.

Workflow Name	Description
Connect virtual machine NIC number to distributed virtual port group	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.
Delete distributed virtual port group	Deletes a specified distributed virtual port group.
Set teaming options	Provides an interface to manage the teaming options for a distributed virtual port group.
Update distributed virtual port group	Updates the configuration of a specified distributed virtual 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.

To access these workflows, navigate to **Library > Workflows** and enter the **vcenter**, **networking** and **distributed_virtual_switch** tags in the workflow search box.

Workflow Name	Description
Create distributed virtual switch	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.
Create private VLAN	Creates a VLAN on the specified distributed virtual switch.
Delete distributed virtual switch	Deletes a distributed virtual switch and all associated elements.
Delete private VLAN	Deletes a VLAN from a specified distributed virtual switch. If a secondary VLAN exists, you must first delete the secondary VLAN.
Update distributed virtual switch	Updates the properties of a distributed virtual switch.
Update private VLAN	Updates a VLAN on the specified distributed virtual switch.

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.

To access these workflows, navigate to **Library > Workflows** and enter the **vcenter**, **networking** and **standard_virtual_switch** tags in the workflow search box.

Workflow Name	Description
Add port group in standard virtual switch	Adds a port group in a standard virtual switch.
Create standard virtual switch	Creates a standard virtual switch.

Workflow Name	Description
Delete port group from standard virtual switch	Deletes a port group from a standard virtual switch
Delete standard virtual switch	Deletes a standard virtual switch from a host network configuration.
Retrieve all standard virtual switches	Retrieves all standard virtual switches from a host.
Update port group in standard virtual switch	Updates the properties of a port group in a standard virtual switch.
Update standard virtual switch	Updates the properties of a standard virtual switch.
Update VNIC for port group in standard virtual switch	Updates a virtual NIC associated with a port group in a standard virtual switch.

Networking Virtual SAN Workflows

With Virtual SAN workflows, you can configure Virtual SAN network traffic.

To access these workflows, navigate to **Library > Workflows** and enter the **vcenter**, **networking** and **vsan** tags in the workflow search box.

Workflow Name	Description
Set a cluster's VSAN traffic network	Sets a Virtual SAN traffic network of the cluster.
Set a host's VSAN traffic network	Sets a Virtual SAN traffic network of the host.

Resource Pool Workflows

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

To access these workflows, navigate to **Library > Workflows** and enter the **vcenter** and **resource_pool** tags in the workflow search box.

Workflow Name	Description
Create resource pool	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.
Create resource pool with specified values	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.
Delete resource pool	Deletes a resource pool and waits for the task to complete.
Get resource pool information	Returns CPU and memory information about a given resource pool.
Reconfigure resource pool	Reconfigures CPU and memory allocation configuration for a given resource pool.
Rename resource pool	Renames a resource pool and waits for the task to complete

Storage Workflows

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

To access these workflows, navigate to **Library > Workflows** and enter the **vcenter** and **storage** tags in the workflow search box.

Workflow Name	Description
Add datastore on iSCSI/FC/local SCSI	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.
Add datastore on NFS	Adds a datastore on an NFS server.
Add iSCSI target	Adds iSCSI targets to a vCenter host. The targets can be of the type <code>Send or Static</code> .
Create VMFS for all available disks	Creates a VMFS volume for all available disks of a specified host.
Delete datastore	Deletes datastores from a vCenter Server host.
Delete iSCSI target	Deletes already configured iSCSI targets. The targets can be of type <code>Send or Static</code> .
Disable iSCSI adapter	Deactivates the software iSCSI adapter of a specified host.
Display all datastores and disks	Displays the existing datastores and available disks on a specified host.
Enable iSCSI adapter	Activates an iSCSI adapter.
List all storage adapters	Lists all storage adapters of a specified host.

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.

To access these workflows, navigate to **Library > Workflows** and enter the **vcenter** and **storage_drs** tags in the workflow search box.

Workflow Name	Description
Add datastore to cluster	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.
Change Storage DRS per virtual machine configuration	Sets Storage DRS settings for each virtual machine.
Configure datastore cluster	Configures datastore cluster setting values for automation and runtime rules.
Create simple datastore cluster	Creates a simple datastore cluster with default configuration. The new datastore cluster contains no datastores.
Create Storage DRS scheduled task	Creates a scheduled task for reconfiguring a datastore cluster. Only automation and runtime rules can be set.
Create virtual machine anti-affinity rule	Creates an anti-affinity rule to indicate that all virtual disks of certain virtual machines must be kept on different datastores.
Create VMDK anti-affinity rule	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.
Remove datastore cluster	Removes a datastore cluster. Removing a datastore cluster also removes all the settings and the alarms for the cluster from the vCenter system.
Remove datastore from cluster	Removes a datastore from a datastore cluster and puts the datastore in a datastore folder.

Workflow Name	Description
Remove Storage DRS scheduled task	Removes a scheduled Storage DRS task.
Remove virtual machine anti-affinity rule	Removes a virtual machine anti-affinity rule for a given datastore cluster.
Remove VMDK anti-affinity rule	Removes a VMDK anti-affinity rule for a given datastore cluster.

Storage VSAN Workflows

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

To access these workflows, navigate to **Library > Workflows** and enter the **vcenter**, **storage** and **vsan** tags in the workflow search box.

Workflow Name	Description
Add disks to a disk group	Adds non-SSD disks to a Virtual SAN disk group.
Claim disks into disk groups	Claims disks for use by the Virtual SAN system and automatically creates disk groups and distributes the disks into existing disk groups.
Create a disk group	Creates a Virtual SAN disk group.
List hosts, disk groups and disks	Lists all hosts in a cluster, their disk groups and disks, used or eligible for use by the Virtual SAN system.
Remove disk groups	Removes Virtual SAN disk groups.
Remove disks from disk groups	Removes non-SSD disks from Virtual SAN disk groups.

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.

To access these workflows, navigate to **Library > Workflows** and enter the **vcenter**, **virtual_machine_management** and **basic** tags in the workflow search box.

Workflow Name	Description
Create custom virtual machine	Creates a virtual machine with the specified configuration options and additional devices.
Create simple dvPortGroup virtual machine	Creates a simple virtual machine. The network used is a Distributed Virtual Port Group.
Create simple virtual machine	Creates a virtual machine with the most common devices and configuration options.
Delete virtual machine	Removes a virtual machine from the inventory and datastore.
Get virtual machines by name	Returns a list of virtual machines from all registered vCenter instances that match the provided expression.
Mark as template	Converts an existing virtual machine to a template, not allowing it to start. You can use templates to create virtual machines.

Workflow Name	Description
Mark as virtual machine	Converts an existing template to a virtual machine, allowing it to start.
Move virtual machine to folder	Moves a virtual machine to a specified virtual machine folder.
Move virtual machine to resource pool	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.
Move virtual machines to folder	Moves several virtual machines to a specified virtual machine folder.
Move virtual machines to resource pool	Moves several virtual machines to a resource pool.
Register virtual machine	Registers a virtual machine. The virtual machine files must be placed in an existing datastore and must not be already registered.
Reload virtual machine	Forces vCenter to reload a virtual machine.
Rename virtual machine	Renames an existing virtual machine on the vCenter system or host and not on the datastore.
Set virtual machine performance	Changes performance settings such as shares, minimum and maximum values, shaping for network, and disk access of a virtual machine.
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 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.

To access these workflows, navigate to **Library > Workflows** and enter the **vcenter**, **virtual_machine_management** and **clone** tags in the workflow search box.

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.

To access these workflows, navigate to **Library > Workflows** and enter the **vcenter**, **virtual_machine_management** and **linked_clone** tags in the workflow search box.

Workflow Name	Description
Linked clone, Linux 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.
Linked clone, Linux with single NIC	Creates a linked clone of a Linux virtual machine, performs the guest operating system customization, and configures one virtual network card.
Linked clone, no customization	Creates the specified number of linked clones of a virtual machine.
Linked clone, Windows 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.
Linked clone, Windows with single NIC and credential	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.
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.

Linux Customization Clone Workflows

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

To access these workflows, navigate to **Library > Workflows** and enter the **vcenter**, **virtual_machine_management**, **clone** and **linux_customization** tags in the workflow search box.

Workflow Name	Description
Clone, Linux with multiple NICs	Clones a Linux virtual machine, performs the guest operating system customization, and configures up to four virtual network cards.
Clone, Linux with a single NIC	Clones a Linux virtual machine, performs the guest operating system customization, and configures one virtual network card.

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.

To access these workflows, navigate to **Library > Workflows** and enter the **vcenter**, **virtual_machine_management**, **clone** and **tools** tags in the workflow search box.

Workflow Name	Description
Get a VirtualEthernetCard to change the network	Returns a new ethernet card to update a virtual device. Contains only the device key of the given virtual device and the new network.
Get Linux customization	Returns the Linux customization preparation.

Workflow Name	Description
Get multiple VirtualEthernetVard device changes	Returns an array of <code>VirtualDeviceConfigSpec</code> objects for add and remove operations on <code>VirtualEthernetCard</code> objects.
Get NIC setting map	Returns the setting map for a virtual network card by using <code>VimAdapterMapping</code> . Changes NIC information for workflows that clone and reconfigure virtual machines. Other clone workflows call this workflow.
Get Windows customization, Sysprep with credentials	Returns customization information about the Microsoft Sysprep process, with credentials. Workflows for cloning Windows virtual machines use this workflow.
Get Windows customization, Sysprep with <code>Unattended.txt</code>	Returns customization information about the Microsoft Sysprep process by using an <code>Unattended.txt</code> file. Workflows for cloning Windows virtual machines use this workflow.
Get Windows customizations for Sysprep	Returns customization information about the Microsoft Sysprep process. Workflows for cloning Windows virtual machines use this workflow.

Windows Customization Clone Workflows

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

To access these workflows, navigate to **Library > Workflows** and enter the **vcenter**, **virtual_machine_management**, **clone** and **windows_customization** tags in the workflow search box.

Workflow Name	Description
Clone thin provisioned, Windows with single NIC and credential	Clones a Windows virtual machine performing the guest operating system customization. Specifies virtual disk thin provisioning policy and configures one network card and a local administrator user account. Sysprep tools must be available on vCenter system.
Clone, Windows Sysprep with single NIC and credential	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 system.
Clone, Windows with multiple NICs and credential	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 system.
Clone, Windows with single NIC	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 system.
Clone, Windows with single NIC and credential	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 system.
Customize, Windows with single NIC and credential	Performs guest operating system customization, configures one virtual network card and a local administrator user account on a Windows virtual machine.

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.

To access these workflows, navigate to **Library > Workflows** and enter the **vcenter**, **virtual_machine_management** and **device_management** tags in the workflow search box.

Workflow Name	Description
Add CD-ROM	Adds a virtual CD-ROM to a virtual machine. If the virtual machine has no IDE controller, the workflow creates one.
Add disk	Adds a virtual disk to a virtual machine.
Change RAM	Changes the amount of RAM of a virtual machine.
Convert disks to thin provisioning	Converts thick-provisioned disks of virtual machines to thin-provisioned disks.
Convert independent disks	Converts all independent virtual machine disks to normal disks by removing the independent flag from the disks.
Disconnect all detachable devices from a running virtual machine	Disconnects floppy disks, CD-ROM drives, parallel ports, and serial ports from a running virtual machine.
Mount CD-ROM	Mounts the CD-ROM of a virtual machine. If the virtual machine has no IDE controller and/or CD-ROM drive, the workflow creates them.
Mount floppy disk drive	Mounts a floppy disk drive FLP file from the ESX datastore.

Move and Migrate Workflows

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

To access these workflows, navigate to **Library > Workflows** and enter the **vcenter**, **virtual_machine_management** and **move_and_migrate** tags in the workflow search box.

Workflow Name	Description
Mass migrate virtual machines with storage vMotion	Uses Storage vMotion to migrate a single virtual machine, a selection of virtual machines, or all available virtual machines.
Mass migrate virtual machines with vMotion	<p>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.</p> <p>Note vCenter Server does not allow storage vMotion and vMotion in the same pass for a powered on virtual machine. You must power off the virtual machine to use storage vMotion and vMotion in the same pass.</p>
Migrate virtual machine with vMotion	Migrates a virtual machine from one host to another by using the <code>MigrateVM_Task</code> operation from the vSphere API.
Move virtual machines to another vCenter Server	Moves a list of virtual machines to another vCenter system.
Quick migrate multiple virtual machines	Suspends the virtual machines if they are powered on and migrates them to another host using the same storage.

Workflow Name	Description
Quick migration of virtual machine	Suspends the virtual machine if it is powered on and migrates it to another host using the same storage.
Relocate virtual machine disks	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.

Other Workflows

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

To access these workflows, navigate to **Library > Workflows** and enter the **vcenter**, **virtual_machine_management** and **others** tags in the workflow search box.

Workflow Name	Description
Disable FT	Deactivates Fault Tolerance for a specified virtual machine.
Enable FT	Activates Fault Tolerance for a specified virtual machine.
Extract virtual machine information	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.
Find orphaned virtual machines	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).
Get VM 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 <code>VC:VirtualMachine</code> type in order to make the correspondence between a particular DynamicOps and vRealize Orchestrator virtual machine.
Get VM 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 <code>VC:VirtualMachine</code> type in order to make the correspondence between a particular DynamicOps and vRealize Orchestrator virtual machine.
Get VM 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 <code>VC:VirtualMachine</code> 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.

To access these workflows, navigate to **Library > Workflows** and enter the **vcenter**, **virtual_machine_management** and **power_management** tags in the workflow search box.

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

Snapshot Workflows

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

To access these workflows, navigate to **Library > Workflows** and enter the **vcenter**, **virtual_machine_management** and **snapshot** tags in the workflow search box.

Workflow Name	Description
Create a snapshot	Creates a snapshot.
Create snapshots of all virtual machines in a resource pool	Creates a snapshot of each virtual machine in a resource pool.
Remove all snapshots	Removes all existing snapshots without reverting to a previous snapshot.
Remove excess snapshots	Finds virtual machines with more than a given number of snapshots and optionally deletes the oldest snapshots. Sends the results by email.
Remove old snapshots	Gets all snapshots that are older than a given number of days and prompts the user to select which ones to delete.
Remove snapshots of a given size	Gets all snapshots that are larger than a given size and prompts the user to confirm deletion.
Revert to current snapshot	Reverts to the current snapshot.
Revert to snapshot and wait	Reverts to a specific snapshot. Does not delete the snapshot.

VMware Tools Workflows

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

To access these workflows, navigate to **Library > Workflows** and enter the **vcenter**, **virtual_machine_management** and **vmware_tools** tags in the workflow search box.

Workflow Name	Description
Mount tools installer	Mounts the VMware Tools installer on the virtual CD-ROM.
Set console screen resolution	Sets the resolution of the console window. The virtual machine must be powered on.
Turn on time synchronization	Turns on time synchronization between the virtual machine and the ESX server in VMware Tools.
Unmount tools installer	Unmounts the VMware Tools CD-ROM.
Update tools on Windows virtual machine without rebooting	Updates VMware Tools on a Windows virtual machine without performing a reboot.
Upgrade tools	Upgrades VMware Tools on a virtual machine.
Upgrade tools at next reboot	Deprecated: use the workflow Update tools on Windows virtual machine without rebooting

Using the vRealize Orchestrator Plug-In for vSphere Web Client

20

You can use the vRealize Orchestrator plug-in for vSphere Web Client (also known as the vCOIN plug-in) to perform operations on certain vRealize Orchestrator content from your vSphere Client.

The content that the plug-in exposes in the vSphere Client includes workflows, workflow runs, scheduled workflows, and workflows waiting for input. In the vSphere Client, users can see workflow details, run or schedule workflows, and manage workflow runs.

Context Actions

You can add vRealize Orchestrator workflows as context actions in the vSphere Web Client. With context actions, you can map a particular workflow to a specific vSphere item and run or schedule the workflow directly from the vSphere inventory.

When you use the vCOIN plug-in for the first time, a configuration element is created in vRealize Orchestrator. You must not change or edit the location and any configuration items in that location because they are managed by the vCOIN service.

Group Assignments

Administrators can use group assignments to assign context actions to specific vRealize Orchestrator groups. Users in these groups can then use context actions in vSphere. Users can run only the workflows that are assigned to their group.

When a user logs into the vSphere Client, they have the same views and/or permissions in the vCOIN plug-in as in vRealize Orchestrator. For example, if a user has the Viewer role in vRealize Orchestrator, they can't run workflows in the vSphere Client.

To use group assignments, you must have at least one group defined in vRealize Orchestrator. If you want to leverage specific user roles, such as **Designer** or **Administrator**, your vRealize Orchestrator environment must use a vRealize Automation license.

When you assign context actions to a group, all of the mapped workflows are added as group items in the assigned group, along with the configuration item.

Functional Requirements for the vCOIN Plug-In

The vRealize Orchestrator plug-in for vSphere Web Client supports the following:

- Standalone vRealize Orchestrator 8.11.x environments only. The plug-in is not supported for instances of vRealize Orchestrator that are embedded in vRealize Automation.
- The vCOIN plug-in is available only with vRealize Orchestrator. To upgrade the plug-in, you must upgrade vRealize Orchestrator.
- vSphere 7.0 U1 and later. Versions 7.0 and earlier are not supported.
- Maximum one vCOIN can be registered per vCenter Server.
- In vSphere deployments with multiple vCenters (in Enhanced Linked Mode), you can register a single vCOIN plug-in with multiple vCenters (maximum one plug-in installation per vCenter), or you can register multiple vCOINs with multiple vCenter servers.
- Before you can use the plug-in, you must configure the plug-in by enabling the service on the vRealize Orchestrator appliance. See [Configuring the vRealize Orchestrator Plug-in for vSphere Web Client](#).

Read the following topics next:

- [Configuring the vRealize Orchestrator Plug-in for vSphere Web Client](#)
- [Create a Context Action](#)

Configuring the vRealize Orchestrator Plug-in for vSphere Web Client

Configure the vCOIN plug-in.

Prerequisites

Verify that your vRealize Orchestrator authenticates with vSphere. The Orchestrator authentication provider must use the vCenter FQDN as the target vCenter instance FQDN. See [Configure a Standalone vRealize Orchestrator Server with vSphere Authentication](#).

Procedure

1 Activate the vCOIN plug-in.

- a Log in to the vRealize Orchestrator Appliance over SSH.
- b Run the following command.

```
vracli capabilities vcoin --enable
```

You can also add the following optional parameters after the `capabilities` command.

Parameter	Description
<code>-h, --help</code>	Get information about optional commands that you can use.
<code>--enable</code>	Activate this capability.
<code>--disable</code>	Deactivate this capability.
<code>--replay</code>	Ignore the current state and potentially replay any state validations.
<code>-f, --force</code>	Force set a value for this capability.

- c Redeploy the vRealize Orchestrator appliance by running the `/opt/scripts/deploy.sh` command.

Running this command restarts vRealize Orchestrator.

2 Register the vCOIN extension in the vSphere Web Client.

```
vracli vro vcoin register --insecure -u username -vcu https://vSphere FQDN/sdk
```

Provide your password when the system prompts you to.

You can also add the following parameters after the `vro vcoin register` command.

Parameter	Description
<code>--insecure</code>	Optional. If you do not use <code>--insecure</code> parameter, the vCenter certificate thumbprint must be passed with the <code>-vct</code> parameter.
<code>-u, --username=username</code>	The user name of a user with permissions to register extensions on the vCenter server. For example, administrator users usually have the necessary Extension.Register permission.
<code>-vcu, --vCenterUrl=https://vSphere FQDN/sdk</code>	The URL of the SDK resource of the vCenter Server.
<code>-st, --serverThumbprint=vRO Server Certificate Thumbprint</code>	Optional. The certificate thumbprint of the vRealize Orchestrator plug-in server. The SHA-1 thumbprint must be in the following format: <code>30:da:0s:da:is:d9:sa:91:29:da:9s:ds:a9:d9:sa</code> . Make sure that every number-letter pair combination is separated by colons (:).

Parameter	Description
<code>-vct, --vCenterServerThumbprint=vCenter Server Thumbprint</code>	The certificate thumbprint of the vCenter server.
<code>-pu, --pluginUrl=https://vRO or LB FQDN/vcoin/plugin.json</code>	The URL of the vCOIN plug-in manifest. If you have a high availability environment with a proxy or a load balancer, you must use the HA proxy FQDN or the LB FQDN instead of the vRealize Orchestrator FQDN. Otherwise, the plug-in cannot be registered.

After the vCOIN plug-in extension is registered, verify that the plug-in is visible in the vSphere Client. You might have to refresh your browser. For troubleshooting information, see the VMware vSphere documentation.

- 3 To activate autofilling for context action inputs, add the vCenter Server as an endpoint in vRealize Orchestrator by running the **Add a vCenter Server Instance** workflow.

For instructions how to run the workflow, see [Configure the Connection to a vCenter Instance](#).

Unregister the vCOIN Plug-In

To unregister the vCOIN extension from the vSphere Web Client, run the following command.

```
vracli vro vcoin unregister --insecure -u username -vcu https://vSphere FQDN/sdk
```

If the vCenter server is unavailable, you can add the `--force` after the main command.

If you do not unregister the plug-in properly when reconfiguring your environment, unexpected behaviour might occur if your vRealize Orchestrator instance is registered with a different vCenter server.

Create a Context Action

The vSphere Client provides an inventory of vSphere items that are available for management. You can define context actions for specific vSphere inventory objects and run vRealize Orchestrator workflows on those objects from the vSphere inventory.

Procedure

- 1 Log in to the vSphere Web Client.
- 2 On the vRealize Orchestrator home page, click the **Context Actions** tab.
- 3 Click **Add**.
- 4 In the search box, enter the name of the workflow that you want to add as a context action.
- 5 Select the workflow in the results table.
- 6 In the **Type of object** dropdown menu, select which vSphere object you want to associate with the context action, such as a vCenter server, for example.

7 Click **Assign**.

You added a vRealize Orchestrator workflow as a context action.

8 Run the workflow on the vCenter server that you associated with the action.

- a In the vSphere Client, right click the vCenter server.
- b Select **vRealize Orchestrator > Run Workflow**.
- c Select the context action workflow to run on the vCenter server.

The workflow input form appears. If the workflow has as an input parameter of the particular inventory type over which you initiated the context action, the form is prepopulated.

- d Click **Run**.

Using the vCloud Suite API (vAPI) Plug-In

21

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*.

Read the following topics next:

- [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 vRealize Orchestrator Client as an administrator.
- 2 Navigate to **Library > Workflows** and enter the **vapi** tag in the workflow search box.
- 3 Locate the **Import vAPI metamodel** workflow and click **Run**.
- 4 In the **vAPI endpoint URL** text box, enter the URL of your vCloud Suite API endpoint.

```
https://vCloud_Suite_API_provider_FQDN/api
```

- 5 Select whether to use a secure protocol connection.

Option	Description
No	Import the vCloud Suite API metamodel, without using a secure protocol connection.
Yes	To import the vCloud Suite API metamodel with secure protocol connection: <ol style="list-style-type: none"> Select whether to ignore certificate warnings and accept the vCloud Suite endpoint automatically. Provide the user credentials to authenticate with the vCloud Suite endpoint.

- 6 Select whether to add a vAPI endpoint using the same credentials.

- 7 Click **Run**.

What to do next

[Add a vCloud Suite API Endpoint](#)

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 vRealize Orchestrator Client as an administrator.
- 2 Navigate to **Library > Workflows** and enter the **vapi** tag in the workflow search box.
- 3 Locate the **Add vAPI endpoint** workflow and click **Run**.
- 4 In the **vAPI endpoint URL** text box, enter the URL of your vCloud Suite API endpoint.

```
https://vCloud_Suite_API_provider_FQDN/api
```

- 5 Choose whether to use a secure protocol connection:

Option	Description
No	Import the vCloud Suite API metamodel, without using a secure protocol connection.
Yes	To import the vCloud Suite API metamodel with secure protocol connection: <ol style="list-style-type: none"> Choose whether to ignore certificate warnings and accept the vCloud Suite endpoint automatically. Provide the user credentials to authenticate with the vCloud Suite endpoint.

- 6 Click **Run**.

Access the vCloud Suite API Plug-In API

vRealize 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 vRealize Orchestrator Client as an administrator.
- 2 Access the API Explorer from either the vRealize Orchestrator Client or from the **Scripting** tabs of the workflow, policy, and action editors.
 - To access the API Explorer from the vRealize Orchestrator Client, click **API Explorer** in the vRealize Orchestrator Client navigation pane.
 - 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*.

Using the vRealize Orchestrator Plug-In for vRealize Automation

22

The VMware vRealize® Orchestrator™ Plug-in for vRealize Automation™ integrates vRealize Orchestrator with vRealize Automation and vRealize Automation Cloud. With the plug-in, you can run vRealize Orchestrator workflows from your vRealize Automation or vRealize Automation Cloud instance. You can use the workflows that are provided with the plug-in to deploy and manage vRealize Automation and vRealize Automation Cloud resources.

Role of vRealize Orchestrator with the vRealize Automation plug-in

vRealize Orchestrator powers the vRealize Automation plug-in. You use the vRealize Orchestrator Client to run and create workflows and access the vRealize Automation plug-in API.

You can use either the embedded vRealize Orchestrator instance in your vRealize Automation deployment, or an external vRealize Orchestrator server.

Installing the vRealize Automation plug-in

Depending on your vRealize Orchestrator setup, you must either download and install the vRealize Automation plug-in yourself, or the plug-in might come preinstalled on your vRealize Automation environment.

The following table provides more information about each scenario.

vRealize Orchestrator deployment	vRealize Automation version	Out of the box plug-in availability	What to do
Embedded	vRealize Automation 8.11.x	Yes	<ol style="list-style-type: none"> 1 Configuring vRealize Automation and vRealize Automation Cloud Hosts. 2 Start Using the vRealize Automation Plug-In Infrastructure Administration Workflows.
External	vRealize Automation 8.11.x	No	<ol style="list-style-type: none"> 1 Download the plug-in from the VMware Marketplace. 2 Install or Update a vRealize Orchestrator Plug-In on your vRealize Orchestrator instance.
Cloud extensibility proxy	vRealize Automation Cloud	No	<ol style="list-style-type: none"> 1 Verify that you have a vRealize Orchestrator integration in Cloud Assembly. See Configure a vRealize Orchestrator integration in Cloud Assembly. 2 Download the plug-in from the VMware Marketplace. 3 Install or Update a vRealize Orchestrator Plug-In on your cloud-enabled vRealize Orchestrator.

The vRealize Automation plug-in supports out-of-the-box proxy-based connection configurations on the vRealize Orchestrator/vRealize Automation Cloud extensibility appliance. You can connect an external proxy with the vRealize Automation host connection object without any additional configuration changes.

Using the default vRealize Automation plug-in workflows and actions

The vRealize Automation plug-in provides out-of-the-box workflows for common tasks, such as [Configuring vRealize Automation and vRealize Automation Cloud Hosts](#) and [Using the vRealize Automation Plug-In Infrastructure Administration Workflows](#). For a full list of available workflows, navigate to **Library > Workflows > vRealize Automation 8.x and Cloud Services** in the vRealize Orchestrator Client.

The plug-in library also contains predefined actions that you can use to build your own custom workflows. To access these actions, navigate to **Library > Actions**, and enter `com.vmware.library.vra` in the action search box.

Using the vRealize Automation plug-in inventory

The vRealize Orchestrator inventory supports objects for hosts, cloud accounts, cloud zones, disks, machines, machine disks and snapshots, networks, projects, and other entities that are required as lookups for create/update workflows, such as tags, data collectors, regions, NSX-T and NSX-V cloud accounts.

To display all available inventory objects, navigate to **Administration > Inventory > vRealize Automation and Cloud Services** in the vRealize Orchestrator Client.

Accessing the vRealize Automation plug-in API

In the vRealize Orchestrator API Explorer, you can search the vRealize Automation plug-in API and see the documentation for JavaScript objects that you can use in scripted elements. You can copy code from API elements and paste it into scripting boxes.

In the vRealize Orchestrator API Explorer, click the **VRA** module in the left pane to expand the hierarchical list of vRealize Automation plug-in API scripting objects.

To access the API reference for your vRealize Automation version, go to `https://vra-hostname/automation-ui/api-docs`.

For up-to-date vRealize Automation API documentation, see the [vRealize Automation 8.11 API Programming Guide](#).

For vRealize Orchestrator sample workflows that leverage the vRealize Automation plug-in REST client for extensibility migration, see [vRealize Automation 8.7 Extensibility Migration Guide Samples](#).

Read the following topics next:

- [Configuring vRealize Automation and vRealize Automation Cloud Hosts](#)
- [Using the vRealize Automation Plug-In Infrastructure Administration Workflows](#)

Configuring vRealize Automation and vRealize Automation Cloud Hosts

You can manage vRealize Automation and vRealize Automation Cloud hosts by running the default workflows provided with the plug-in.

To access these workflows in the vRealize Orchestrator Client, navigate to **Library > Workflows**, and enter the **vra** or **vra-cloud** tag in the workflow search box.

Workflow	Description
Add vRA Host	<p>Adds a vRealize Automation host to the plug-in inventory. See Add a vRealize Automation Host</p> <hr/> <p>Note If you use the plug-in in an embedded vRealize Orchestrator instance, a default host is created in the plug-in inventory. The logged-in user session is used to execute operations on the default host.</p> <p>When you perform operations on the default host from vRealize Automation Service Broker, the vRealize Orchestrator Gateway service token is used to execute the operation.</p>
Add vRA Cloud Host	<p>Adds a vRealize Automation Cloud host to the plug-in inventory. When running this workflow, you must select only a relevant region-specific cloud host. See Add a vRealize Automation Cloud Host host.</p>

Workflow	Description
Update vRA Host Update vRA Cloud Host	Updates a vRealize Automation or vRealize Automation Cloud host in the plug-in inventory.
Remove Host	Removes a vRealize Automation or vRealize Automation Cloud host from the plug-in inventory.
Validate Host	Validates the configuration of the vRealize Automation or vRealize Automation Cloud host.

Multi-tenancy support

You can configure the vRealize Automation plug-in to work with your multi-tenant environment.

- For on-premises connections, you must add a dedicated vRealize Automation host for each tenant, using the tenant FQDN as the host name.

If you use the plug-in with a vRealize Orchestrator instance that is embedded in vRealize Automation, the default connection does not use a tenant-specific URL. Instead, it uses the default vRealize Automation host name.

- For cloud connections, the plug-in uses the API token to differentiate between the tenants.

Invoking REST operations on hosts

The vRealize Orchestrator plug-in for vRealize Automation supports generic REST operations on dynamically created hosts. You can run the default plug-in workflows to invoke any public vRealize Automation and vRealize Automation Cloud APIs.

The plug-in supports the following REST operations.

Workflow	Description
Get operation	Generic REST client support for HTTP GET operation.
Put operation	Generic REST client support for HTTP PUT operation.
Post operation	Generic REST client support for HTTP POST operation.
Patch operation	Generic REST client support for HTTP PATCH operation.
Delete operation	Generic REST client support for HTTP DELETE operation.

Using the default vRealize Automation host in the plug-in inventory

If you use the default vRealize Automation host that is created in the plug-in inventory in an embedded vRealize Orchestrator instance, the logged-in user session is used by default to execute operations on the host.

In versions 8.10.1 and earlier, when you run operations on the default host from vRealize Automation Service Broker, the vRealize Orchestrator Gateway service token is used to execute the operation.

In versions 8.10.2 and later, the operations are run with the logged-in user token roles and permissions. If you want to keep the same behavior as in < 10.2, you can update the default host to use a Shared session with providing a user that has the same roles & permissions as the vRealize Orchestrator Gateway service user. If users need to keep the same behaviour they can create a new user and assign the required roles and permissions as vro-gateway user and start using “Shared Session”.

For users who require their workflow/actions to use the roles and permissions of logged-in user, they can still use “Per User Session” to execute their vRA APIs.

Add a vRealize Automation Host

You add a vRealize Automation host and configure the connection parameters by running a vRealize Orchestrator workflow.

Procedure

- 1 Log in to the vRealize Orchestrator Client as an administrator.
- 2 Navigate to **Library > Workflows** and enter the **vra** and **configuration** tags in the workflow search box.
- 3 Locate the **Add vRA host** workflow and click **Run**.
- 4 Enter a unique name for the host in the **vRA Host Name** text box.
- 5 Enter the URL address of the host in the **vRA Host URL** text box.
For example: `https://vra-hostname`.
- 6 Select whether to install the SSL certificates automatically without user confirmation.
- 7 On the **User credentials** tab, select the type of connection to the host.

Option	Actions
Shared Session	Connect using the credentials for a vRealize Automation user that you provide in the Authentication User Name and Authentication Password text boxes.
Per User Session	Connect using the credentials of the user that is currently logged in. You must be logged in to the vRealize Orchestrator Client with the credentials of the vRealize Automation administrator.

- 8 Click **Run**.

Results

You have added a vRealize Automation host.

Add a vRealize Automation Cloud Host

You add a vRealize Automation Cloud host and configure the connection parameters by running a vRealize Orchestrator workflow.

Prerequisites

- Verify that you have a vRealize Orchestrator integration in Cloud Assembly. See [Configure a vRealize Orchestrator integration in Cloud Assembly](#).
- Verify that you have an API access token. See [Generate API Tokens](#).

Procedure

- 1 Log in to the vRealize Orchestrator Client as an administrator.
- 2 Navigate to **Library > Workflows** and enter the **vra-cloud** and **configuration** tags in the workflow search box.
- 3 Locate **Add vRA Cloud Host** workflow and click **Run**.
- 4 Enter a unique name for the host in the **vRA Cloud Host Name** text box.
- 5 Select the URL address of the vRealize Automation Cloud host that contains the cloud region for which you want to configure the host.
- 6 Select whether to install the SSL certificates automatically without user confirmation.
- 7 On the **User credentials** tab, provide the API access token.
- 8 Click **Run**.

Results

You have added a vRealize Automation Cloud host.

Using the vRealize Automation Plug-In Infrastructure Administration Workflows

You can use the infrastructure administration workflows to manage cloud accounts, cloud zones, machines, and projects.

Cloud Accounts Workflows

The Cloud Accounts category contains workflows that you can use for managing vSphere cloud accounts.

To access these workflows in the vRealize Orchestrator Client, navigate to **Library > Workflows**, and enter the **vsphere_cloud_accounts** tag in the workflow search box.

Workflow Name	Description
Create vSphere Cloud Account	Adds a vCenter cloud account in vRealize Automation or vRealize Automation Cloud and to the plug-in inventory in vRealize Orchestrator. See Add a vSphere Cloud Account .
Create vSphere Cloud Account Async	Creates a cloud account asynchronously with the selected configuration. On request submission, the workflow returns a <code>RequestTracker</code> object. You can get the status of the requested cloud account by using the <code>RequestService</code> and <code>RequestTracker</code> objects and the <code>getRequestTrackerById</code> action.
Delete vSphere Cloud Account	Removes a vCenter cloud account in vRealize Automation or vRealize Automation Cloud and from the plug-in inventory in vRealize Orchestrator.
Update vSphere Cloud Account	Updates a vCenter cloud account in vRealize Automation or vRealize Automation Cloud and in the plug-in inventory in vRealize Orchestrator.
Update vSphere Cloud Account Async	Requests to update a cloud account asynchronously with the selected configuration. On request submission, the workflow returns a <code>RequestTracker</code> object. You can get the status of the requested cloud account by using the <code>RequestService</code> and <code>RequestTracker</code> objects and the <code>getRequestTrackerById</code> action.

Cloud Zones Workflows

To access the Cloud Zones workflows in the vRealize Orchestrator Client, navigate to **Library > Workflows**, and enter the **ccloud_zones** tag in the workflow search box.

Workflow Name	Description
Create Cloud Zone	Adds a cloud zone in vRealize Automation or vRealize Automation Cloud and to the plug-in inventory in vRealize Orchestrator. See Add a Cloud Zone .
Delete Cloud Zone	Removes a cloud zone in vRealize Automation or vRealize Automation Cloud and from the plug-in inventory in vRealize Orchestrator.
Update Cloud Zone	Updates a cloud zone in vRealize Automation or vRealize Automation Cloud and in the plug-in inventory in vRealize Orchestrator.

Disks Workflows

To access the Disks workflows in the vRealize Orchestrator Client, navigate to **Library > Workflows**, and enter the **disks** tag in the workflow search box.

Workflow Name	Description
Create Disk	Creates a disk (block device) synchronously and adds it to the plug-in inventory in vRealize Orchestrator. See Add a Disk .
Delete Disk	Requests to delete a disk (block device) with the selected configuration asynchronously. On request submission, the workflow returns a <code>RequestTracker</code> object. You can get the status of the requested machine disk by using the <code>RequestService</code> and <code>RequestTracker</code> objects.

In addition to the default workflows, the vRealize Automation plug-in supports the Promote disk and Resize block device operations.

Machines Workflows

To access the Machines workflows in the vRealize Orchestrator Client, navigate to **Library > Workflows**, and enter the **machines** tag in the workflow search box.

Workflow Name	Description
Create Machine	Creates a virtual machine in vRealize Automation or vRealize Automation Cloud and adds it to the plug-in inventory in vRealize Orchestrator. See Add a Machine .
Create Machine Async	Asynchronously creates a virtual machine with the selected configuration. On request submission, the workflow returns a <code>RequestTracker</code> object. You can get the status of the requested machine by using the <code>RequestService</code> and <code>RequestTracker</code> objects and the <code>getRequestTrackerById</code> action.
Delete Machine	Removes a virtual machine from vRealize Automation or vRealize Automation Cloud and from the plug-in inventory in vRealize Orchestrator. On request submission, the workflow returns a <code>RequestTracker</code> object. You can get the status of the requested machine by using the <code>RequestService</code> and <code>RequestTracker</code> objects and the <code>getRequestTrackerById</code> action.
Resize Machine	Resizes a vRealize Automation machine asynchronously. On request submission, the workflow returns a <code>RequestTracker</code> object. You can get the status of the requested machine by using the <code>RequestService</code> and <code>RequestTracker</code> objects. Note that the Resize Machine workflow does not validate the maximum value for the CPU Count, Core Count, and Memory inputs. The vRealize Orchestrator API does not support fetching the maximum configured values for these machine attributes.

Workflow Name	Description
Update Machine Custom Properties	Updates virtual machine custom properties.
Update Machine Tags	Updates virtual machine tags.

In addition to the default workflows, the vRealize Automation plug-in supports various machine power operations, including Power On/Off, Reset, Reboot, Resize, Shutdown.

To access these actions, navigate to **Library > Actions**, and search for the **com.vmware.library.vra.infrastructure.machine.power** tag in the action search box.

Note that the power operation action fails if you run an unsupported operation or if the target criteria is not met.

Machine Disks Workflows

To access the Machine Disks workflows in the vRealize Orchestrator Client, navigate to **Library > Workflows**, and enter the **machines** and **disks** tags in the workflow search box.

Workflow Name	Description
Attach Machine Disk	Attaches a vRealize Automation machine disk (block device) asynchronously.
Detach Machine Disk	Detaches a vRealize Automation machine disk (block device) asynchronously.

On request submission, the Machine Disks workflows return a `RequestTracker` object. You can get the status of the requested machine disks by using the `RequestService` and `RequestTracker` objects.

Machine Snapshots Workflows

To access the Machine Snapshots workflows in the vRealize Orchestrator Client, navigate to **Library > Workflows**, and enter the **machines** and **snapshots** tags in the workflow search box.

Workflow Name	Description
Create Machine Snapshot	Requests to create a machine snapshot with the selected configuration asynchronously.
Delete Machine Snapshot	Requests to delete a machine snapshot with the selected configuration asynchronously.
Revert Machine Snapshot	Reverts a vRealize Automation machine snapshot asynchronously.

On request submission, the Machine Snapshots workflows return a `RequestTracker` object. You can get the status of the requested machine snapshots by using the `RequestService` and `RequestTracker` objects.

Networks Workflows

To access the Networks workflows in the vRealize Orchestrator Client, navigate to **Library > Workflows**, and enter the **networks** tag in the workflow search box.

Workflow Name	Description
Create Network	Creates a network asynchronously. See Add a Network .
Delete Network	Requests to delete a network with the selected configuration asynchronously. On request submission, the workflow returns a <code>RequestTracker</code> object. You can get the status of the requested network operation by using the <code>RequestService</code> and <code>RequestTracker</code> objects.

Projects Workflows

To access the Projects workflows in the vRealize Orchestrator Client, navigate to **Library > Workflows**, and enter the **projects** tag in the workflow search box.

Workflow Name	Description
Create Project	Adds a project in vRealize Automation or vRealize Automation Cloud and to the plug-in inventory in vRealize Orchestrator. See Add a Project .
Delete Project	Removes a project in vRealize Automation or vRealize Automation Cloud and from the plug-in inventory in vRealize Orchestrator.
Update Project	Updates a project in vRealize Automation or vRealize Automation Cloud and in the plug-in inventory in vRealize Orchestrator.
Update Project Resource Metadata	Updates the resource metadata, such as tags, that are associated with a project.

Add a vSphere Cloud Account

You add a vSphere cloud account and configure its parameters by running a vRealize Orchestrator workflow.

Prerequisites

- To configure and work with cloud accounts in vRealize Automation, verify that you have the necessary credentials. See [Credentials required for working with cloud accounts in vRealize Automation](#).
- For information about creating vSphere cloud accounts, see [Create a vCenter cloud account in vRealize Automation](#).

Procedure

- 1 Log in to the vRealize Orchestrator Client as an administrator.

- 2 Navigate to **Library > Workflows** and enter the **cloud_accounts** tag in the workflow search box.
- 3 Locate **Create vSphere Cloud Account** workflow and click **Run**.
- 4 Select a vRealize Automation host for which you want to configure a vSphere Cloud account.
- 5 Enter a name for the cloud account.
- 6 Configure the **vCenter Server** credentials.
 - a Enter the IP address or fully-qualified domain name of your vCenter server.
 - b Select whether to accept self-signed certificated automatically without user confirmation.
 - c Provide the vCenter server user name and password.
- 7 If you want to add tags to support a tagging strategy, add capability tags.
 You can add tags now, or later when you edit the cloud account. For information about tagging, see [How do I use tags to manage vRealize Automation Cloud Assembly resources and deployments](#).
- 8 Click **Run**.

Results

You have added a vSphere cloud account.

What to do next

Configure a cloud zone within the cloud account you just created. See [Add a Cloud Zone](#).

Add a Cloud Zone

You add a cloud zone and configure its parameters by running a vRealize Orchestrator workflow.

Prerequisites

- Verify that you configured at least one cloud account. See [Add a vSphere Cloud Account](#).
- For information about cloud zones, see [Learn more about vRealize Automation Cloud Assembly cloud zones](#).

Procedure

- 1 Log in to the vRealize Orchestrator Client as an administrator.
- 2 Navigate to **Library > Workflows** and enter the **cloud_zones** tag in the workflow search box.
- 3 Locate **Create Cloud Zone** workflow and click **Run**.

- 4 On the **Summary** tab, configure the cloud zone properties.

Option	Description
Host	Select the vRealize Automation host for which you want to configure a cloud zone.
Region	Select a cloud region.
Name	Enter a name for the cloud zone.
Description	Add a description.
Placement policy	<p>Placement policy drives host selection for deployments within the specified cloud zone.</p> <p>Apply one of the following placement strategies:</p> <ul style="list-style-type: none"> ■ DEFAULT. Places compute resources on random hosts. ■ BINPACK. Places compute resources on the most loaded host that has enough resources to run the given compute. ■ SPREAD. Provisions compute resources, at a deployment level, to the cluster or host with the least number of virtual machines. For vSphere, Distributed Resource Scheduler (DRS) distributes the virtual machines across the hosts.

- 5 On the **Capabilities** tab, add capability tags if you want to support a tagging strategy.

You can add tags now, or later when you edit the cloud account. For information about tagging, see [How do I use tags to manage vRealize Automation Cloud Assembly resources and deployments](#).

- 6 Click **Run**.

Results

You have added a cloud zone.

What to do next

Configure a project and add your cloud zones to it. See [Add a Project](#).

Add a Disk

You create a disk and configure its parameters by running a vRealize Orchestrator workflow.

Procedure

- 1 Log in to the vRealize Orchestrator Client as an administrator.
- 2 Navigate to **Library > Workflows** and enter the **disks** tag in the workflow search box.
- 3 Locate **Create Disk** workflow and click **Run**.
- 4 Select a vRealize Automation host for which you want to configure a disk.
- 5 Enter a name for the disk.
- 6 (Optional) Add a description for the disk.

- 7 Select a project to which to add the disk.
- 8 On the **Specification** tab, define the capacity of the disk in gigabytes.
- 9 Configure any additional settings for the disk, such as tags, constraints, and custom properties.
- 10 Click **Run**.

Results

You have added a disk in vRealize Automation or vRealize Automation Cloud and to the vRealize Orchestrator plug-in inventory.

Add a Machine

You add a virtual machine and configure its parameters by running a vRealize Orchestrator workflow.

Procedure

- 1 Log in to the vRealize Orchestrator Client as an administrator.
- 2 Navigate to **Library > Workflows** and enter the **machines** tag in the workflow search box.
- 3 Locate **Create Machine** workflow and click **Run**.
- 4 Select a vRealize Automation host for which you want to configure a machine.
- 5 Enter a name for the machine.
- 6 Define how many machines you want to provision.
- 7 Select a project to which to add the machine.
- 8 Go to the **Flavors and Images** tab.

- a Select the type of image used for the machine.
- b Select the direct image reference used for the machine.

This setting is required if you have multiple zones of the same cloud connection configured under image mapping.

- c Select the flavor of the machine.

If multi-zone cloud accounts are attached to a project, machines are created for AWS, GCP, and Azure only if the flavor reference is not attached to the machine specification object.

- d (Optional) Select the provider-specific flavor reference for the machine.

This setting is required if you have multiple zones of the same cloud connection configured under flavor mapping.

- 9 Configure any additional settings for the machine, such as tags, custom properties, remote access, and others.

Keep in mind the following considerations:

- When you configure disk specifications, you must select a block device. The block device must be a data disk and must be available to attach to the machine.

You can create a block device using the `createBlockDevice` method. For more information about this method, go to **Plugins > VRA > Objects > VraDiskService** in the vRealize Orchestrator API Explorer.

- When you configure network specifications, you must select a fabric network. Do not specify a Network ID. Network ID and fabric network can't be passed at the same time.

For more information about these settings, consult the signpost help.

- 10 Click **Run**.

Results

You have added a machine in vRealize Automation or vRealize Automation Cloud and to the vRealize Orchestrator plug-in inventory.

Add a Network

You create a network and configure its parameters by running a vRealize Orchestrator workflow.

Prerequisites

Before you add a network in vRealize Orchestrator, make sure that you have an on-demand network isolation policy set up under Network Profiles in vRealize Automation. See [Learn more about network profiles in vRealize Automation](#).

Procedure

- 1 Log in to the vRealize Orchestrator Client as an administrator.
- 2 Navigate to **Library > Workflows** and enter the **networks** tag in the workflow search box.
- 3 Locate the **Create Network** workflow and click **Run**.
- 4 Select a vRealize Automation host for which you want to configure a network.
- 5 Enter a name for the network.
- 6 (Optional) Add a description for the network.
- 7 Select a project to which to add the network.

- 8 On the **Specification** tab, configure the following settings.
 - a Select whether you want to activate or deactivate outbound access.
By default, outbound access is activated, which creates an outbound network.
 - b Select whether you want to create a gateway in the network.
 - c Enter the deployment ID of your vRealize Automation or vRealize Automation Cloud instance.
- 9 Configure any additional settings for the network, such as tags, constraints, and custom properties. For example:
 - a Use constraints to select a network policy defined in vRealize Automation.
 - b Use custom properties to select the Network type in the workflow request, for example, `networkType:private`.

The vRealize Automation plug-in supports the `private`, `routed`, and `outbound` network types.
- 10 Click **Run**.

Results

You have added a network in vRealize Automation or vRealize Automation Cloud and to the vRealize Orchestrator plug-in inventory.

Add a Project

You add a project and configure its parameters by running a vRealize Orchestrator workflow.

Prerequisites

- Verify that you configured at least one cloud account. See [Add a vSphere Cloud Account](#).
- Verify that you configured at least one cloud zone. See [Add a Cloud Zone](#).
- For information about projects, see [Adding and managing vRealize Automation Cloud Assembly projects](#).

Procedure

- 1 Log in to the vRealize Orchestrator Client as an administrator.
- 2 Navigate to **Library > Workflows** and enter the `projects` tag in the workflow search box.
- 3 Locate **Create Project** workflow and click **Run**.
- 4 On the **Summary** tab, select the vRealize Automation host for which you want to configure a project, and enter a name for the project.

- 5 On the **Provisioning: Zones** tab, add one or more cloud zones.
 - a Configure the cloud zone properties.
 - b Select a placement policy.
- 6 Go to the **Provisioning: Resource Tags & Constraints** tab, add tags and constraints for your project.
- 7 Go to the **Provisioning: Custom Properties, Custom Naming & Request Timeout** tab.
 - a Add custom properties that you want added to all requests in the project.
 - b Specify the naming template for machines, networks, security groups, and disks in the project.
 - c If the workloads requested for this project take more than two hours to deploy, enter a longer value for the **Timeout**. The default value is 2 hours.
- 8 Click **Run**.

Results

You have added a project.

Using the vRealize Orchestrator Plug-In for vSphere Update Manager

23

The vRealize Orchestrator plug-in for VMware vSphere® Update Manager™ allows interaction between vRealize Orchestrator and VMware vSphere Update Manager/vSphere Lifecycle Manager. You can use the plug-in to run vRealize Orchestrator workflows that automate vSphere Update Manager 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 vSphere environment.

Role of vRealize Orchestrator with the vSphere Update Manager Plug-in

vRealize Orchestrator powers the vSphere Update Manager plug-in.

You can use the plug-in to run vRealize Orchestrator workflows that interact with vSphere Update Manager to perform automated tasks in the vSphere infrastructure. vSphere Update Manager enables centralized, automated patch and version management for VMware vSphere, and offers support for VMware ESX/ESXi hosts and virtual machines.

With the vSphere Update Manager plug-in, you can perform the following tasks:

- Upgrade and patch ESX/ESXi hosts.
- Install and update third-party software on hosts.
- Upgrade virtual machine hardware and VMware Tools.

To learn more about vSphere Update Manager, see the [vSphere Update Manager Installation and Administration Guide](#).

Functional Prerequisites for the vSphere Update Manager Plug-In

vSphere Update Manager requires network connectivity with VMware vCenter™. Each installation of vSphere Update Manager must be associated with a single vCenter instance. For instructions about configuring a vCenter connection, see [Connect the vSphere Update Manager Plug-In to vCenter](#).

vSphere Update Manager Plug-In Scripting API

The vSphere Update Manager plug-in supports the following:

- vSphere API versions 6.7, 7.0, 7.0 U1, 7.0 U2.
- vSphere Update Manager API version 6.0 to 8.0.

The vSphere Update Manager plug-in scripting API contains classes, with their respective attributes and methods, that allow interaction between vRealize Orchestrator and vSphere Update Manager. You can use the API to develop custom workflows that interact with vSphere Update Manager.

In the vRealize Orchestrator API Explorer, click the **VUM** module in the left pane to expand the hierarchical list of vSphere Update Manager plug-in API scripting objects. You can search the plug-in API and see the documentation for JavaScript objects that you can use in scripted elements. You can copy code from API elements and paste it into scripting boxes.

Using the vSphere Update Manager Plug-In Inventory

The vSphere Update Manager plug-in exposes all objects in the connected vSphere Update Manager instance in the Inventory view in the vRealize Orchestrator Client. You can use the Inventory view to add authorization elements or to run workflows on vSphere objects.

To display all available inventory objects, navigate to **Administration > Inventory > VMware Update Manager** in the vRealize Orchestrator Client.

Access the vSphere Update Manager Plug-In Workflow Library

The vSphere Update Manager plug-in workflow library contains workflows that you can use to run automated processes related to the management of vSphere objects in the inventory of the vCenter with which vSphere Update Manager is registered.

You can integrate standard workflows from the workflow library to create custom workflows.

For a full list of available workflows, navigate to **Library > Workflows > vCenter Update Manager** in the vRealize Orchestrator Client. See [vSphere Update Manager Plug-In Workflow Library](#).

Read the following topics next:

- [Connect the vSphere Update Manager Plug-In to vCenter](#)
- [vSphere Update Manager Plug-In Workflow Library](#)

Connect the vSphere Update Manager Plug-In to vCenter

Before you start running the vSphere Update Manager plug-in workflows, you must associate the plug-in with a vCenter instance.

Procedure

- 1 Log in to the vRealize Orchestrator Client as an administrator.
- 2 Navigate to **Library > Workflows**, and enter the **vcenter** and **configuration** tags in the workflow search box.
- 3 To add a vCenter instance, run the **Add a vCenter server instance** workflow. For detailed instructions about running this workflow, see [Configure the Connection to a vCenter Instance](#).
 - If you add a single vCenter server, it is set as the default vCenter for the vSphere Update Manager plug-in. That server is pre-selected for all vSphere Update Manager plug-in workflows that you run.
 - If you add more than one vCenter server, all instances are registered with the vSphere Update Manager plug-in. No server is pre-selected as the default for the workflows that you run. To set a default server, you must run the **Set a default vCenter with Update Manager** workflow.
- 4 To select a default vCenter to work with, run the **Set a default vCenter with Update Manager** workflow. The default vCenter server that you set is pre-selected for all other workflows that you run.

If you call the REST API directly, instead of run workflows from the vRealize Orchestrator Client, you must set a default host by running the **Set a default vCenter with Update Manager** workflow.
- 5 Depending on the version of vCenter that you use, you might have to accept and import the vSphere Update Manager certificates.
 - If you have vCenter instances that are version 6.x, you must import a vSphere Update Manager server certificate for each vCenter connection by running the **Import a certificate from URL** workflow. The URL must follow the `https://vCenter-Server-IP-address:8084` format.
 - If you have vCenter instances that are 7.x and later, no further configuration is required because vCenter and vSphere Update Manager use the same certificate.

What to do next

You can browse the vSphere Update Manager plug-in inventory, or run some of the workflows that are provided with the plug-in. For a full list of available workflows, see [vSphere Update Manager Plug-In Workflow Library](#).

vSphere Update Manager Plug-In Workflow Library

The vCenter Update Manager workflow category contains a set of standard workflows that cover the most common tasks that you can perform with vSphere Update Manager. You can use the workflows as building blocks for creating complex custom workflows. By combining standard workflows, you can automate multi-step processes in your vSphere environment.

You can manage baselines, patches, and inventory objects by running the default workflows provided with the plug-in.

To access these workflows in the vRealize Orchestrator Client, navigate to **Library > Workflows**, and enter the `vcenter_update_manager` tag in the workflow search box.

Baseline Workflows

Workflow	Description
Attach a baseline	<p>Attaches baselines to a selected vSphere object. The object can be a template, virtual machine, vApp, ESX/ESXi host, folder, cluster, or a data center.</p> <p>Attaching a baseline to a container object, such as a folder or data center, transitively attaches the baseline to all objects in the container.</p>
Create a patch baseline	<p>Creates a new patch baseline. You can apply patch baselines to hosts or virtual machines.</p> <p>Depending on the patch criteria that you select, patch baselines can be dynamic or static (fixed). You can explicitly select the patches to include in the baseline by using the <code>includePatch</code> parameter. You can also use the <code>searchSpec</code> attribute to filter the patches that you want to include. You can filter by product, vendor, severity, and release date. Patches that have been excluded by using the <code>excludePatch</code> parameter will not be included in the baseline, even if they correspond to the filter criteria defined by the <code>searchSpec</code> attribute.</p>
Detach a baseline	Detaches baselines from the selected vSphere inventory objects. To detach inherited baselines, you must detach them from the parent object.
Export baselines	Detaches baselines from the selected vSphere inventory objects. To detach inherited baselines, you must detach them from the parent object.
Filter baselines Filter baselines with no user interaction	Filters baselines depending on the provided filter parameters. You can manually select a baseline from the filtered list to include it as a workflow result.
Get attached entities	Selects entities attached to baselines or baseline groups.
Import baselines	Imports baselines from the .xml file that the Export baselines workflow generates.
Update a patch baseline	Modifies the properties of an existing patch baseline.
Remove baselines	Deletes the baselines you select. Before deletion, the baselines are detached from all vSphere objects that they are attached to.

Patch Workflows

Workflow	Description
Download all patches Download all patches asynchronously	Check for new patches and updates and depending on the availability, download the new patches to the plug-in repository.
Filter patches Filter patches without user interaction	Filter patches and allow you to select a subset of the filtered patch for further processing.
Stage Stage asynchronously	Stage patches to hosts. Staging allows you to download patches and extensions from the vSphere Update Manager/vSphere Lifecycle Manager repository to ESX/ESXi hosts, without applying the patches and extensions immediately. You can stage patches to hosts or container objects such as clusters or data centers. This way, the remediation process is faster because the patches and extensions are already available locally on the hosts. The Stage asynchronously workflow returns an array with task keys for all of the started vCenter tasks.

Compliance and Inventory Workflows

Workflow	Description
Export compliance report	Exports the compliance report to an external file format (CSV, PDF, or HTML). Run this workflow with a selected entity, baseline and compliance status to verify that the upgraded host is compliant against the baseline. Alternatively, you can run this workflow before you create a new patch baseline to discover which hosts are not compliant and need an upgrade. To run the workflow, you select relevant vSphere objects, a set of baselines that you want to check compliance against, a file location, and a file format. After you run the workflow, you can find the report at <code>/data/vco/var/run/vco/</code> .
Get compliance	Retrieves compliance data for the specified object. The object can be a template, virtual machine, vApp, host, cluster, folder or datacenter. The workflow returns information about the compliance state of the vSphere object against the baselines that are attached to it. If the vSphere object is a container, you receive compliance data for all objects in the container.
Remediate Remediate asynchronously	Remediates an inventory object against the specified baselines. You can remediate vSphere objects such as templates, virtual machines, vApps, hosts, folders, clusters, and data centers. The Remediate asynchronously workflow returns an array of vCenter task keys.

Workflow	Description
Scan inventory asynchronously	Scans vSphere objects for applicable patches and updates that are included in the attached baselines. You can scan vSphere objects such as templates, virtual machines, vApps, hosts, folders, clusters, and data centers. If the objects are different types, the workflow starts a separate vCenter task for each object type.
Set a default vCenter with Update Manager	Set a default vCenter to use with vSphere Update Manager. The default vCenter instance is pre-selected for all vSphere Update Manager plug-in workflows that you run. If you have a single vCenter instance it will automatically be set as default.

Using the XML Plug-In

24

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 Workflows with vRealize Orchestrator*.

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

Read the following topics next:

- [Running the XML Plug-In Sample Workflows](#)

Running the XML Plug-In Sample Workflows

You can run the XML plug-in sample workflows from the vRealize 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*.

What to read next

- [Create a Simple XML Document](#)

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

Create a Simple XML Document

You can run a workflow from the vRealize 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 Log in to the vRealize Orchestrator Client.
- 2 Navigate to **Library > Workflows** and enter the `xml` and `samples_xml_(simple)` tags in the workflow search box.
- 3 Locate the **Create a simple XML document** workflow and click **Run**.
- 4 Enter the filepath to the XML document to create.
For example, `c:/orchestrator/filename.xml`.
- 5 Click **Run**.

Results

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 vRealize 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 Log in to the vRealize Orchestrator Client.
- 2 Navigate to **Library > Workflows** and enter the `xml` and `samples_xml_(simple)` tags in the workflow search box.
- 3 Locate the **Find element in document** workflow and click **Run**.
- 4 Type the filepath to the XML document.
For example, `c:/orchestrator/filename.xml`.

- 5 Click **Run**.

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 vRealize Orchestrator Client and click **Logs** on the **Schema** tab.

Modify an XML Document

You can run a workflow from the vRealize 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 Log in to the vRealize Orchestrator Client.
- 2 Navigate to **Library > Workflows** and enter the `xml` and `samples_xml_(simple)` tags in the workflow search box.
- 3 Locate the **Modify XML document** workflow and click **Run**.

- 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 **Run**.

Results

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 vRealize 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 Log in to the vRealize Orchestrator Client.
- 2 Navigate to **Library > Workflows** and enter the `xml` and `samples_xml_(address_book)` tags in the workflow search box.
- 3 Locate the **Full address book test** workflow and click **Run**.
- 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 **Run**.

Results

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