

Using the VMware vRealize Orchestrator Client

vRealize Orchestrator 7.6



vmware®

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

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

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

docfeedback@vmware.com

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

Copyright © 2008-2019 VMware, Inc. All rights reserved. [Copyright and trademark information.](#)

Contents

| | | |
|----------|-----------------------------------------------------------------------------------|-----------|
| 1 | Using the VMware vRealize Orchestrator Client | 4 |
| 2 | The VMware vRealize Orchestrator Client | 5 |
| | Using the API Explorer in the vRealize Orchestrator Client | 6 |
| | vRealize Orchestrator Object Version History | 6 |
| | Using the vRealize Orchestrator Client to Revert a Workflow to an Earlier State | 7 |
| | vRealize Orchestrator Client Usage Dashboard | 7 |
| 3 | Managing vRealize Orchestrator Client Roles and Groups | 8 |
| | Assign Roles in the vRealize Orchestrator Client | 10 |
| | Create Groups in the vRealize Orchestrator Client | 10 |
| 4 | Managing Orchestrator Objects | 12 |
| | Managing Workflows in the vRealize Orchestrator Client | 12 |
| | Standard Workflows in the vRealize Orchestrator Workflow Library | 12 |
| | Create Workflows in the vRealize Orchestrator Client | 13 |
| | vRealize Orchestrator Input Form Designer | 13 |
| | Requests for User Interaction in the vRealize Orchestrator Client | 17 |
| | Schedule Workflows in the vRealize Orchestrator Client | 17 |
| | Create Actions in the vRealize Orchestrator Client | 18 |
| | Configuration Elements in the vRealize Orchestrator Client | 19 |
| | Managing Policies in the vRealize Orchestrator Client | 20 |
| | Resource Elements in the vRealize Orchestrator Client | 22 |
| | Using vRealize Orchestrator Audit Logs | 22 |
| 5 | Using vRealize Orchestrator Packages | 23 |
| | Create a Package in the vRealize Orchestrator Client | 23 |
| | Export a Package in the vRealize Orchestrator Client | 24 |
| | Import a Package in the vRealize Orchestrator Client | 25 |
| 6 | Troubleshooting in the vRealize Orchestrator Client | 27 |
| | Metric Data in the vRealize Orchestrator Client | 27 |
| | Profile Workflows in the vRealize Orchestrator Client | 27 |
| | Using the vRealize Orchestrator System Dashboard | 28 |
| | Using Workflow Token Replay in the vRealize Orchestrator Client | 29 |
| | Validating vRealize Orchestrator Workflows | 29 |
| | Validate a Workflow and Fix Validation Errors in the vRealize Orchestrator Client | 30 |
| | Debug Workflow Scripts in the vRealize Orchestrator Client | 30 |

Using the VMware vRealize Orchestrator Client

1

Using the VMware vRealize Orchestrator Client provides information about the workflow automation features and functionality of the new HTML5-based vRealize Orchestrator Client.

Intended Audience

This information is intended for experienced system administrators who are looking for a tool that can help them to run and manage vRealize Orchestrator workflows.

Note The Java-based Orchestrator Legacy Client is deprecated and is scheduled for removal for future releases of vRealize Orchestrator. For information on using the Orchestrator Legacy Client, see *Using the VMware vRealize Orchestrator Legacy Client*.

The VMware vRealize Orchestrator Client

2

Use the new HTML5-based vRealize Orchestrator Client to manage vRealize Orchestrator services and objects.

You can create and manage vRealize Orchestrator objects by using the vRealize Orchestrator Client. You can find the vRealize Orchestrator Client at [https://your_orchestrator_server_ip_or_dns_address:8283 / vco-controlcenter/client](https://your_orchestrator_server_ip_or_dns_address:8283/vco-controlcenter/client).

| | |
|---------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| REST API communication | The vRealize Orchestrator Client runs on the vRealize Orchestrator Control Center server. The client communicates with the vRealize Orchestrator REST API through a REST proxy. |
| Workflow management | Create, edit, schedule, run, and delete workflows in the vRealize Orchestrator Client. |
| Action management | Create, edit, and delete actions in the vRealize Orchestrator Client. The action editor supports automatic completion for common script elements included in the vRealize Orchestrator API Explorer. |
| Policy management | Create, edit, run, and delete policies in the vRealize Orchestrator Client. |
| Configuration management | Create, run, and delete configuration elements in the vRealize Orchestrator Client. |
| Resource management | Export, import, and update resource elements in the vRealize Orchestrator Client. |
| Metric data | Use the vRealize Orchestrator Client System Dashboard and profiling feature to gather useful metric data about your vRealize Orchestrator environment and workflows. |
| Package management | Create, delete, export, and import packages containing vRealize Orchestrator objects through the vRealize Orchestrator Client. |
| Permissions management | Users with administrator rights can assign roles to users in the vRealize Orchestrator Client and add them to groups. |
| API Explorer | Explore the API commands available in the vRealize Orchestrator Client. |

This chapter includes the following topics:

- [Using the API Explorer in the vRealize Orchestrator Client](#)
- [vRealize Orchestrator Object Version History](#)
- [vRealize Orchestrator Client Usage Dashboard](#)

Using the API Explorer in the vRealize Orchestrator Client

You can browse the vRealize Orchestrator API Explorer to view the documentation for the JavaScript objects that you can use in scripted workflow elements.

The API Explorer includes a list of the vRealize Orchestrator compliant REST objects. The API Explorer includes information about the attributes, constructors, and HTTP methods associated with specific objects. The API Explorer is accessible from the top-right menu of the vRealize Orchestrator Client dashboard and the scripting tabs of the vRealize Orchestrator workflow and action editors.

vRealize Orchestrator Object Version History

The vRealize Orchestrator Client maintains a version history record for each vRealize Orchestrator object. Using the version history, you can compare different vRealize Orchestrator object versions and revert to a previous version.

vRealize Orchestrator creates a version history record for each vRealize Orchestrator object when you save the object. Subsequent changes to vRealize Orchestrator objects create a new version history record. The previous versions history records are preserved and can be used to track changes to the object and revert the object to a previous version. Reverting an object to a previous version creates a new version history record.

The vRealize Orchestrator Client tracks the version history of the following vRealize Orchestrator objects:

- Workflows
- Actions
- Packages
- Policies
- Configuration elements

You can access the version history of an object from the **Version History** tab of the object editor page. If you are attempting to edit an object at the same time as another user, a merge conflict might occur. To resolve the merge conflict, click **Resolve** to the right of the error message. On the **Resolve Conflicts** window you have three options:

- **Use theirs.** Resolve the merge conflict by using the changes made by the other user.
- **Use ours.** Resolve the merge conflict by using your changes.
- **Resolve.** Resolve the merge conflict by editing the displayed change model. If the provided model is invalid, this option is unavailable.

Using the vRealize Orchestrator Client to Revert a Workflow to an Earlier State

You can use version history to revert a workflow to a previously saved state. You can revert the workflow state to an earlier or a later workflow version. You can also compare the differences between the current state of the workflow and a saved version of the workflow.

Procedure

- 1 Log in to the vRealize Orchestrator Client.
- 2 While you are editing a workflow, click **Version History**.
- 3 Select a workflow version and select a version from the **Diff against** drop-down menu to compare the differences.

A window displays the differences between the current workflow version and the selected workflow version.

- 4 Click **Revert** on a workflow version to restore the state of the workflow.

The workflow state is reverted to the state of the selected version.

vRealize Orchestrator Client Usage Dashboard

The vRealize Orchestrator Client dashboard provides a useful tool for monitoring, managing, and troubleshooting vRealize Orchestrator Client workflows.

Information on the vRealize Orchestrator Client dashboard is spread among five panels.

| Window | Description |
|----------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Workflow runs | Provides visual data about the number of running, waiting, and failed workflow runs. |
| Favorite workflows | Displays workflows added to favorites. |
| Waiting for input | Displays pending workflow runs that require further user interaction. These workflows are also displayed in the notifications menu in the upper-right corner of the UI. |
| Recent workflow runs | Manage recent workflow runs. Shows the name, state, start date, and end date of the workflow run. |
| Requiring Attentions | Displays failed workflow runs and workflow run performance metrics. |

Managing vRealize Orchestrator Client Roles and Groups

3

As an administrator, you can use the vRealize Orchestrator Client to set user roles and group permissions for vRealize Orchestrator features and content.

After the vRealize Orchestrator instance is authenticated, the administrator can set permissions that control access to features and content. Permissions in the vRealize Orchestrator Client are separated into role management and group permissions. With role management, you control what vRealize Orchestrator Client features users can view and use. With group permissions, you control what vRealize Orchestrator Client content users can view and use. Content access covered in group permissions includes workflows, actions, policies, configuration elements, and resource elements. You can use groups to organize users into common projects. For example, you can create a group that includes users working on developing a custom vRealize Orchestrator plug-in.

Note Access to pre-configured vRealize Orchestrator content like standard workflows and actions is shared among all users, unless configured otherwise through group permissions.

Roles

Role management is only available for vRealize Orchestrator instances authenticated with vRealize Automation. Permissions management for vRealize Orchestrator instances authenticated with vSphere is limited to group permissions.

| Role | Description |
|--------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Administrator | Can access all vRealize Orchestrator Client features and content, including the content created by specific groups. Responsible for setting user roles, creating and deleting groups, and adding users to groups. Note Tenant administrators from the vRealize Automation environment used to authenticate vRealize Orchestrator have Administrator rights, by default. |
| Workflow Designer | Can create, run, edit, and delete their own vRealize Orchestrator Client content. Can add their own content to their assigned group. Does not have access to the administration and troubleshooting features of the vRealize Orchestrator Client. |

Note vRealize Automation users with no predefined role can still log in to the vRealize Orchestrator Client, but have limited access to client features. If they are part of a group, these users can view and run content associated with that group.

Groups

Group permissions in the vRealize Orchestrator Client are not tied to Active Directory user groups. For information on Active Directory user groups, see *User Group Workflows* in the *Using VMware vRealize Orchestrator Plug-ins* documentation.

| Group user permissions | Description |
|------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Run and edit | Only available for vRealize Orchestrator instances authenticated with vRealize Automation. Can create, edit, add, and run vRealize Orchestrator objects for use in the group. |
| Run | Can view and run vRealize Orchestrator objects included in the group. |

Note Group permissions are tied to the role management system in the vRealize Orchestrator Client. For example, users with no predefined role can have **Run and edit** permissions, but can only view and run their own content or group content, without the ability to create, edit, and add content.

This chapter includes the following topics:

- [Assign Roles in the vRealize Orchestrator Client](#)
- [Create Groups in the vRealize Orchestrator Client](#)

Assign Roles in the vRealize Orchestrator Client

As an administrator, you can add users to the vRealize Orchestrator Client and set what features they can view and use.

Role management controls the access of users from the vRealize Orchestrator identity provider to the features of the vRealize Orchestrator Client. Role management covers both the vRealize Orchestrator Client user interface and the API functionality.

Note Role management is only available for vRealize Orchestrator instances authenticated with vRealize Automation. Permissions management for vRealize Orchestrator instances authenticated with vSphere is limited to group permissions. For more information on setting group permissions, see [Create Groups in the vRealize Orchestrator Client](#).

Procedure

- 1 Log in to the vRealize Orchestrator Client as an administrator.
- 2 Navigate to **Administration > Roles Management**.
- 3 Click **Add**.
- 4 Search for the user or group you want to add to the vRealize Orchestrator Client.
- 5 Select the user's role. For more information on roles, see [Chapter 3 Managing vRealize Orchestrator Client Roles and Groups](#).
- 6 Click **Save**.

Create Groups in the vRealize Orchestrator Client

As an administrator, you can use groups to set what vRealize Orchestrator content users can view and access in the vRealize Orchestrator Client.

You can use the vRealize Orchestrator Client to set group permissions to vRealize Orchestrator workflows, actions, policies, configuration elements, and resource elements.

Note For users from vRealize Orchestrator instances authenticated with vSphere, you can only set **Run** group permissions.

Procedure

- 1 Log in to the vRealize Orchestrator Client as an administrator.
- 2 Navigate to **Administration > Groups**.
- 3 Click **New Group**.
- 4 On the **Summary** tab, add a name and description for the group.

- 5 On the **Users** tab, click **Add**.
 - a Search for a user you want to add to the group.
 - b Assign group permissions to the user.
 - c Click **Add**.
- 6 On the **Items** tab, add vRealize Orchestrator objects to the group.

Note You can also add an object to existing groups when that object is being created in the vRealize Orchestrator Client. To add the object, select the group from the **Accessible by** drop-down menu on the **Summary/General** tab of the object editor.

- 7 Click **Save**.

Managing Orchestrator Objects

You can use the vRealize Orchestrator Client to create, edit, run, delete, and troubleshoot vRealize Orchestrator objects like workflows, actions, policies, configuration elements, and resource elements.

This chapter includes the following topics:

- [Managing Workflows in the vRealize Orchestrator Client](#)

Managing Workflows in the vRealize Orchestrator Client

A workflow is a series of actions and decisions that you run sequentially. vRealize Orchestrator provides a library of workflows that perform common management tasks. vRealize Orchestrator also provides libraries of the individual actions that the workflows perform.

Workflows combine actions, decisions, and results that, when performed in a particular order, finish a specific task or a specific process in a virtual environment. Workflows perform tasks such as provisioning virtual machines, backing up, performing regular maintenance, sending emails, performing SSH operations, managing the physical infrastructure, and other general utility operations. Workflows accept inputs according to their function. You can create workflows that run according to defined schedules, or that run if certain anticipated events occur. Information can be provided by you, by other users, by another workflow or action, or by an external process such as a Web service call from an application. Workflows perform some validation and filtering of information before they run.

Workflows can call upon other workflows. For example, you can reuse in several different workflows a workflow that starts a virtual machine.

You create workflows by using the vRealize Orchestrator Client interface's integrated development environment (IDE), that provides access to the workflow library and the ability to run workflows on the workflow engine. The workflow engine can also take objects from external libraries that you plug in to vRealize Orchestrator. This ability allows you to customize processes or implement functions that third-party applications provide.

Standard Workflows in the vRealize Orchestrator Workflow Library

vRealize Orchestrator provides a standard library of workflows that you can use to automate operations in your virtual infrastructure. The workflows in the standard library are locked in the read-only state. To customize a standard workflow, you must duplicate that workflow. Duplicate workflows or custom workflows that you create are fully editable.

The contents of the workflow library are accessible through the **Library > Workflows** menu of the HTML5-based vRealize Orchestrator Client. Both standard and custom workflows in the client are organized by using tags. For example, you can access the **Generate key pair** workflow by entering **SSH** in the workflow library search box.

Note You cannot add new tags to standard workflows, unless you duplicate the workflow.

Create Workflows in the vRealize Orchestrator Client

You can use the vRealize Orchestrator Client to create and edit workflows.

The vRealize Orchestrator Client uses a new presentation engine. As a result, workflows created in the vRealize Orchestrator Client can only be run and edited in the vRealize Orchestrator Client. Workflows created in the Orchestrator Legacy Client can be run in the vRealize Orchestrator Client, but cannot be edited in the vRealize Orchestrator Client.

Procedure

- 1 Log in to the vRealize Orchestrator Client.
- 2 Navigate to **Library > Workflows**.
- 3 Click **New Workflow**.
- 4 Enter the name of the new workflow and click **Create**.
- 5 Use the workflow editor to configure the variables, workflow inputs and outputs, schema structure, and presentation of the workflow.
- 6 To finish editing the workflow, click **Save**.

Note You can track changes to workflows in the **Version History** tab. For more information, see [vRealize Orchestrator Object Version History](#).

What to do next

You can use the vRealize Orchestrator token replay feature to optimize the performance of your workflows. For more information, see [Using Workflow Token Replay in the vRealize Orchestrator Client](#).

vRealize Orchestrator Input Form Designer

If a workflow requires input parameters, it opens a dialog box in which users enter the required values. You can organize the content, layout, and presentation of this dialog box with the input form designer.

The input form designer is located in the **Input Form** tab of the workflow editor. This designer consists of a navigation menu, design canvas, and properties menu. You can drag inputs and generic elements from the left menu to the design canvas. In the canvas, you can set the position of the input parameters, organize them into separate input tabs, and configure the input parameter properties. The vRealize Orchestrator Client input form designer is the equivalent of the **Presentation** tab of the Orchestrator Legacy Client.

Note You cannot use content from the **Variables** tab of the workflow editor in the input form designer. You can only use parameters from the **Input/Output** tab.

Generic elements You can add generic elements, like drop-down menus and password text boxes, to the input form designer. Generic elements do not correspond to actual input parameters, but can be bound to input parameters.

Create the Workflow Input Parameters Dialog Box in the vRealize Orchestrator Client

You can use the input form designer to create and customize the workflow input parameter dialog box.

Prerequisites

Verify that the workflow has a defined list of input parameters.

Procedure

- 1 Log in to the vRealize Orchestrator Client.
- 2 Navigate to **Library > Workflows**.
- 3 Select your custom workflow.
- 4 Click the **Input Form** tab.
- 5 (Optional) Create tabs for use in the input dialog box.
You can use tabs to organize the structure of your dialog box.
- 6 Select your input parameters.
- 7 Edit the properties of the input parameters.
For more information on input parameter properties, see [Input Parameter Properties in the vRealize Orchestrator Client](#).
- 8 (Optional) Add generic elements to the canvas and bind them to input parameters.
- 9 (Optional) Add external validation to the input parameters. For more information, see [Using Actions to Validate vRealize Orchestrator Workflow Inputs](#).
- 10 Click **Save**.

You created the layout of the workflow dialog box and set the properties of the input parameters.

Input Parameter Properties in the vRealize Orchestrator Client

You can set parameter properties to constrain the input parameters that users provide when they run vRealize Orchestrator workflows.

With vRealize Orchestrator, you can define the parameter properties used to quantify the input parameter values used in workflows. The parameter properties you define impose limits on the types and values of the input parameters that users can provide in vRealize Orchestrator workflows.

Parameter properties validate the input parameters and modify the presentation of the text boxes that appear in the input parameters dialog box. Some parameter properties can create dependencies between parameters.

| Parameter Property | Description |
|---------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Label | Set the input parameter label. |
| Display type | Set the input text box display type. |
| Visibility | Set the visibility of the input parameter. |
| Read-only | Set the input text box as read-only. |
| Custom help | Set the input parameter signpost description. |
| Default value | Set the default value of the input parameter. |
| Step | Used for number type inputs. Set by how much the value of the input parameter increases per click. |
| Required | Sets if the input parameter value is mandatory or not. |
| Regular expression | Validates the input by using a regular expression. |
| Minimum value | Set the minimum value or length of the parameter. |
| Maximum value | Set the maximum value or length of the parameter. |
| Match text box | Set the input parameter value to match the value of another input parameter. |
| Value source | Set the value source of the parameter properties in the Appearance , Value , and Constraints tabs. Note You can import the value of external actions by using External source . The filtering of available actions is done by parameter type. |

Using Actions to Validate vRealize Orchestrator Workflow Inputs

Use external actions to validate the inputs of your custom workflows.

Prerequisites

Create a custom workflow with input parameters. For more information, see

[Create Workflows in the vRealize Orchestrator Client](#). You can use the input form designer to create external validations for your workflow inputs. External validations use action scripts that return a string value when the input parameter value contains an error. If the input parameter value is valid, the external validation returns nothing.

Procedure

- 1 Log in to the vRealize Orchestrator Client.
- 2 Create a validation action.
 - a Navigate to **Library > Actions**.
 - b Click **New Action**.
 - c Enter the required information on the **Summary** tab.
 - d Enter the validation action input parameters.

Note The names of the validation action input parameters must be identical to the names of the workflow input parameters that are being validated.

- e Enter the script of the validation action on the **Script** tab.

```

if (in_1=="invalid") {
                                return "in_1 can't be invalid!";
}

                                if (in_2=="invalid") {
                                    return "in_2 can't be invalid!";
                                }

//inputs are valid, return nothing

```

Note The preceding script is a simple example and does not represent the full scope of the validation scripts that can be used.

- f Click **Save**.
- 3 Apply external validation.
 - a Navigate to **Library > Workflows**.
 - b Select your custom workflow.
 - c Select the **Input Form** tab.
 - d Select the clipboard icon on the top-left of the screen.
 - e Drag an vRealize Orchestrator validation element into the canvas.
 - f Select the validation element, enter a validation label, and select the validation action.
 - g (Optional) Create additional validation elements.
 - h Click **Save**.
- 4 Run the workflow.

If the validation encounters an error, it returns a string. If the validation is successful, the validation returns nothing and workflow run continues.

You have created an external validation for your custom vRealize Orchestrator workflow.

Requests for User Interaction in the vRealize Orchestrator Client

Workflows can request additional user input before they can finish.

Workflows requiring further user interaction suspend operations until the requested input parameters are provided by the user. Workflows define which users can provide the requested information and send requests for interaction accordingly. Workflows waiting for user input are displayed in the **Recent Workflow Runs** panel of the vRealize Orchestrator Client dashboard and the top-right notification menu.

Schedule Workflows in the vRealize Orchestrator Client

You can use scheduling to automate your vRealize Orchestrator workflow runs.

When you schedule workflow runs, you set the date, time, and intervals at which the scheduled task runs.

Procedure

- 1 Log in to the vRealize Orchestrator Client.
- 2 Select your workflow from the **Library** menu, and on the workflow panel, click **Schedule**.
- 3 Configure the scheduled task parameters in the **General**, **Scheduling**, and **Workflow** categories.

Note The **Workflow** parameter category is visible only for workflows that require input parameters .

| Parameter | Description |
|-----------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Name | The name of the scheduled task. |
| Description | A short description detailing the purpose of the scheduled task. |
| Start | The date and time of the first scheduled run of the workflow. |
| Start if in the past | Select whether to start the workflow, if the scheduled time is in the past. Yes starts the scheduled workflow immediately. No starts the workflow at the next scheduled recurrence. |
| Schedule | Set the recurrence pattern and event trigger entries of the scheduled task. |
| End date | Only visible if No Recurrence is selected. Set the date and time of when the scheduled task ends. |
| Workflow | Enter the input parameters of the workflow. |

- 4 Click **Create**.

You have created a scheduled task for the workflow. Scheduled workflows appear under **Activity > Scheduled**. You can delete scheduled tasks by clicking **Delete** on the schedule panel.

Edit Scheduled Task in the vRealize Orchestrator Client

Scheduled tasks can be edited to change parameters such as date, time, and recurrence of the scheduled workflow.

Prerequisites

Create a scheduled workflow task.

Procedure

- 1 Log in to the vRealize Orchestrator Client.
- 2 Select your scheduled task from **Activity > Scheduled**.
- 3 Click **Edit** on the workflow panel.
- 4 Edit the schedule and click **Save**.

Note Input parameters set when creating the scheduled task are read-only and cannot be edited. To change these parameters, create a new scheduled task for this workflow.

Create Actions in the vRealize Orchestrator Client

You can use the vRealize Orchestrator Client to create, edit, and delete action scripts.

The vRealize Orchestrator Client provides libraries of predefined actions and an action editor for custom action scripts. Actions represent individual functions that you use as building blocks in workflows.

Actions are JavaScript functions. Actions can take multiple input parameters and have a single return value. Actions can call on any object in the vRealize Orchestrator API, or objects in any API that you import into vRealize Orchestrator by using a plug-in.

When a workflow runs, an action takes input parameters from the workflow's variables. These variables can be either the workflow's initial input parameters, or variables that other elements in the workflow set when they run.

The action editor includes an autocomplete feature for scripts and an API Explorer featuring the available scripting types and their documentation.

Procedure

- 1 Log in to the vRealize Orchestrator Client.
- 2 Navigate to **Library > Actions**.
- 3 Click **New Action**.
- 4 On the **General** tab, enter the name and module name of the action.

Note The name and module name must be unique for every action. The action name must be a valid JavaScript function. The action name must be a single word that can only contain letters, numbers, and the dollar ("\$\$") and underscore ("_") symbols. The module name must consist of words separated by the dot (".") character.

- 5 (Optional) Create a description, version number, tags, and group permissions for the action.
- 6 On the **Script** tab, add action inputs, select the return type of the output, and write the script.

- 7 To finish editing the action, click **Save**.

A message states that the action is saved.

What to do next

You can use the new custom action in workflows.

Configuration Elements in the vRealize Orchestrator Client

A configuration element is a list of variables, that you can use to configure constants across a whole vRealize Orchestrator server deployment.

You can use configuration elements to make variables available to all the workflows, actions, and policies running on the vRealize Orchestrator server.

If you create a package containing a workflow, action, or policy that uses a variable from a configuration element, vRealize Orchestrator automatically includes the configuration element in the package. If you import a package containing a configuration element into another vRealize Orchestrator server, you can import the configuration element variable values as well. For example, if you create a workflow that requires variable values that depend on the vRealize Orchestrator server on which it runs, setting those variables in a configuration element lets you export that workflow, so that another vRealize Orchestrator server can use it. Configuration elements therefore allow you to exchange workflows, actions, and policies between servers more easily.

Note You cannot import values of a configuration element variable from a configuration element exported from vRealize Orchestrator 5.1 or earlier.

Create Configuration Elements in the vRealize Orchestrator Client

With configuration elements, you can set common variables across an vRealize Orchestrator server. All elements that are running in the server can use the variables you set in a configuration element.

Procedure

- 1 Log in to the vRealize Orchestrator Client.
- 2 Navigate to **Assets > Configurations**.
- 3 Select **New Configuration**.
- 4 Enter the configuration element name.
- 5 Select the **Variables** tab.
- 6 To create a local variable, click **New**.
 - a Enter the variable name.
 - b Select the variable type.

Note To create an array of configuration variables, select the **Array** check box.

- c (Optional) Enter a value for the configuration variable.
- d Click **Save**.

7 To finish creating a configuration element, click **Save**.

What to do next

You can use the configuration element to provide variables to workflows, actions, or policies.

Managing Policies in the vRealize Orchestrator Client

Policies are event triggers that monitor the activity of the system. Policies respond to predefined events issued by changes in the status or performance of specific vRealize Orchestrator objects.

Policies are a series of rules, gauges, thresholds, and event filters that run certain workflows or scripts when specific predefined events occur in vRealize Orchestrator or in the technologies that vRealize Orchestrator accesses through plug-ins. vRealize Orchestrator constantly evaluates the policy rules while the policy is running. For instance, you can implement policy gauges and thresholds that monitor the behavior of vCenter Server objects of the VC:HostSystem and VC:VirtualMachine types.

Create and Apply Policies in the vRealize Orchestrator Client

You can use policies to monitor the activity of the vRealize Orchestrator system for specific events.

Procedure

- 1 Log in to the vRealize Orchestrator Client.
- 2 Navigate to **Library > Policies**.
- 3 Select **New Policy**.
You have created a blank policy.
- 4 Enter a policy name and version number.
- 5 Select the **Variables** tab.
- 6 To create a local variable, click **New**.
 - a Enter the variable name.
 - b Select the variable type.

Note To create an array of policy variables, select the **Array** check box.

- c Enter the variable value.

Note To import the value of a configuration element variable, you can use **Bind to configuration**.

- d Click **Save**.

- 7 On the **Definition** tab, add policy elements and set event handlers.

For more information on policy elements, see [Policy Elements in the vRealize Orchestrator Client](#).

- 8 Click **Save**.

You have configured the policy.

What to do next

To start a policy, select the policy and click **Run**. Enter the policy run name and, if prompted, the required input parameters.

To view the policy status, navigate to **Activity > Policy Runs**.

Policy Elements in the vRealize Orchestrator Client

You can use policy elements to run predefined vRealize Orchestrator workflows or scripts when an event occurs.

You can add a policy element to trigger workflow or script runs as a response to events triggered by objects. With the periodic event element, you can schedule workflow or script runs. With the root element, you can set the start or stop behavior of policies. Policy elements can have event handlers that define when policy elements must run.

Note Event handlers that activate policy elements can be either workflows or action scripts. If you add both a workflow and a script to an event handler, the policy ignores the script trigger and only uses the workflow trigger.

| Event Handler | Description |
|------------------|--------------------------------------------------------------------------------------------------------------------------|
| OnInit | The policy element is triggered every time you start the policy. |
| OnExit | The policy element is triggered every time you stop the policy. |
| OnExecute | Used by the periodic event element. Triggers the policy element during the time specified in the periodic event element. |

Note Technologies plugged in to the vRealize Orchestrator database can possess unique event handlers. For example, through the SNMP plug-in, you can use the **OnTrap** event handler when creating SNMP-based policy elements.

Policy elements are configured on the **Definition** tab of the policy edit window.

Manage Policy Runs in the vRealize Orchestrator Client

You can use the vRealize Orchestrator Client to manage the policy priority and server start-up behavior of policies for when the vRealize Orchestrator server is restarted.

Prerequisites

Create and run a policy. For more information, see [Create and Apply Policies in the vRealize Orchestrator Client](#).

Procedure

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

2 Navigate to **Activity > Policy Runs**.

3 Click the policy run you want to manage.

4 Click **Stop**.

The policy state changes to **Stopped**.

5 On the **General** tab, set the policy priority and server start-up behavior.

6 To restart the policy, click **Run**.

The policy state changes to **Running**.

Resource Elements in the vRealize Orchestrator Client

Workflows might use objects that you create independently of vRealize Orchestrator as attributes. To use external objects as attributes in workflows, you import them into the server as resource elements.

Objects that vRealize Orchestrator workflows can use as resource elements include image files, scripts, XML templates, HTML files, and so on. Any workflows that run in the vRealize Orchestrator server can use any resource elements that you import into vRealize Orchestrator.

After you import an object into vRealize Orchestrator as a resource element, you can make changes to the object in a single location, and propagate those changes automatically to all the workflows that use this resource element.

The maximum size for a resource element is 16 MB.

You can import, export, restore, update, and delete a resource element.

Using vRealize Orchestrator Audit Logs

You can view the audit logs of a vRealize Orchestrator object and their runs. You can also view the logs of all the events in the vRealize Orchestrator Client.

Procedure

1 Log in to the vRealize Orchestrator Client.

2 To view the logs of a particular vRealize Orchestrator object, open the object and click the **Audit** tab.

3 To view the logs of all events occurring in the vRealize Orchestrator Client, click **Audit Logs** on the client navigation menu.

Note Filtering audit logs by severity level displays all log entries equal or greater than the selected severity level. For example, filtering by **info** also displays **warning** and **error** log entries.

Using vRealize Orchestrator Packages

5

Use the vRealize Orchestrator Client to create, export, and import packages. Packages can be used to export workflow objects for use on other vRealize Orchestrator instances.

Packages can contain workflows, actions, policies, configuration elements, or resource elements.

When you add an element to a package, vRealize Orchestrator checks for dependencies and adds any dependent elements to the package. For example, if you add a workflow that uses actions or other workflows, vRealize Orchestrator adds those actions and workflows to the package.

When you import a package, the server compares the versions of the different elements of its contents to matching local elements. The comparison shows the differences in versions between the local and imported elements. The user can decide whether to import the package, or can select specific elements to import.

For most objects created in the vRealize Orchestrator Client, aside from resource elements, packages are the only way to export and import these objects.

Packages use digital rights management to control how the receiving server can use the contents of the package. vRealize Orchestrator signs packages and encrypts the packages for data protection. Packages can track which users export and redistribute elements by using X509 certificates.

Note The vRealize Orchestrator Client does not support the creation, import, or export of encrypted packages. To use encrypted packages in the vRealize Orchestrator Client, you must import them to the Orchestrator Java Client and remove the package encryption.

Create a Package in the vRealize Orchestrator Client

You can export and import workflows, policies, actions, plug-in references, resource elements, and configuration elements in packages. All dependent elements related to package objects are added to the package automatically, to ensure compatibility between versions. To delete dependent elements, you must first remove the related package object.

For most objects created in the vRealize Orchestrator Client, aside from resource elements, packages are the only way to export and import these objects.

Prerequisites

Verify that the vRealize Orchestrator server contains objects like workflows, actions, and policies, that you can add to a package.

Procedure

- 1 Log in to the vRealize Orchestrator Client.
- 2 Navigate to **Assets > Packages**.
- 3 Click **New Package**.
- 4 On the **General** tab, enter a name and description for the package.

Note You cannot use special characters when naming packages in the vRealize Orchestrator Client.

- 5 On the **Content** tab, click **Add**.
- 6 Select the objects that you want to add to the package and click **Add**.

Note Dependant elements are added to the package automatically, but are not displayed in the **Content** tab during package creation. To view dependant elements, select the **Content** tab after package creation.

- 7 To finish creating the package, click **Create**.

Export a Package in the vRealize Orchestrator Client

You can use the vRealize Orchestrator Client to export packages to another vRealize Orchestrator environment.

Prerequisites

Create a package containing the vRealize Orchestrator objects you want to export. For more information, see [Create a Package in the vRealize Orchestrator Client](#) .

Procedure

- 1 Log in to the vRealize Orchestrator Client.
- 2 Navigate to **Assets > Packages**.
- 3 Click **Export** on the package.

4 (Optional) Select additional export options.

| Option | Description |
|------------------------------------------------------------|------------------------------------------------------------|
| Add configuration attribute values to package | Export the attribute values of the configuration elements. |
| Add configuration SecureString attribute values to package | Export the SecureString configuration attribute values. |
| Add global tags to package | Export the global tags. |

5 Set the access rights for users who import the package.

| Option | Description |
|----------------|-----------------------------------------------------------------------|
| View contents | The user can view the package content. |
| Add to package | The user can add content from the imported package to other packages. |
| Edit contents | The user can edit the package content. |

6 Click **Ok**.

Note Files with the .package extension are saved to a default folder on your local machine. To set a custom folder, you can change the storage settings in your browser.

You exported the package. You can now use the exported objects on another vRealize Orchestrator environment.

Import a Package in the vRealize Orchestrator Client

You can use the vRealize Orchestrator Client to import workflow packages. By importing packages, you can reuse objects from one vRealize Orchestrator server on another server.

Prerequisites

- Back up any standard vRealize Orchestrator objects that you have modified.
- On the remote server, create and export a package with the objects you want to import.

Procedure

- 1 Log in to the vRealize Orchestrator Client.
- 2 Navigate to **Assets > Packages**.
- 3 Click **Import**, browse to the .package file that you want to import, and click **Open**.

4 Review the imported package information.

- a The **General** tab contains information about the imported package like the name, description, number of contained items, and certificate information.

You might be prompted to indicate that you trust the publisher certificate of the source vRealize Orchestrator instance before you can import the file.

- b The **Package elements** tab lists the objects included in the import file. If the version of an object in the package is later than the version on the server, the system selects that object version for import. Earlier versions of vRealize Orchestrator elements must be selected manually.
- c Deselect **Import Configuration Attribute Values** if you do not want to import the attribute values of the configuration elements from the package.
- d From the drop-down menu, select whether you want to import tags.

5 Click **Import**.

Troubleshooting in the vRealize Orchestrator Client

6

You can troubleshoot and monitor your vRealize Orchestrator instance by using metrics, token replay, validation, and debugging.

This chapter includes the following topics:

- [Metric Data in the vRealize Orchestrator Client](#)
- [Using Workflow Token Replay in the vRealize Orchestrator Client](#)
- [Validating vRealize Orchestrator Workflows](#)
- [Debug Workflow Scripts in the vRealize Orchestrator Client](#)

Metric Data in the vRealize Orchestrator Client

vRealize Orchestrator administrators can use workflow profiling and the System Dashboard metrics to troubleshoot the vRealize Orchestrator system and workflows.

The profiling feature gathers metric data about workflow runs. Workflow profiling is enabled by default. You can disable automatic profiling in **Control Center > Extension Properties > profiler-7.6.0**.

The other source for metric data in the vRealize Orchestrator Client is the System Dashboard, that provides system level metrics. For more information, see [Using the vRealize Orchestrator System Dashboard](#).

Profile Workflows in the vRealize Orchestrator Client

You can profile your workflow runs to troubleshoot and optimize your vRealize Orchestrator environment.

You can use the profiling feature of the vRealize Orchestrator Client to gather useful metric data about your workflow runs. This data can be used to optimize the performance of your workflows. By default, workflow runs are profiled automatically. You can disable automatic profiling from the **Extension Properties** page of the vRealize Orchestrator Control Center and run the profiler manually. To do a manual profiling run, find your workflow in the library and select **Actions > Profile**.

Prerequisites

Run a workflow.

Procedure

- 1 Log in to the vRealize Orchestrator Client.
- 2 Navigate to **Activity > Workflow Runs**.
- 3 Select a workflow run.

On the workflow run schema, you can see data about the individual workflow items. Data includes total run duration, max duration, and number of item runs. You can filter this information from the drop-down menu on the top right of the page.

- 4 Select the **Performance** tab.

This tab provides you with metric data on workflow run CPU times, run duration, token size, and workflow item data.

Note If the workflow run is suspended, for example when the workflow is waiting for further input, the CPU times metric only captures the runtime thread that occurred before completion.

What to do next

Use the data gathered from profiling to optimize your workflow.

Using the vRealize Orchestrator System Dashboard

As an administrator, you can use the vRealize Orchestrator Client System Dashboard to gather useful metric data about the nodes of your vRealize Orchestrator environment.

You can access the System Dashboard from by clicking the **System** tab, on the top of vRealize Orchestrator Client dashboard page. Provided data includes:

- Node status
- Node properties
- Cluster settings. You can only view the cluster settings from the System Dashboard. To change these settings, go to the **Orchestrator Cluster Management** page of the vRealize Orchestrator Control Center.
- Threads info
- Heap memory
- Non-heap memory
- File system use
- Authentication data
- Orchestrator database connection pool
- Process input arguments

This data can be used to monitor the state of individual nodes of your vRealize Orchestrator environment and troubleshoot problems. To navigate between individual nodes, click the tab associated with a node on the top of the System Dashboard.

Using Workflow Token Replay in the vRealize Orchestrator Client

You can use the token replay feature to view the transitions between items in workflow runs.

The token replay feature records contextual information for each transition between workflow items. For each workflow item, token replay records when the workflow run started, ended, and what variables were changed at the end of the workflow item run. Token replay also references the generated script log messages for each workflow item.

Note Data about workflow item transitions is stored in the vRealize Orchestrator PostgreSQL database. This data is removed from the database when the workflow run is deleted.

Prerequisites

Run a workflow.

Procedure

- 1 Log in to the vRealize Orchestrator Client as an administrator.
- 2 Navigate to **Activity > Workflow Runs**.
- 3 Select a workflow run.
- 4 Select a workflow run item from the left menu.

The **Variable** and **Logs** tabs now display information specific for that workflow item.

Validating vRealize Orchestrator Workflows

vRealize Orchestrator provides a workflow validation tool. Validating a workflow helps identify errors in the workflow and checks that the data flows from one element to the next correctly.

By default, vRealize Orchestrator always performs a workflow validation when you run a workflow.

When you validate a workflow, the validation tool creates a list of any errors or warnings. Clicking an error in the list highlights the workflow element that contains the error.

If you run the validation tool in the workflow editor, the tool provides suggested quick fixes for the errors it detects. Some quick fixes require additional information or input parameters. Other quick fixes resolve the error for you.

Workflow validation checks the data bindings and connections between elements. Workflow validation does not check the data processing that each element in the workflow performs, and so, a valid workflow might run incorrectly and produce erroneous results if a function in a schema element is incorrect.

Validate a Workflow and Fix Validation Errors in the vRealize Orchestrator Client

You must validate a workflow before you can run it. You can only fix validation errors if you have opened the workflow for editing.

Prerequisites

Verify that you have a complete workflow to validate, with schema elements linked and bindings defined.

Procedure

- 1 Log in to the vRealize Orchestrator Client as an administrator.
- 2 Navigate to **Library > Workflows** and select the workflow you want to validate.
- 3 Click **Edit**.
- 4 Click **Validate** from the top menu.

If the workflow is valid, a confirmation message appears. If the workflow is invalid, a list of errors appears.

- 5 For an invalid workflow, click an error message and take appropriate steps to resolve the problem.

The validation tool highlights the schema element in which the error occurs by adding a red icon to it. Where possible, the validation tool displays a quick fix action.

- If you agree with the proposed quick fix action, click it to perform that action.
- If you disagree with the proposed quick fix action, close the Workflow Validation dialog box and fix the schema element manually.

Important Always check that the fix that vRealize Orchestrator proposes is appropriate.

For example, the proposed action might be to delete an unused attribute, when in fact that attribute might not be bound correctly.

- 6 Repeat the preceding steps until you have eliminated all validation errors.

You validated a workflow and fixed the validation errors.

What to do next

You can run the workflow.

Debug Workflow Scripts in the vRealize Orchestrator Client

You can debug workflow runs by inserting breakpoints in the script of workflow items.

When a breakpoint is reached, you have several options to continue the debugging process. When you debug an element from the workflow schema, you can view the general information about the workflow run, modify the workflow variables, add expressions to watch, and view log messages.

Note Perform all script debugging in a non-production environment.

Procedure

- 1 Log in to the vRealize Orchestrator Client as an administrator.
- 2 Select a workflow from the library.
- 3 Open the workflow schema, select a workflow element, and click the **Scripting** tab.
- 4 To insert a breakpoint, click the red circle to the left of the line number.

Note You can only insert breakpoints in workflow elements with scripting.

- 5 To run the workflow in the debugging mode, click **Debug**.

If the workflow requires input parameters, you must provide them.

- 6 When the workflow run is paused after reaching a breakpoint, select one of the available options.

| Option | Description |
|------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------|
| Continue | Resumes the workflow run until another breakpoint is reached or the workflow run finishes. |
| Step into | You can use this option to step into a workflow element. You cannot step into a nested workflow element when you debug a workflow in the workflow editor. |
| Step over | Skips the current element in the schema and pauses the workflow run on the next element. |

Note You can instruct the debugger to ignore the current breakpoint by clicking the breakpoint. This changes the breakpoint symbol to a green triangle.

- 7 (Optional) On the **Debugger** tab, insert expressions to watch.
You can use expressions to follow the completion of specific variables.
- 8 (Optional) On the **Debugger** tab, modify the values of variables.